

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Projeto de uma Interface Gráfica para o Ambiente BInXS

Fabíola Weis Jaeger

**Florianópolis - SC
2007/2**

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

Projeto de uma Interface Gráfica para o Ambiente BInXS

Fabíola Weis Jaeger

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

**Florianópolis - SC
2007/2**

Fabíola Weis Jaeger

Projeto de uma Interface Gráfica para o Ambiente BInXS

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

Orientador: Prof. Ronaldo dos Santos Mello, Dr.

Universidade Federal de Santa Catarina

ronaldo@inf.ufsc.br

Banca examinadora

André Fabiano Dyck

Universidade Federal de Santa Catarina

dyck@npd.ufsc.br

Prof. José Leomar Todesco, Dr.

Universidade Federal de Santa Catarina

tite@inf.ufsc.br

SUMÁRIO

| | | |
|------------|---|-----------|
| 1 | Introdução..... | 10 |
| 1.1 | Motivação | 10 |
| 1.2 | Objetivos..... | 11 |
| 1.3 | Organização do Texto | 12 |
| 2 | O Ambiente BInXS..... | 13 |
| 2.1 | XML - Extensible Markup Language | 13 |
| 2.2 | Descrição do Ambiente..... | 14 |
| 2.3 | Intervenção do Usuário..... | 15 |
| 2.3.1 | Etapa da Conversão do Esquema XML..... | 15 |
| 2.3.2 | Etapa da Integração Semântica | 24 |
| 3 | Técnicas de Projeto de Interface | 37 |
| 3.1 | Usabilidade | 37 |
| 3.2 | Abordagem The Usage-Centered Design | 38 |
| 3.2.1 | Modelo de Usuários..... | 40 |
| 3.2.2 | Modelo de Casos de Tarefas | 41 |
| 3.2.3 | Modelo de conteúdo da interface | 44 |
| 3.2.4 | Maquete/protótipo da interface | 45 |
| 3.2.5 | Crítica da Abordagem..... | 46 |
| 3.3 | Abordagem The Brigde..... | 46 |
| 3.3.1 | Etapa 1 – Fluxo de tarefas..... | 46 |
| 3.3.2 | Etapa 2 – Mapear os fluxos da tarefa em objetos da tarefa | 47 |
| 3.3.3 | Etapa 3 – Mapear objetos de tarefa em objetos de interface..... | 50 |
| 3.3.4 | Crítica da Abordagem..... | 51 |
| 3.4 | Considerações Finais do Capítulo..... | 51 |
| 4 | Projeto da Interface Gráfica do Ambiente BInXS | 52 |
| 4.1 | Layout da Interface Gráfica | 52 |
| 4.2 | Fase de Pré-processamento do Esquema..... | 56 |
| 4.2.1 | Etapa 1 para as tarefas 1 e 2..... | 56 |
| 4.2.2 | Etapa 2 para as tarefas 1 e 2..... | 57 |
| 4.2.3 | Etapa 3 para as tarefas 1 e 2..... | 58 |
| 4.3 | Fase de Conversão do Esquema..... | 60 |
| 4.3.1 | Etapa 1 para as tarefas 3 e 4..... | 60 |
| 4.3.2 | Etapa 2 para as tarefas 3 e 4..... | 61 |
| 4.3.3 | Etapa 3 para as tarefas 3 e 4..... | 62 |
| 4.4 | Fase de Reestruturação do Esquema..... | 63 |
| 4.4.1 | Etapa 1 para as tarefas 5, 6, 7, 8 e 9 | 63 |
| 4.4.2 | Etapa 2 para as tarefas 5, 6, 7, 8 e 9 | 63 |
| 4.4.3 | Etapa 3 para as tarefas 5, 6, 7, 8 e 9 | 64 |
| 4.5 | Fase da Unificação..... | 68 |
| 4.5.1 | Etapa 1 para as tarefas 10, 11 e 12 | 68 |
| 4.5.2 | Etapa 2 para as tarefas 10, 11 e 12 | 70 |
| 4.5.3 | Etapa 3 para as tarefas 10, 11 e 12 | 70 |
| 4.6 | Fase da Inclusão de relações de herança | 72 |
| 4.6.1 | Etapa 1 para a tarefa 13 | 72 |
| 4.6.2 | Etapa 2 para a tarefa 13 | 72 |

| | | |
|------------|---|--------------------------------------|
| 4.6.3 | Etapa 3 para a tarefa 13 | 73 |
| 4.7 | Fase de Reestruturação | 73 |
| 4.7.1 | Etapa 1 para a tarefa 14, 15 e 16 | 73 |
| 4.7.2 | Etapa 2 para as tarefas 14, 15 e 16 | 75 |
| 4.7.3 | Etapa 3 para as tarefas 14, 15 e 16 | 75 |
| 5 | Atividades Futuras | Erro! Indicador não definido. |
| | Referências Bibliográficas..... | 78 |

LISTA DE FIGURAS

| | |
|--|----|
| Figura 2.1 - Arquitetura do BInXS (MELLO, 2005)..... | 15 |
| Figura 2.2 - Processo de Conversão de um esquema XML..... | 16 |
| Figura 2.3 - Esquema XML (DTD) de um documento XML com livros de preferência de um leitor..... | 17 |
| Figura 2.4 - Remoção de elementos componentes e atributos sem semântica..... | 17 |
| Figura 2.5 - DTD que esquematiza publicações científicas de autores..... | 18 |
| Figura 2.6- Conversão de elemento componente com repetição..... | 20 |
| Figura 2.7 - Exemplo de esquema conceitual resultante após a intervenção do usuário na regra de conversão de elementos componente com repetição..... | 20 |
| Figura 2.8 - DTD que considera os periódicos publicados por uma editora..... | 21 |
| Figura 2.9 - Exemplo de esquema conceitual resultante após a intervenção do usuário na regra de conversão de auto-relacionamentos..... | 21 |
| Figura 2.10 - Exemplo de generalização de conceitos..... | 23 |
| Figura 2.11 - Etapa da integração semântica..... | 25 |
| Figura 2.12 - Exemplo da regra da unificação de nome..... | 26 |
| Figura 2.13 - Resultado das unificações léxicas para os clusters de afinidade..... | 26 |
| Figura 2.14 - Exemplo de afinidade entre relacionamentos..... | 27 |
| Figura 2.15 - Exemplo de um relacionamento que possui afinidade com mais de um relacionamento..... | 28 |
| Figura 2.16 - Exemplo de relacionamento que não possui afinidade com outro relacionamento..... | 28 |
| Figura 2.17 - Exemplos de relacionamentos globais de associação..... | 30 |
| Figura 2.18 - Exemplo de unificação mista..... | 31 |
| Figura 2.19 - Exemplo de inclusão de relacionamento de herança..... | 33 |
| Figura 2.20 - Exemplo de união de disjunções..... | 34 |
| Figura 2.21 - Exemplo de inclusão de disjunções..... | 34 |
| Figura 2.22 - Exemplo de definição de disjunções..... | 35 |
| Figura 3.1 - Estrutura de usabilidade..... | 38 |
| Figura 3.2 - Modelos da abordagem The Usage-Centered Design..... | 39 |
| Figura 3.3 - Exemplo de papel de usuário..... | 40 |
| Figura 3.4 - Exemplo de um caso de tarefa..... | 41 |
| Figura 3.5 - Exemplo de especialização entre casos de tarefa..... | 42 |
| Figura 3.6 - Exemplo de extensão entre casos de tarefa..... | 43 |
| Figura 3.7 - Exemplo de composição entre casos de tarefa..... | 43 |
| Figura 3.8 - Exemplo de afinidade entre casos de tarefa..... | 43 |
| Figura 3.9 - Notações para os mapas de navegação..... | 45 |
| Figura 3.10 - Exemplo de fluxo de tarefa..... | 47 |
| Figura 3.11 - Exemplo de classe de objeto de tarefa..... | 49 |
| Figura 4.1 - Layout da interface do ambiente BInXS..... | 53 |
| Figura 4.2 - Opções do menu File..... | 54 |
| Figura 4.3 - Opções do menu Edit..... | 54 |
| Figura 4.4 - Opções do menu Schema Conversion..... | 55 |
| Figura 4.5 - Opções do menu Integration..... | 55 |
| Figura 4.6 - Opções do menu Help..... | 56 |
| Figura 4.7 - Fluxo de tarefa para a tarefa 1..... | 57 |
| Figura 4.8 - Fluxo de tarefa para a tarefa 2..... | 57 |
| Figura 4.9 - Identificação dos objetos de tarefa para as tarefas 1 e 2..... | 58 |
| Figura 4.10 - Interface da fase de pré-processamento (modo texto) | 59 |

| | |
|---|----|
| Figura 4.11 - Interface da fase de pré-processamento (modo diagrama). | 59 |
| Figura 4.12 - Fluxo de tarefa para a tarefa 3. | 60 |
| Figura 4.13 - Fluxo de tarefa para a tarefa 4. | 61 |
| Figura 4.14 - Identificação dos objetos de tarefa para as tarefas 3 e 4..... | 61 |
| Figura 4.15 - Interface da fase de conversão do esquema XML..... | 62 |
| Figura 4.16 - Mensagem de erro exibida quando o usuário tenta avançar para a fase seguinte sem resolver as pendências..... | 62 |
| Figura 4.17 - Fluxo de tarefa para as tarefas 5 a 9..... | 63 |
| Figura 4.18 - Identificação dos objetos de tarefa para as tarefas 5 a 9..... | 64 |
| Figura 4.19 - Interface da fase de reestruturação, validação de nomes de conceitos..... | 65 |
| Figura 4.20 - Interface da fase de reestruturação, validação de cardinalidades..... | 65 |
| Figura 4.21 - Mensagem de erro exibida se houver valores de cardinalidades incorretos..... | 66 |
| Figura 4.22 - Interface da fase de reestruturação, validação tipos de dados..... | 66 |
| Figura 4.23 - Exemplo de interação para validação de tipos de relacionamentos..... | 67 |
| Figura 4.24 - Esquema conceitual definitivo..... | 68 |
| Figura 4.25 - Fluxo de tarefa para a tarefa 10..... | 69 |
| Figura 4.26 - Fluxo de tarefa para a tarefa 11..... | 69 |
| Figura 4.27 - Fluxo de tarefa para a tarefa 12..... | 69 |
| Figura 4.28 - Identificação dos objetos de tarefa para as tarefas 10,11 e 12..... | 70 |
| Figura 4.29 - Interface da fase de unificação..... | 71 |
| Figura 4.30 - Mensagem de erro exibida se o usuário selecionar somente um arquivo..... | 71 |
| Figura 4.31 - Fluxo de tarefa para a tarefa 13. | 72 |
| Figura 4.32 - Identificação dos objetos de tarefa para a tarefa 13..... | 72 |
| Figura 4.33 - Interface da fase de inclusão de relações de herança..... | 73 |
| Figura 4.34 - Fluxo de tarefa para a tarefa 14..... | 74 |
| Figura 4.35 - Fluxo de tarefa para a tarefa 15..... | 74 |
| Figura 4.36 - Fluxo de tarefa para a tarefa 16..... | 74 |
| Figura 4.37 - Identificação dos objetos de tarefa para as tarefas 14, 15 e 16..... | 75 |
| Figura 4.38 - Interface da fase de reestruturação..... | 75 |
| Figura 4.39 - Esquema global definitivo..... | 76 |

ACRÔNIMOS

| | |
|---------------|---|
| API..... | Application Program Interface |
| BD..... | Bando de Dados |
| BInXS..... | Bottom-up Integration of XML Schematas |
| DOM..... | Document Object Model |
| DTD..... | Document Type Definition |
| HTML..... | Hiper Text Markup Language |
| IBM..... | International Business Machines |
| IDL..... | Interface Definition Language |
| IHC..... | Interação Homem-Computador |
| MCC..... | Modelo Conceitual Canônico |
| ORM/NIAM..... | Object with Roles Model/Natural Language Information Analysis Method |
| OWL..... | Web Ontology Language |
| SGBD..... | Sistema de Gerenciamento de Banco de Dados |
| XML..... | Extensible Markup Language |
| XPATH..... | XML Path Language |
| XSD..... | XML Schema Definition |
| W3C..... | World Wide Web Consortium |

RESUMO

Atualmente, tem-se discutido muito na literatura a integração de dados XML heterogêneos e diversas soluções já foram propostas. O que muitas têm em comum é a criação de um esquema de dados uniforme e integrado de representação (modelo canônico global) que permita uma abstração única das diferentes representações dos dados. Este esquema global pode servir de base para a consulta a várias fontes de dados XML de forma transparente.

Uma destas soluções é o ambiente BInXS (*Bottom-up Integration of XML*), em desenvolvimento no Grupo de Banco de Dados da UFSC (<http://www.grupobd.inf.ufsc.br>). Através do BInXS, que utiliza um processo *bottom-up*, é possível gerar um esquema conceitual global a partir da integração de esquemas XML de diferentes fontes de dados, servindo como base para a realização de consultas a estas fontes e excluindo a necessidade de se conhecer a origem e a estruturação específica dos dados XML de cada origem diferente.

Visando facilitar a interação dos usuários durante o processo de integração de esquemas XML heterogêneos, esse trabalho apresenta um projeto de interface gráfica para o ambiente BInXS. Será desenvolvido o projeto e uma possível prototipação da interface gráfica do ambiente, com base em uma metodologia de interface adequada para projeto de interfaces com o usuário.

Palavras-chave: integração de dados XML, intervenção do usuário, intenção semântica, interface gráfica, usabilidade.

ABSTRACT

Nowadays, the integration of heterogeneous XML data has been discussed in computing literature and several solutions already had been proposed. A lot of them have something in common, the creation of a uniform and integrated schemata of representation (global canonic model) that allows a unique abstraction of the different data representations. This global schema can be the base for the search to several sources of XML data in a transparent way.

One of these solutions is the BInXS environment (Bottom-up Integration of XML), in development by the Group of Data Base at UFSC (<http://www.grupobd.inf.ufsc.br>). Through the BInXS, that uses a process bottom-up, it is possible to generate global conceptual schema with the integration of XML schemata defined in different data sources, used as base for the accomplishment of queries to these sources without having to know the origin and the specific structure of each XML data.

Aiming at to facilitate the user interaction during the integration of heterogeneous XML schemas, this research presents a project of graphic interface for BInXS environment. The project will be developed and present a possible prototype of the graphic interface, based on an interface methodology adjusted to interface projects with the user.

Keywords: integration of XML data, user intervention, semantic intention, graphic interface, usability.

1 Introdução

1.1 Motivação

Nas últimas décadas, muitas mudanças causadas na ciência da computação, com a popularização da Internet, propiciaram o aparecimento de novas pesquisas na área de banco de dados. Diversas áreas do conhecimento discutem atualmente qual a melhor possibilidade de se aproveitar a grande quantidade de informações disponíveis na Internet, transformando-as em algo mais útil e com gerenciamento apropriado.

A XML (*Extensible Markup Language*) é um formato para transferência de dados semi-estruturados entre computadores e representa hoje um grande número de dados disponíveis em ambientes como a Internet ou gerados a partir de dados em banco de dados (XML, 2007). Os tipos de dados semi-estruturados não estão na forma de um texto livre, mas também não estão sujeitos às restrições impostas por uma estrutura rígida (por isso, a denominação semi-estruturado) (ABI, 2000).

O intercâmbio e a publicação de dados no formato XML através da Internet é visível em diversas transações de negócio como comércio eletrônico e na apresentação de referências bibliográficas (DBL, 2002; SIG, 1999), por exemplo.

Parte do volume de dados XML sobre um mesmo domínio, disponível na Internet é heterogêneo, pois foram definidos por pessoas e aplicações diferentes. Em função disto, cria-se a necessidade de uma organização e método de acesso simples e transparente a esses dados. Contudo, não haveria sentido algum disponibilizar esses dados sem que houvesse uma forma de integrá-los e consultá-los de forma global, pois, neste caso, os dados estariam disponíveis, mas não poderiam ser acessados de forma integrada, e nem mesmo utilizados.

No contexto de integração e consulta a dados semi-estruturados na Web existem, na literatura, diversas abordagens. Uma delas chama-se BInXS (*Bottom-up Integration of XML Schemata*) (MELLO, 2005), que apresenta uma

proposta de solução para o problema de integração de dados XML (MELLO, 2005).

1.2 Objetivos

Esse trabalho de conclusão de curso tem como principal objetivo o projeto de uma Interface Gráfica para o ambiente BInXS, que é uma necessidade urgente desse ambiente.

A prototipação dessa interface visa tornar mais amigável a interação dos usuários especialistas com este ambiente, uma vez que será através da interface que os usuários poderão executar com mais facilidade as atividades do processo de integração que requerem a intervenção dos mesmos.

Numa posterior extensão do BInXS, a interface também permitirá a formulação de consultas sobre o esquema global, que serão executadas sobre os dados XML presentes nas fontes de dados heterogêneas.

Uma vez que os sistemas com boa usabilidade impactam a tarefa no sentido da eficiência, eficácia e produtividade da interação, a principal meta é projetar uma interface humano-computador que seja útil aos seus usuários, bem como intuitiva, agradável e fácil de usar (CYBIS, 2003). Dessa maneira, os usuários poderão atingir plenamente seus objetivos com menos esforço e mais satisfação.

Serão aqui apresentadas duas abordagens para definição de modelos conceituais de interface com o usuário: The Bridge, proposta por Tom Dayton em 1996, e Usage-Centered Design, proposta por Lary Constantine e Lucy Lockwood em 1999. Por fim, escolher-se-á uma das abordagens para o projeto de interface gráfica do ambiente BInXS. Será escolhida a abordagem que consiga elucidar e validar os requisitos do usuário da melhor maneira possível, de forma prática e eficiente, para que o usuário possa alcançar seus objetivos. Assim a partir da definição dos modelos de interface poderão ser demonstrados os conceitos, opções de design, aumento do conhecimento sobre os problemas e sobre as possíveis soluções para esse ambiente.

1.3 Organização do Texto

Este trabalho é composto por mais seis capítulos, organizados da seguinte forma:

Capítulo 2: O Ambiente BInXS

Descreve o ambiente para integração semi-automática de esquemas XML, chamado Ambiente BInXS.

Capítulo 3: Técnicas de Projeto de Interface

Descreve duas técnicas de projeto de interface justificando a escolha da técnica mais adequada para a prototipação da interface para o ambiente BInXS.

Capítulo 4: Projeto da Interface Gráfica do Ambiente BInXS

Apresenta os detalhes do projeto da interface gráfica do BInXS de acordo com a técnica de projeto de interface escolhida.

Capítulo 5: Conclusões e Trabalhos Futuros

Conclui este trabalho apontando os benefícios alcançados com a solução proposta e apresentando os possíveis trabalhos que podem vir a serem desenvolvidos a partir do protótipo da interface.

2 O Ambiente BInXS

2.1 XML - Extensible Markup Language

A linguagem XML, desenvolvida pela W3C (*World Wide Web Consortium*), é um padrão de especificação de dados, consolidado mundialmente, que permite definir o conteúdo de informações a serem publicadas, bem como de mensagens a serem trocadas sobre a Internet e transportadas por protocolos como HTTP, HTTPS, FTP, SMTP, entre outros (W3C, 2007). Esta padronização possibilita que fornecedores e consumidores publiquem e se comuniquem utilizando um protocolo comum.

O conteúdo dos dados é marcado por *tags* que distinguem blocos de dados. As *tags* podem ser determinadas pelo usuário responsável pela definição da linguagem, sendo que cada *tag* determina o significado do dado que está sendo marcado.

Um arquivo no formato XML é validado por um documento DTD (*Document Type Definition*) ou XSD (*XML Schema Definition*) que determina a estrutura do arquivo XML correspondente, assim como as *tags* que podem ser utilizadas. A validação visa certificar que o documento XML obedece à estrutura definida pelo DTD (ou XSD) e possui apenas *tags* válidas (SINGH et al., 2004). Em suma um esquema XML define os elementos que podem aparecer em um documento XML; os atributos de elementos; a hierarquia permitida de elementos, ou seja, quais elementos são elementos filhos de outros; a ordem na qual os elementos filhos aparecem em uma hierarquia; a quantidade mínima e máxima de elementos filhos em uma hierarquia (cardinalidade); se um elemento deve ser vazio ou pode incluir texto puro; os tipos de dados dos elementos simples (que não possuem uma hierarquia de elementos); e os valores padrões e valores fixos para elementos e atributos.

A validação através de DTDs não é a forma mais adequada de definir um documento XML, visto que estes não expressam facilmente tipos de dados e estruturas complexas. Por outro lado, estas facilidades estão disponíveis em esquemas XSD, que é uma recomendação mais recente da W3C válidas (SINGH et al., 2004).

2.2 Descrição do Ambiente

O ambiente BInXS, já explicado na Introdução, é um ambiente para integração semi-automática de esquemas XML presentes em fontes de dados diferentes, que leva em conta todas as particularidades da definição das estruturas de dados XML (MELLO, 2005). Trata-se de um processo *bottom-up*, que a partir da definição de esquemas XML, constrói um esquema conceitual global de dados para fins de consulta a fontes de dados XML na Web.

O ambiente BInXS define um processo que ocorre em duas etapas:

- **Conversão do esquema XML:** os esquemas XML das fontes de dados são convertidos todos para uma representação canônica conceitual, com relacionamentos de associação e herança que modelam a intenção semântica dos dados. Esta etapa divide-se em 3 sub-etapas: Pré-processamento, Conversão e Reestruturação. A etapa de Pré-processamento realiza a limpeza e simplificação do esquema lógico, como exemplo, remoção de elementos semanticamente desnecessários, redefinição de estruturas aninhadas, entre outros. A etapa de Conversão converte o esquema pré-processado em um esquema conceitual preliminar, definindo relacionamentos, conceitos e atributos. A última etapa, a de Reestruturação, realiza tarefas para simplificação e validação do esquema conceitual preliminar;
- **Integração Semântica:** esta etapa é responsável pela integração propriamente dita dos esquemas conceituais representativos de cada esquema XML. Ele realiza a determinação de equivalências e resolução de conflitos dos esquemas resultantes do passo anterior, gerando um esquema conceitual global.

A figura 2.1 apresenta essas duas etapas:

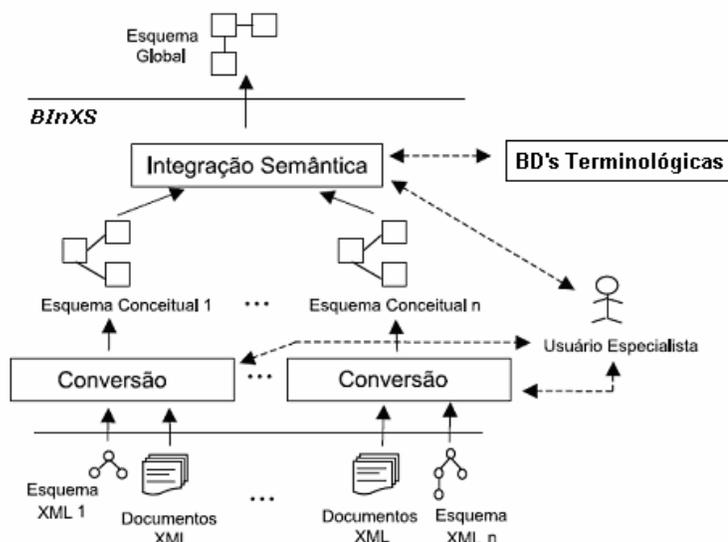


Figura 2.1 - Arquitetura do BInXS (MELLO, 2005)

Os esquemas conceituais gerados seguem o modelo ORM/NIAM e ficam armazenados em documentos OWL (WEB Ontology Language). Esses documentos OWL são utilizados pelo BInXS para descrever os esquemas XML em nível conceitual.

BInXS é um processo semi-automático, pois considera a intervenção de um usuário especialista do domínio para validar ou confirmar alternativas de resultado produzidas automaticamente.

2.3 Intervenção do Usuário

O BInXS requer a intervenção do usuário em algumas atividades realizadas nas duas etapas do seu processo de integração.

2.3.1 Etapa da Conversão do Esquema XML

Na primeira etapa, durante a conversão do esquema XML de cada fonte, são analisadas as informações dos esquemas XML e seus dados originais. A partir dessa análise, define-se um esquema conceitual. Esta etapa divide-se em três sub-etapas: Pré-processamento, Conversão e Reestruturação, de acordo com a Figura 2.2 a seguir.

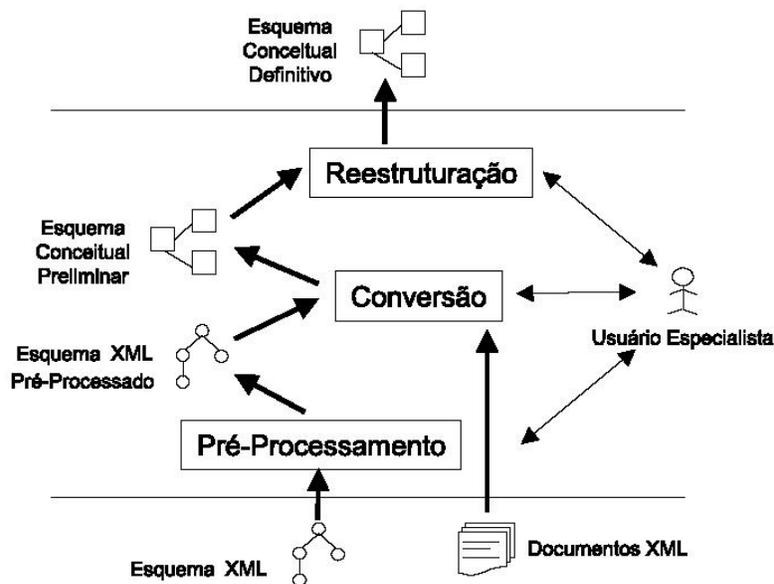


Figura 2.2 – Processo de conversão de um esquema XML

A seguir são apresentadas as atividades que requerem a intervenção do usuário dentro dessas sub-etapas.

2.3.1.1 Pré-processamento

A sub-etapa de pré-processamento tem a função de “limpar” o esquema XML, fazendo com que ele fique livre de detalhes técnicos de estruturação, os quais não contribuem para a compreensão da semântica dos seus dados.

Para efetuar essa “limpeza” são executadas algumas tarefas, tais como: substituição de entidades, remoção de elementos e atributos semanticamente desnecessários, remoção de estruturas aninhadas, tratamento de elementos compostos opcionais ou com repetição, renomeação de elementos e atributos e por fim informações de mapeamento. Dentre essas tarefas, apenas duas requerem a intervenção do usuário, as quais são detalhadas a seguir.

- **Remoção de elementos componentes e atributos semanticamente irrelevantes**

Em um esquema XML podem existir elementos que servem apenas para estruturar partes do modelo de conteúdo de um elemento composto, mas que do ponto de vista semântico são irrelevantes, ou seja, não são informações necessárias para o domínio em questão. Tais elementos devem ser removidos. Esta é uma tarefa manual pois apenas o usuário especialista tem condições de identificar esses elementos. Um elemento só é removido se não tiver atributos. Porém, atributos considerados semanticamente irrelevantes também podem ser removidos.

```

<!-- DTD 1: Livros Preferidos de um Leitor ---->

<!ELEMENT MyFavouriteBooks (Book+, Best-Referenced)>
<!ELEMENT Book (WriterList, Year, PublishingHouse?, (Technical | Fiction))>
<!ATTLIST Book Title CDATA #REQUIRED>
<!ELEMENT WriterList (Writer)+>
<!ELEMENT Writer (Name, EMail, EMail?)>
<!ATTLIST Writer Style CDATA (science | arts | romance | horror | drama
    | comedy)>
    PreferenceRating CDATA>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT EMail (#PCDATA)>
<!ELEMENT Year (#PCDATA)>
<!ELEMENT PublishingHouse (#PCDATA)>
<!ELEMENT Technical EMPTY>
<!ELEMENT Fiction EMPTY>
<!ELEMENT Best-Referenced ANY>

```

Figura 2.3 – Esquema XML (DTD) de um documento XML com livros de preferência de um leitor

A Figura 2.3 mostra um esquema XML (uma DTD) e a Figura 2.4 exemplifica a remoção do elemento **Writerlist** de **Book**, considerado semanticamente irrelevante.

```

<!ELEMENT Book (WriterList, Year, PublishingHouse, (Technical | Fiction))>
    <!ELEMENT WriterList (Writer)+>
    ↓ remoção do elemento WriterList
<!ELEMENT Book ((Writer)+, Year, PublishingHouse, (Technical | Fiction))>

```

Figura 2.4 – Remoção de elementos componentes e atributos sem semântica

Writerlist apresenta cardinalidade (1,1) em **Book** e cardinalidade (1, N) no seu modelo de conteúdo. Sendo assim, a cardinalidade resultante da inclusão de **Writer** em **Book** é (1, N).

Observa-se, então, que um elemento composto *E* que possui apenas um elemento componente é um forte candidato a ser removido, pois mantém

separadamente apenas um sub-elemento que poderia ser definido diretamente nos elementos compostos que tem *E* como sub-elemento.

- **Alteração de nomes de elementos e atributos**

Esta tarefa tem por objetivo definir nomes mais adequados para elementos e atributos do esquema XML, pois certos esquemas XML apresentam elementos e atributos cujos nomes que não contribuem para compreensão da intenção semântica, ou então, estão inseridos no contexto do domínio, mas agregados a particularidades do seu criador.

```
<!-- DTD 3: Curriculum Vitae de Publicações Científicas -->
<!ELEMENT MyCurriculumVitae (PersonalData, Publication+)>
<!ELEMENT PersonalData (Name, Sex, Address, Address?, (University |
ResearchInstitute | Business | Office), Occupation)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Sex (#PCDATA)>
<!ELEMENT Address (#PCDATA)>
<!ELEMENT University (#PCDATA)>
<!ELEMENT ResearchInstitute (#PCDATA)>
<!ELEMENT Business (#PCDATA)>
<!ELEMENT Office (#PCDATA)>
<!ELEMENT Occupation (Teacher | Researcher | Student | Developer
| (Teacher, Student))>
<!ELEMENT Teacher (ResearchInterest+, Project*)>
<ATTLIST Teacher rank CDATA (BsC, MsC, PhD) SSN ID #REQUIRED>
<!ELEMENT ResearchInterest (#PCDATA)>
<!ELEMENT Project (#PCDATA)>
<!ELEMENT Researcher (Project, Function)+>
<!ELEMENT Function (#PCDATA)>
<!ELEMENT Student (Course)>
<ATTLIST Student level CDATA (BsC, MsC, PhD) advisor IDREF>
<!ELEMENT Developer (Function, Platform+)>
<!ELEMENT Platform (#PCDATA)>
<!ELEMENT Publication (Book | Proceedings)>
<!ELEMENT Book (Title, Year, Pages, Publisher)>
<ATTLIST Book ISBN ID #REQUIRED>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Year (#PCDATA)>
<!ELEMENT Pages (#PCDATA)>
<!ELEMENT Publisher (#PCDATA)>
<!ELEMENT Article (#PCDATA)>
<!ELEMENT Proceedings (Conference, Year, Article)>
<!ELEMENT Conference (#PCDATA)>
```

Figura 2.5 – DTD que esquematiza publicações científicas de autores

Um exemplo, mostrado na Figura 2.5, é o elemento **MyCurriculumVitae**, o qual se refere às informações de autores. Assim sendo, para esse elemento valeria a seguinte renomeação:

MyCurriculumVitae => Author

Nomes de elementos virtuais devem necessariamente ser renomeados pelo usuário, pois não carregam semântica. Um exemplo de possível renomeação para elementos virtuais, de acordo com a Figura 2.5, seria:

ResearcherGroup1 => Job
OccupationGroup1 => Teacher-Student

2.3.1.2 Conversão

Essa sub-etapa demonstra um esquema XML em um esquema conceitual. Ela é composta por um processo baseado em um conjunto de regras de conversão, os quais tratam três tipos de definições de um esquema XML: definição de elementos, definição de atributos e definição de elementos componentes.

O princípio básico é que um elemento *E* composto apenas por elementos e/ou atributos torna-se um conceito não-léxico (quadrado contínuo). Esses conceitos não-léxicos possuem relacionamentos de herança (representados por uma seta do conceito especializado para o genérico) ou associação (representados por uma linha) com conceitos correspondentes aos seus sub-elementos e/ou atributos. Caso o relacionamento seja de associação há restrições de cardinalidade que devem ser definidas em ambas as direções do relacionamento. Elementos com um conteúdo texto (#PCDATA) ou atributos tornam-se conceitos léxicos (quadrados tracejados) que possuem um tipo de dado (string, integer, date etc). Quando há uma escolha na definição de *E*, uma disjunção é definida para os relacionamentos envolvidos.

A seguir são detalhadas as regras que necessitam da intervenção do usuário na sub-etapa de conversão.

- **Conversão de Elemento Componente com Repetição**

Essa é uma das regras de conversão, que trata elementos componentes com repetição e que é semi-automática. Exige-se intervenção do usuário para definir as intenções semânticas de cada ocorrência de um elemento componente.

Para cada grupo de ocorrências com a mesma intenção semântica é definido um relacionamento nomeado ou com papéis ou ambos, a critério do usuário. Se todas as ocorrências têm o mesmo significado, apenas um relacionamento de associação não nomeado é definido.

<!ELEMENT Writer (Name, EMail, EMail?)> ⇒ <Writer, EMail, (1,2), (1,1)> ∈ A (DTD1)
 <!ELEMENT Author (Name, Sex, Address, Address?, ...)> ⇒
 <Author, Address, (1,1), (1,N), WorkAddress> ∈ A
 <Author, Address, (0,1), (1,N), HomeAddress> ∈ A (DTD3)

Figura 2.6– Conversão de elemento componente com repetição

No exemplo da primeira linha da Figura 2.6, assume-se que ambas as ocorrências de **Email** são meramente endereços eletrônicos para correspondência, podendo um autor ter até dois destes endereços. Já no exemplo da segunda linha, pode-se assumir que cada ocorrência de endereço tem um significado diferente: a primeira diz respeito ao endereço profissional do autor e a segunda, opcional, é o seu endereço residencial. Na Figura 2.7 a seguir pode-se observar um esquema conceitual resultante, gerado após a intervenção do usuário.

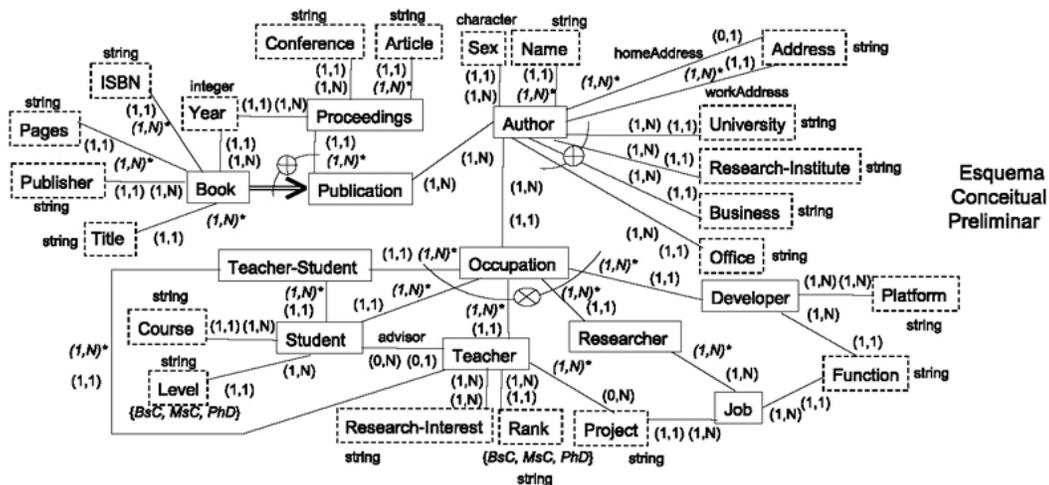


Figura 2.7 – Exemplo de esquema conceitual resultante após a intervenção do usuário na regra de conversão de elementos componente com repetição

- **Conversão de Auto-Relacionamentos**

Outra regra de conversão, chamada conversão de elemento componente nomeado auto-relacionamento, também é semi-automática e trata de auto-relacionamentos. Da mesma forma que na modelagem conceitual tradicional de banco de dados, auto-relacionamentos necessitam que a sua intenção semântica esteja explícita para permitir a sua compreensão.

Essa intenção é descrita através da descrição dos papéis de cada conceito participante ou da descrição de ambos (nome e papéis). A definição destas intenções semânticas requer a intervenção do usuário.

```

<! ----- DTD 2: Acervo de Periódicos da Editora ComputerScience ----->
<!ENTITY PublicationType "Proceedings | Journal">
<!ELEMENT ComputerScienceInclLiterature (Publication+)>
<!ELEMENT Publication (Title, Year, (%PublicationType;))*>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Year (#PCDATA)>
<!ELEMENT Proceedings (Forum, Paper+, Schedule)>
<!ELEMENT Forum (#PCDATA)>
<!ATTLIST Forum ID CDATA venue CDATA #REQUIRED>
<!ELEMENT Paper (Title, Author+, ReferencedPapers?)*>
<!ELEMENT ReferencedPapers(Paper*)>
<!ELEMENT Author (Name, Address?, (University+ | Enterprise), Office?)>
<!ELEMENT University (#PCDATA)>
<!ELEMENT Enterprise (#PCDATA)>
<!ELEMENT Office (#PCDATA)>
<!ELEMENT Address ((Street, Number)?, City, Country)>
<!ELEMENT Street (#PCDATA)>
<!ELEMENT Number (#PCDATA)>
<!ELEMENT City (#PCDATA)>
<!ELEMENT Country (#PCDATA)>
<!ELEMENT Schedule (#PCDATA | Session | Talk)*>
<!ELEMENT Session (#PCDATA)>
<!ELEMENT Journal (Name, Volume, Number, Paper+)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Volume (#PCDATA)>

```

Figura 2.8 – DTD que considera os periódicos publicados por uma editora

Baseado na Figura 2.8, tem-se o exemplo de esquema conceitual apresentado na Figura 2.9.

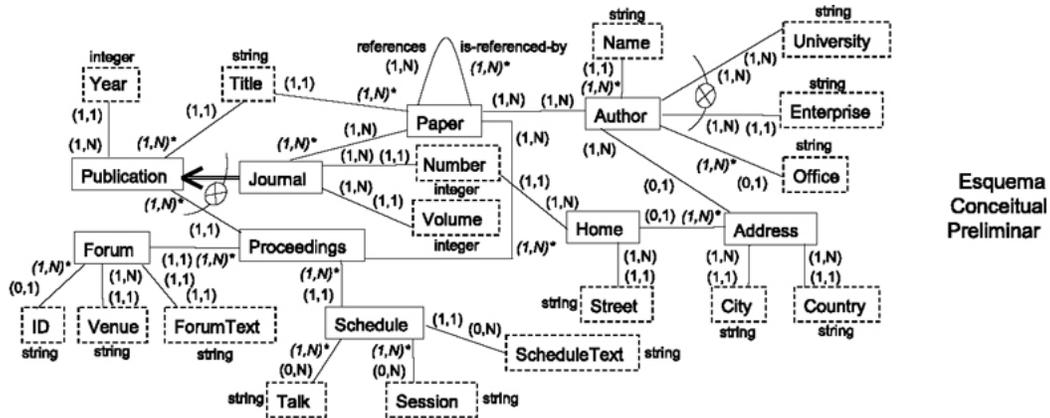


Figura 2.9 – Exemplo de esquema conceitual resultante após a intervenção do usuário na regra de conversão de auto-relacionamentos

Nesse exemplo, os papéis definem a intenção semântica do relacionamento, indicando que um artigo tanto pode estabelecer uma referencia como ser referenciado por outros artigos.

2.3.1.3 Reestruturação

Um esquema conceitual preliminar deve ser verificado pelo usuário antes de tornar-se um esquema definitivo, pois certos padrões e inferências automáticas, produzidas pelas regras de conversão ou pelos procedimentos de análise de documentos XML, podem não ser condizentes com o domínio em questão. Há algumas atualizações automáticas e outras manuais que devem ser realizadas para que o esquema torne-se semanticamente correto.

As validações automáticas (generalização e remoção de redundâncias) referem-se à simplificação do esquema, que acontece através da redução do número de relacionamentos. As validações manuais, que requerem a intervenção do usuário, são detalhadas a seguir.

- **Atualização de nomes de conceitos**

Nomes padrões assumidos para conceitos especiais ou derivados de elementos virtuais são artificiais e devem ser atualizados pelo usuário. Por exemplo, no esquema conceitual preliminar da Figura 2.9, o nome de conceito *ShceduleText* deveria ser atualizado para um nome mais coerente: *ShceduleDetails*.

- **Atualização das cardinalidades**

Cardinalidades inversas *defaults* devem ser validadas pelo usuário. Um exemplo para o esquema da Figura 2.9 seria a validação das cardinalidades do relacionamento *shcedule-proceedings*, atualizando-as de (1,N) para (1,1).

- **Atualização dos tipos de dados**

A atualização de tipos de dados pelo usuário ocorre quando a análise de documentos XML assume um tipo muito específico para conceito, como por exemplo, uma informação do tipo inteiro pode assumir valores reais.

- **Atualização de tipos de relacionamentos**

Um relacionamento de herança entre conceitos pode não ter sido detectado no passo da conversão através da consulta à base de dados terminológica, sendo definido como um relacionamento de associação. Da mesma forma, um relacionamento de herança assumido para dois conceitos pode não corresponder à semântica desejada para um determinado esquema XML. Nesses casos o usuário pode alterar o tipo de relacionamento.

- **Generalização de conceitos**

As validações citadas até aqui são obrigatórias, ou seja, o usuário deve fazê-las para garantir a coerência do esquema conceitual definitivo.

Além dessas, há um validação opcional que o usuário pode realizar, que é a generalização de conceitos.



Figura 2.10 - Exemplo de generalização de conceitos

Pode-se observar, na Figura 2.10, que ambos os conceitos **Manual** e **Book** apresentam relacionamentos de mesmo tipo com os conceitos **Author**, **Title** e **Year**. Dessa maneira, é possível definir um conceito **Publication** que generaliza estes relacionamentos comuns, reduzindo o número de relacionamentos do esquema.

Essa validação opcional só é permitida se os relacionamentos em comum forem do mesmo tipo e apresentarem as mesmas restrições de cardinalidade, papéis e nomes, no caso de relacionamentos de associação.

A possibilidade de generalização de conceitos é deixada a cargo do usuário pois nem sempre uma generalização é significativa do ponto de vista conceitual.

2.3.2 Etapa da Integração Semântica

Esta etapa é responsável pela integração propriamente dita de esquemas conceituais representativos de esquemas XML. Ele realiza a determinação de equivalências e resolução de conflitos dos esquemas resultantes do passo anterior, gerando um esquema conceitual global.

Esta etapa também é semi-automática, pois considera a intervenção do usuário para confirmar se dois ou mais conceitos passíveis de integração apresentam realmente a mesma intenção semântica.

Além da intervenção do usuário, utiliza-se uma base de dados terminológica para o suporte à determinação de equivalências semânticas entre conceitos.

Para essa etapa uma terminologia é adotada: **esquema local** é um esquema conceitual representativo de um esquema XML e **esquema global** é o esquema resultante da integração de esquemas locais.

Essa etapa segue as fases tradicionais da integração de esquemas de banco de dados: agrupamento de sinônimos, unificação, inclusão de relações de herança e reestruturação. No entanto, a intervenção do usuário é necessária somente nas fases de unificação, inclusão de relações de herança e reestruturação, conforme mostra a Figura 2.11 a seguir.

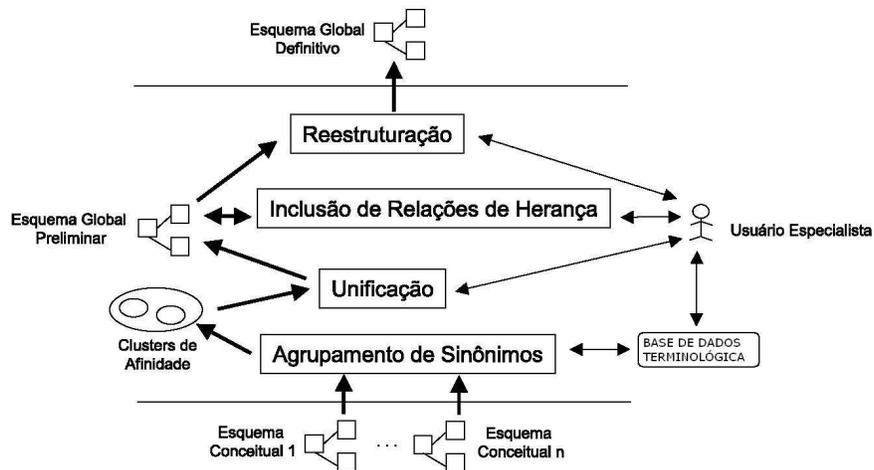


Figura 2.11 - Etapa da integração semântica

2.3.2.1 Unificação

Esta sub-etapa é o núcleo da etapa de Integração Semântica. Ela realiza a integração semântica de conceitos presentes em um mesmo cluster de afinidade, que são estruturas de dados que mantêm um grupo de conceitos semanticamente equivalentes, gerando um conceito global representativo do cluster e relacionamentos para outros conceitos globais ou clusters ainda não unificados.

A intervenção do usuário pode ser solicitada aqui para confirmar ou definir algumas correspondências semânticas entre conceitos. Um esquema global preliminar é obtido ao final dessa etapa.

Essa etapa é baseada em um conjunto de regras semi-automáticas de unificação que resolvem conflitos de heterogeneidade entre conceitos. As regras semi-automáticas são aplicadas no contexto de três casos de unificação. Cada caso de unificação trata de um determinado tipo de cluster, conforme descrito a seguir.

a) Caso LxL: Unificação Léxica

Unifica um cluster de afinidade composto por apenas conceitos léxicos, gerando um conceito global léxico. Corresponde a unificação textual presente em fontes XML.

Pelo fato de cada conceito léxico possuir um nome, um tipo de dado e opcionalmente uma enumeração de valores permitidos, conflitos de heterogeneidade relacionados a essas características devem ser resolvidos pelas regras que são descritas a seguir.

- **Regra da Unificação de Nome**

Essa regra estabelece que o nome de um conceito global é o nome que ocorre o maior numero de vezes em um conjunto de nomes de conceitos. Caso não exista um nome com maioria, o nome é definido pelo usuário, podendo ser um dos nomes do conjunto ou um sinônimo adequado a todos eles, conforme a Figura 2.12.

```
{style (1)} ⇒ style;  
{title (1), title (2), title (3)} ⇒ title;  
{writer (1), author (2), author (3)} ⇒ author;  
{enterprise (2), business (3)} ⇒ enterprise.
```

Figura 2.12 – Exemplo da regra da unificação de nome

- **Regra da Unificação de Enumeração**

Essa regra diz que caso todos os conceitos léxicos de um cluster possuam enumerações associadas, a enumeração unificada é a união dessas enumerações.

Uma validação manual do usuário é requerida para eliminar valores sinônimos redundantes. Por fim, se um ou mais conceitos não possuam enumeração, não é definida uma enumeração unificada, pois a mesma não é representativa de todos os conceitos do cluster.

A seguir é mostrado o resultado das unificações léxicas para alguns clusters de afinidade que apresentam mais de um conceito.

```
{<Publishing-House, string> (1), <Publisher, string> (3)} ⇒ <Publisher, string>;  
{<Year, integer> (1), <Year, integer> (2), <Year, integer> (3)} ⇒ <Year, integer>;  
{<Title, string> (1), <Title, string> (2), <Title, string> (3)} ⇒ <Title, string>;  
{<Nome, string> (1), <Nome, string> (2), <Nome, string> (3)} ⇒ <Nome, string>;  
{<University, string> (2), <University, string> (3)} ⇒ <University, string>;  
{<Enterprise, string> (2), <Business, string> (3)} ⇒ <Enterprise, string>;  
{<Office, string> (2), <Office, string> (3)} ⇒ <Office, string>.
```

Figura 2.13 – Resultado das unificações léxicas para os clusters de afinidade

Clusters com apenas um conceito produzem um conceito semelhante no esquema global.

b) Caso NLxNL: Unificação Não-Léxica

Unifica um cluster de afinidade composto apenas por conceitos não-léxicos. Corresponde à unificação de informação estruturada presente nas fontes XML. Gera um conceito global não-léxico representativo de todos os conceitos do cluster.

Regras são aplicadas para resolver conflitos de nome e de relacionamentos. Conflitos de nome são resolvidos pela regra da unificação de nome, já relatada.

A unificação de relacionamentos é realizada sobre pares de conceitos de cluster, ou seja, um par c_i e c_j é substituído por um conceito unificado c_{un} resultante da união dos relacionamentos c_i e c_j . Esse processo se repete até que c_{un} seja o único conceito presente no cluster, resultando no conceito global.

Na unificação iterativa dos relacionamentos de um par de conceitos é possível que um relacionamento c_i possua afinidade com um relacionamento c_j , conforme os exemplos da Figura 2.14.

Exemplo 1: $\langle \text{Book}, \text{Publishing-House}, (1,1), (1,N) \rangle \in A (1)$ e
 $\langle \text{Book}, \text{Publisher}, (1,1), (1,N) \rangle \in A (3)$
Exemplo 2: $\langle \text{Publication}, \text{Proceedings} \rangle \in H (2)$ e
 $\langle \text{Publication}, \text{Proceedings} \rangle \in H (3)$

Figura 2.14 – Exemplo de afinidade entre relacionamentos

No primeiro exemplo, um cluster de afinidade mantém os conceitos não-léxicos *Book* dos esquemas locais 1 e 3. Ambos possuem relacionamentos de associação com os conceitos *Publishing-House* e *Publisher* que se encontram em um mesmo cluster de afinidade. No segundo exemplo, tem-se um cluster de afinidade que mantém os conceitos *Publication* dos esquemas locais 2 e 3, ocorrendo afinidade entre os relacionamentos de herança destes conceitos para os conceitos afins *Proceedings*.

Um relacionamento r_i de c_i pode ter afinidade com mais de um relacionamento de c_j . Nesses casos o usuário elege um relacionamento de c_j

para ser unificado com r_i ou assume que r_i não possui afinidade com relacionamentos de c_j , ou seja, a intenção semântica de r_i não corresponde à intenção semântica dos relacionamentos de c_j . Então, r_i passa a ser um relacionamento no nível global e o usuário deve definir um nome ou papéis para ele a fim de evitar conflitos com os relacionamentos de c_j no momento da unificação. Um exemplo de um para muitos ocorre no cluster que mantém os conceitos *Author* dos esquemas 2 e 3 e seus relacionamentos com os conceitos *Address*, sendo a afinidade definida de acordo com a Figura 2.15.

$$\begin{aligned}
 &\langle \text{Author}, \text{Address}, (0,1), (1,N) \rangle \in A (2) \quad \text{e} \\
 &\left\{ \begin{array}{l} \langle \text{Author}, \text{Address}, (1,1), (1,N), \text{workAddress} \rangle \in A (3) \\ \langle \text{Author}, \text{Address}, (0,1), (1,N), \text{homeAddress} \rangle \in A (3) \end{array} \right. \\
 &\quad \downarrow \\
 &\langle \text{Author}, \text{Address}, (0,1), (1,N) \rangle (2) \equiv \langle \text{Author}, \text{Address}, (1,1), (1,N), \text{workAddress} \rangle (3)
 \end{aligned}$$

Figura 2.15 – Exemplo de um relacionamento que possui afinidade com mais de um relacionamento

No exemplo da Figura 2.15, o usuário elegeu o relacionamento *workAddress* do esquema 3 como sendo o relacionamento afim ao relacionamento do esquema 2, assumindo que ambos dizem respeito ao endereço profissional do autor.

Quando um relacionamento r_i de c_i não possui afinidade com um relacionamento de c_j , ou vice-versa, r_i é definido como um relacionamento opcional de c_{un} se $r_i \in A$, porem antes disso deve-se verificar se ele não possui afinidade com outro relacionamento r_g definido para um conceito genérico c_g de c_i . Em caso positivo, r_i fica como relacionamento obrigatório de c_{un} se for obrigatório para c_i e r_g for obrigatório também para c_g , de acordo com o exemplo ilustrado a seguir:

$$\begin{aligned}
 r_i &= \langle \text{Proceedings}, \text{Article}, (1,1), (1,1) \rangle \in A (3) \text{ e} \\
 r_g &= \langle \text{Publication}, \text{Paper}, (1,N), (1,1) \rangle \in A (2) \text{ e} \\
 &\exists \langle \text{Publication}, \text{Proceedings} \rangle \in H (2)
 \end{aligned}$$

Figura 2.16 – Exemplo de relacionamento que não possui afinidade com outro relacionamento

Neste exemplo, o cluster mantém os conceitos *Article* do esquema local 3 e *Paper* do esquema local 2, sendo que *Paper* não possui relacionamento com *Proceedings* no esquema local 2, mas *Article* sim. Assim sendo, o relacionamento r_i do conceito global referente ao cluster { *Paper* (2), *Article* (3)} com o conceito global *Proceedings* deveria ser opcional. No entanto, *Paper* se relaciona de forma obrigatória com o conceito genérico de *Proceedings*, que é *Publication* (r_g). Como *Article* também se relaciona de forma obrigatória com o *Proceedings*, r_i passa a ser um relacionamento obrigatório.

Dois relacionamentos de afinidade são tratados pela regra de unificação de relacionamentos, que considera c_{C_x} o nome do conceito global correspondente ao cluster de afinidade C_x sendo x o cluster que mantém os conceitos relacionados à c_i e c_j . Caso C_x não tenha sido ainda unificado, c_{C_x} é deixado temporariamente indefinido, devendo \underline{r} ser atualizado posteriormente com esse nome.

Quando r_i e/ou r_j possuem nome ou papéis, não é possível determinar automaticamente o resultado da unificação, devendo o usuário intervir para decidir entre:

- Gerar um relacionamento unificado \underline{R} com uma definição adequada de nome ou papéis, caso r_i e r_j possuam a mesma intenção semântica.
- Manter ambos r_i e r_j associados ao conceito unificado c_{un} r_i , caso contrário. O usuário deve definir nomes ou papéis adequados para ambos, a fim de exemplificar a intenção semântica de cada um.

A unificação de dois relacionamentos de associação que não possuem nome nem papéis pode ser relaxado na regra de unificação de relacionamentos quando os dois relacionamentos não possuem a mesma intenção semântica. Nessa situação o usuário pode desfazer o relacionamento \underline{R} e definir dois relacionamentos de associação globais com nomes e/ou papéis adequados a cada intenção semântica, conforme os exemplos abaixo:

Exemplo1: $\langle \text{Book}, \text{Publishing-House}, (0,1), (1,N) \rangle \in A (1)$ e
 $\langle \text{Book}, \text{Publisher}, (1,1), (1,N) \rangle \in A (3)$
 \downarrow
 $\langle \text{Book}, \text{Publisher}, (0,1), (1,N) \rangle$

Exemplo2: $\langle \text{Author}, \text{Address}, (0,1), (1,N) \rangle \in A (2)$ e
 $\langle \text{Author}, \text{Address}, (1,1), (1,N), \text{workAddress} \rangle \in A (3)$
 \downarrow
 $\langle \text{Author}, \text{Address}, (0,1), (1,N), \text{workAddress} \rangle$

Exemplo3: $\langle \text{Publication}, \text{Proceedings} \rangle \in H (2)$ e
 $\langle \text{Publication}, \text{Proceedings} \rangle \in H (3)$
 \downarrow
 $\langle \text{Publication}, \text{Proceedings} \rangle$

Figura 2.17 – Exemplos de relacionamentos globais de associação

No primeiro exemplo, tem-se dois relacionamentos de associação com afinidade que não possuem nome nem papéis. Ambos pertencem a conceitos *Book*, associando-se com conceitos relacionados a editoras. Considerando que ambos possuem a mesma intenção semântica, um relacionamento entre o conceito global *Book* e o conceito global *Publisher* é gerado.

No segundo exemplo, tem-se também dois relacionamentos de associação com afinidade, sendo que um dele é nomeado. O usuário assume que ambos os relacionamentos apresentam a mesma intenção semântica e um relacionamento de associação unificado nomeado com as devidas cardinalidades generalizadas é gerado. O nome *workAddress* é mantido.

No terceiro exemplo é mostrado o relacionamento de herança com afinidade, resultando em um relacionamento de herança unificado.

c) Caso NLxL: Unificação Mista

Este caso unifica um cluster de afinidade composto por conceitos léxicos e não-léxicos, gerando um conceito global não-léxico representativo dos conceitos não léxicos do cluster. Corresponde a unificação de informação textual e estruturada presente nas fontes XML.

Regras são aplicadas para resolver conflitos de heterogeneidade de nome e de relacionamentos.

Esse caso de unificação de um cluster misto C_M é realizado através da unificação de nomes de todos os conceitos em C_M (regida pela regra de unificação de nome). Depois disso, procede-se a unificação não-léxica de todos os conceitos não-léxicos de C_M , sendo gerado um conceito global não-léxico c_{un} . Após isso, para cada conceito $c_L \in C_M$ e dado um conjunto de conceitos léxicos globais S_L alcançados a partir de c_{un} , o usuário decide por uma dentre as seguintes alternativas:

- Mapeia o conceito léxico para algum conceito léxico relacionado ao conceito não léxico do mesmo cluster, que tenha afinidade com ele.
- O conceito não léxico global se especializa em dois conceitos não léxicos disjuntos, cada um com sua representação .

A seguir na Figura 2.18 um exemplo da unificação mista onde o usuário decide pelo caso de unificação.

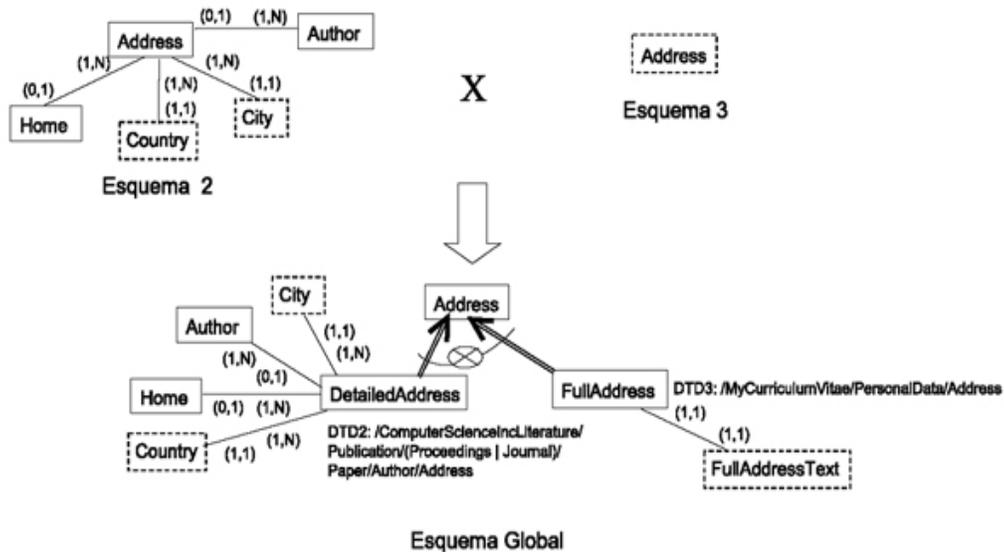


Figura 2.18 – Exemplo de unificação mista

O exemplo da Figura 2.18 diz respeito ao cluster composto pelos conceitos *Address* dos esquemas locais 2 (não-léxico) e 3 (léxico). O usuário deve decidir pelo caso de unificação pois se supõe que o conteúdo do conceito *Address* no esquema local 3 é um endereço completo que corresponde à união dos conceitos *Street*, *Number*, *City* e *Country* no esquema local 2. Assim

sendo, o usuário representa o conceito léxico *Address* do esquema local 3 como uma alternativa especializada não-léxica do conceito global *Address*, chamada *FullAddress*, e esse se torna disjunto do conceito virtual *DetailedAddress* que encapsula os demais relacionamentos de *Address*. O conteúdo de *FullAddress* é representado pelo conceito léxico associado *FullAddressText*.

Vale ressaltar que a representação global de um conceito léxico c_L de um cluster misto pode ser postergada pelo usuário caso nem todos os clusters mistos de conceitos relacionados à c_{un} tenham sido unificados. Isto ocorre quando o usuário não encontra um mapeamento um-para-um para c_L . Pode ser que c_L corresponda a um conceito léxico relacionado a um conceito cujo cluster misto ainda não foi unificado.

A partir do esquema global preliminar gerado iniciam-se tarefas de ajuste semântico, que são descritas a seguir.

2.3.2.2 Inclusão de relações de herança

Visto que o esquema global preliminar é o resultado da integração de conceitos provenientes de diversos esquemas XML, essa etapa identifica os possíveis novos relacionamentos de herança entre conceitos no esquema global preliminar com o auxílio de uma base de dados terminológica.

Estes relacionamentos, se confirmados (válidos para o domínio em questão) pelo usuário especialista, são definidos no esquema global.

Assim sendo, o usuário identifica novos relacionamentos de herança entre conceitos do esquema global e define esses relacionamentos no esquema.

Baseando-se no esquema global da apresentada anteriormente, um exemplo de relacionamento irrelevante retornado pela base de dados terminológica é <country BT value> pois o conceito de *country* é encarado no esquema global como um componente do endereço de um autor e não como uma generalização de um local de encontro de um fórum.

Já um exemplo de relacionamento considerado relevante é <fórum BT conference> pois os conceitos *fórum* e *conference* dizem respeito ao contexto de encontros científicos dos quais são produzidos *proceedings*.

Além disso, o usuário pode inclusive definir outros relacionamentos de herança além dos advindos da base de dados terminológica, regidos pela regra IncRH (inclusão de relacionamento de herança). Um exemplo ocorre na Figura 2.19, para os conceitos *Fórum* e *Conference* do esquema global preliminar, considerando o relacionamento relevante <Fórum BT Conference>.

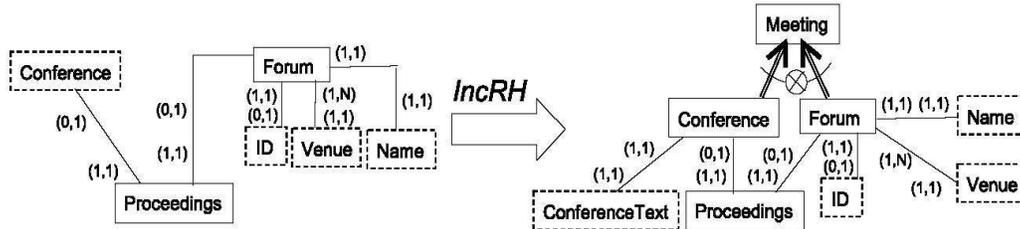


Figura 2.19 – Exemplo de inclusão de relacionamento de herança

Fórum é um conceito mais geral que *Conference* no domínio. No entanto, nem todos os conceitos associados a *Fórum*, como *ID* e *Venue*, estão associados a *Conference*. Assim sendo, não se pode definir *Conference* como uma especialização direta de *Fórum*, pois *Conference* herdaria relacionamentos que não possui nos esquemas locais em que está definido. Neste caso, define-se então um conceito genérico comum a ambos, chamado *Meeting*. O nome desse conceito é definido pelo usuário dentre um conjunto de nomes de termos mais gerais que *Fórum* e *Conference* disponibilizado por uma base de dados terminológica.

2.3.2.3 Reestruturação

Esta sub-etapa realiza ajustes automáticos e manuais no esquema global preliminar a fim de obter-se um esquema global definitivo.

As validações a serem consideradas pelo usuário referem-se às disjunções de relacionamento, definição de conceitos genéricos relevantes e modificação de nomes de conceitos especiais.

- **Tratamento de Disjunções**

Somente o usuário é capaz de identificar se disjunções de relacionamentos de um conceito não-léxico devem ser unidas ou definidas, por exemplo, conforme a intenção semântica do domínio.

Três tipos de atualizações de disjunções podem ser realizadas pelo usuário: união, inclusão ou definição.

Um exemplo de **união** de disjunções ocorre no esquema global preliminar para os relacionamentos do conceito *Publication*. A Figura 2.20 mostra esse exemplo:

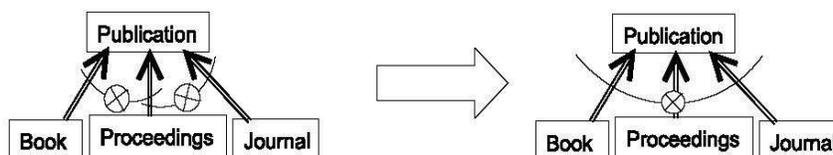


Figura 2.20 – Exemplo de união de disjunções

Podemos observar que as disjunções definidas para os relacionamentos *Book*, *Proceedings* e *Proceedings*, *Journal* são provenientes de esquemas XML diferentes. Pelo fato de serem especializações disjuntas de uma publicação, o usuário as unifica em uma única disjunção.

Com base em um exemplo fictício onde um relacionamento de herança é definido de um conceito *Publication* para um conceito *Manual*, tem-se um exemplo de **inclusão** de relacionamento em uma disjunção na Figura 2.21, considerando que *Manual* também é uma especialização disjunta de *Publication*.

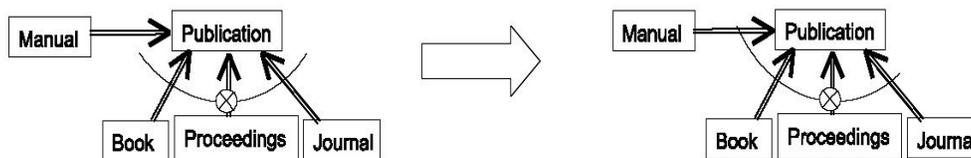


Figura 2.21 – Exemplo de inclusão de disjunções

Outro exemplo, hipotético, mostrado na Figura 2.22, ilustra uma **definição** de disjunção de relacionamentos.

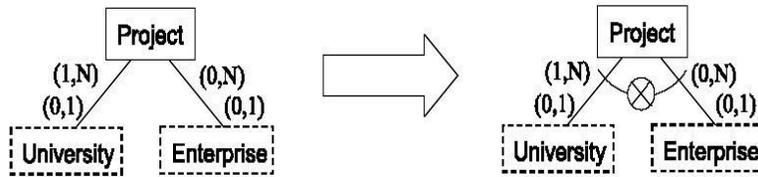


Figura 2.22 – Exemplo de definição de disjunções

Supõe-se um conceito *Project* como sendo o resultado da unificação de um conceito referente a projetos de pesquisa de universidade em um esquema local e a projetos de pesquisa de uma empresa em outro esquema local. Visto que uma ocorrência de projeto é sempre ou um projeto universitário (*University*) ou de uma empresa (*Enterprise*), uma disjunção é definida para garantir esta restrição.

- **Generalização de Conceitos**

Um conceito genérico que seja relevante para fins de consulta sobre o domínio em questão pode ser definido no esquema global.

Supondo que o conceito *Publication* não estivesse definido em nenhum esquema local, mas os conceitos *Book* e *Journal* sim, em esquemas locais diferentes, existiriam apenas os conceitos representativos de *Book* e *Journal* no esquema global. Se esses conceitos apresentassem relacionamentos com conceitos comuns (*Year* e *Author*, por exemplo), seria interessante definir um conceito *Publication* como uma generalização de ambos. Os relacionamentos comuns tornar-se-iam relacionamentos de *Publication*.

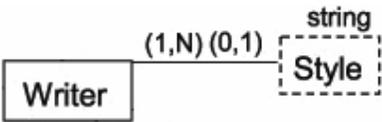
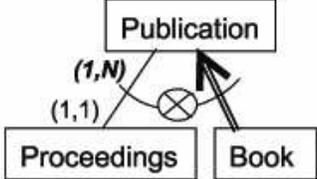
- **Modificação de nomes de conceitos especiais**

Conceitos especiais podem ser gerados durante os passos de Unificação e Inclusão de relações de herança. Assim, o usuário deve fornecer um nome mais adequado para esses conceitos.

Um exemplo seria para o conceito *FullAddressText*, que é renomeado para *Content* pois representa o conteúdo de um endereço.

Com a criação do esquema global definitivo, o processo de integração semântica de esquemas XML é finalizado, sendo o mesmo uma referência para a consulta integrada à dados de fontes XML que definem estes esquemas. Além disso, novos esquemas XML neste mesmo domínio poderão ser integrados a esse esquema global por meio da re-execução do processo, sendo ele um dos esquemas de entrada.

A tabela a seguir descreve as regras dos componentes gráficos gerados no esquema conceitual a partir de um esquema XML..

| | |
|---|---|
| <p>Um elemento composto apenas por elementos e/ou atributos torna-se um conceito não-léxico.</p> |  |
| <p>Conceitos não-léxicos possuem relacionamentos de herança ou associação com conceitos correspondentes aos seus sub-elementos e/ou atributos.</p> |  |
| <p>Um relacionamento de associação possui restrições de cardinalidade que devem ser definidas em ambas as direções.</p> |  |
| <p>Elementos com um conteúdo textual ou atributos tornam-se conceitos léxicos que possuem um tipo de dado (string, integer, date etc).</p> |  |
| <p>Quando há uma escolha na definição de um elemento, uma disjunção é definida para os relacionamentos envolvidos.</p> |  |

3 Técnicas de Projeto de Interface

A Interação homem-computador é a simulação de um ambiente real, que permite o ser humano (usuário) a interagir com o computador (hardware) através de programas (softwares). A interação também é envolvida por um outro aspecto: a extensão de usabilidade dessa mesma interação homem-computador. Esses níveis de usabilidade têm como objetivo atingir um grau de harmonia perfeito entre o usuário e o sistema.

Dessa maneira, a área de IHC se concentra em apoiar o projeto de sistemas com alta qualidade de uso, considerando as necessidades e preferências do usuário.

Pelo fato da usabilidade estar diretamente ligada ao diálogo na interface e a capacidade do software em permitir que o usuário alcance suas metas de interação com o sistema, o objetivo desse capítulo é apresentar duas abordagens bem conceituadas para projeto de interfaces com o usuário que facilitem e auxiliem na interação do usuário com o ambiente BlnXS.

3.1 Usabilidade

A usabilidade está relacionada aos estudos de ergonomia e de interação humano-computador e pode ser vista como um sinônimo de facilidade de uso.

Pela definição da International Organization for Standardization, usabilidade é a extensão na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com efetividade, eficiência e satisfação em um contexto de uso específico (ISO 1997), de acordo com a figura 3.1.

A efetividade permite que o usuário alcance os objetivos iniciais de interação, e tanto é avaliada em termos de finalização de uma tarefa quanto também em termos de qualidade do resultado obtido.

A eficiência se refere à quantidade de esforço e recursos necessários para se chegar a um determinado objetivo. Os desvios que o usuário faz durante a interação e a quantidade de erros cometidos pode servir para avaliar o nível de eficiência do site.

A terceira medida de usabilidade, a satisfação, é a mais difícil de medir e quantificar, pois, está relacionada com fatores subjetivos. De maneira geral, satisfação se refere ao nível de conforto que o usuário sente ao utilizar a interface e qual a aceitação como maneira de alcançar seus objetivos.

Segundo a norma citada acima (parte 11 da norma ISO 9241) a usabilidade pode ser especificada ou medida segundo outras perspectivas, como por exemplo:

- Facilidade de aprendizado - o usuário rapidamente consegue explorar o sistema e realizar suas tarefas;
- Facilidade de memorização - após um certo período sem utilizá-lo, o usuário não freqüente é capaz de retornar ao sistema e realizar suas tarefas sem a necessidade de reaprender como interagir com ele;
- Baixa taxa de erros - o usuário realiza suas tarefas sem maiores transtornos e é capaz de recuperar erros, caso ocorram;

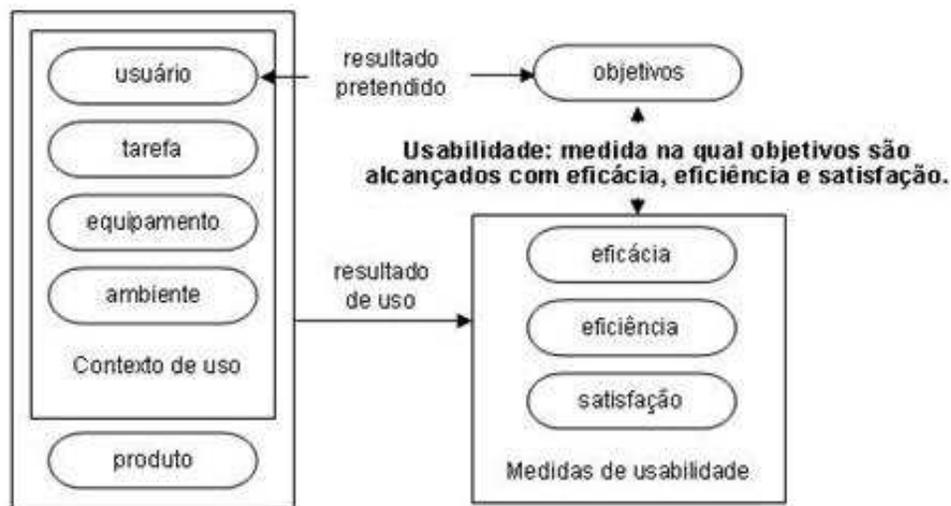


Figura 3.1 - Estrutura de usabilidade

3.2 Abordagem The Usage-Centered Design

A abordagem de projeto proposta por Constantine e Lockwood (LARRY, 1999) privilegia o “uso” do sistema e o desempenho do usuário na tarefa. É o projeto centrado no uso, no qual as decisões de projeto são baseadas em análises objetivas de modelos de usuários, de tarefas, de conteúdos e de interface. Nessa abordagem, a participação do usuário tende a ser mais

seletiva e as decisões de projeto priorizam a produtividade, eficácia e eficiência, na interação.

Baseada na concepção e projeto com o foco no uso, é uma abordagem sistemática que se baseia em transformações entre chamados modelos essenciais, os quais representam a essência das interações do usuário com o sistema. Cada um dos modelos define representações abstratas, desprovidas de detalhes de implementação. O foco dos mesmos são as intenções dos usuários e as responsabilidades do sistema.

Quatro são os modelos que fazem parte da abordagem, conforme mostra a Figura 3.2.

- Modelo de usuários
- Modelo de casos de tarefas
- Modelo de conteúdo da interface
- Maquete/protótipo da interface

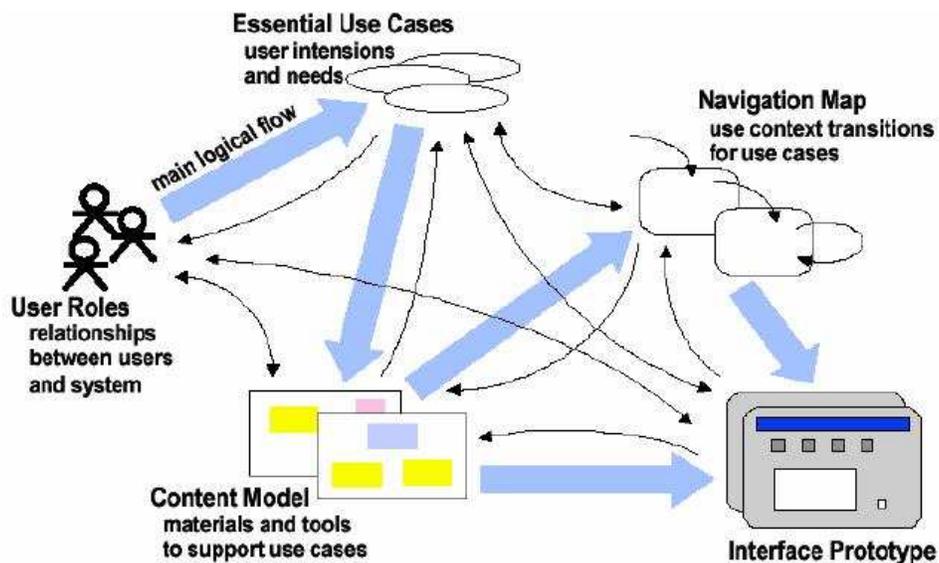


Figura 3.2 – Modelos da abordagem The Usage-Centered Design (LARRY, 1999)

Estes modelos são detalhados a seguir.

3.2.1 Modelo de Usuários

O modelo de usuários contém a descrição de atores, usuários, papéis de usuários e mapa de papéis de usuários.

Ator é todo aquele que interage com o sistema (incluindo outros sistemas). **Usuário direto** é o ser humano que interage diretamente com o sistema. **Usuário indireto** é o ser humano que interage com o sistema por meio de outros seres humanos. **Papel de usuário** (para usuários diretos) são as diferentes relações que se estabelecem entre usuários diretos e o sistema e que são caracterizadas por um conjunto específico de intenções dos usuários e de responsabilidades do sistema, ou seja, é um conjunto de necessidades, interesses, expectativas, comportamentos e responsabilidades, caracterizando um relacionamento entre classes ou tipos de usuários e o sistema. Exemplos: Administrador – Professor – Monitor – Aluno (Moodle).

Uma descrição de papel de usuário é baseada em três propriedades: contexto, características e critérios. A Figura 3.3 exemplifica um exemplo de papel de usuário.

| |
|---|
| PO3 – Papel de vendedor de bilhetes presencial |
| CONTEXTO: isolado, na cabine, de frente ao cliente, provavelmente há uma fila atrás dele; alguns vendedores precisarão de treinamento, e assume-se que todos tenham experiência; |
| CARACTERÍSTICAS: processo relativamente simples, realizado repetidas vezes; geralmente sob pressão temporal, que aumenta na medida em que a hora do espetáculo se aproxima; |
| CRITÉRIOS : interação rápida, simples e organizada. |

Figura 3.3 – Exemplo de papel de usuário

Podemos observar no exemplo acima que cada papel é descrito pelas condições típicas de contexto em que eles são desempenhados, pelas características típicas das pessoas que o desempenham e por critérios para a satisfação dessas pessoas em relação a sua interação com o programa.

Nessa abordagem, a descrição de papel assemelha-se bastante à descrição de um cenário de uso específico para um usuário específico em uma situação específica.

Os papéis estão relacionados entre si e estes relacionamentos são modelados por meio de um mapa estruturado com todos os papéis definidos para o sistema. Dessa forma, mapa de papéis de usuários é a representação das relações entre todos os papéis de usuário: semelhança, especialização e inclusão. Os papéis mais importantes devem ser definidos como papéis focais e terão atenção especial no decorrer do projeto.

3.2.2 Modelo de Casos de Tarefas

Esse modelo é baseado na descrição de casos de tarefas e também em mapa de casos de tarefas.

3.2.2.1 Casos de tarefas

São definidos como narrativas estruturadas e simplificadas (abstratas e livres de detalhes de tecnologia e de implementação) de interação realizada pelo usuário desempenhando seu papel por meio do sistema. São expressos em linguagem direta e correspondem a seqüências de associações entre intenções do usuário e responsabilidades do programa. A figura 3.4 exemplifica um caso de tarefa:

| Pegar Dinheiro | |
|----------------------|------------------------------|
| INTENÇÕES DO USUÁRIO | RESPONSABILIDADES DO SISTEMA |
| Identificar-Se | Verificar a identificação |
| Escolher uma opção | Oferecer opções de quantias |
| Pegar o dinheiro | Fornecer o dinheiro |

Figura 3.4 – Exemplo de um caso de tarefa

Compreendem os casos de uso essenciais. Representam o foco da atenção ao redor do qual a interface ou alguma parte sua será organizada. Além disso, incorporam os casos de usos mais importantes, centrais e representativos para o usuário. Quanto à natureza dos casos de tarefas eles podem ser classificados em:

- **Essencial:** aspectos essenciais da interação. Eles se referem somente àquilo que é essencial para a tarefa (ex. se a tarefa é de cadastro, renomear um item não é essencial);
- **Abstrato:** abstrai aspectos da tecnologia. Eles são abstratos, pois não trazem detalhes das soluções (ex. a responsabilidade do sistema é convidar o usuário, o que é abstrato, e não apresentar um botão "ENTRAR", o que é concreto);
- **Geral:** trata de intenções dos usuários e responsabilidades do sistema. A generalidade se refere ao fato de muitos casos de tarefa serem genéricos, como "interagir com o sistema". Eles são genéricos pois se especializam em muitos outros.

Os casos de tarefas previstos para o sistema podem ser ligados, em mapa de casos de tarefa, conforme descrito a seguir.

3.2.2.2 Mapa de casos de tarefas

Pode haver a repartição da funcionalidade total do sistema em coleção de casos de tarefa previstos para o sistema, e esses têm a possibilidade de serem ligados, em mapas de casos de tarefa, por relações de especialização, extensão, composição e afinidade. Na figura 3.5 pode-se observar um exemplo de especialização entre casos de tarefa.

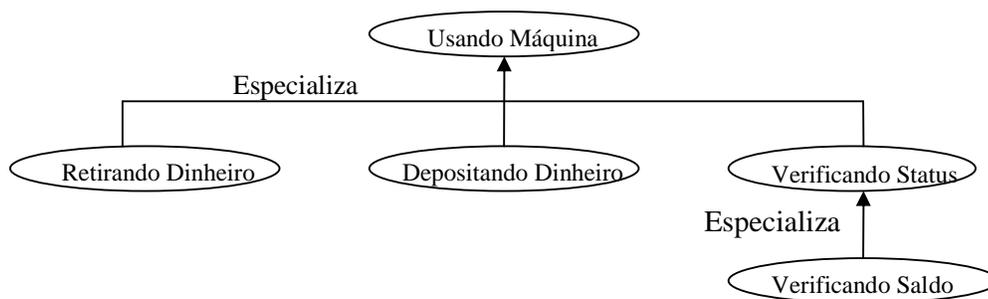


Figura 3.5 - Exemplo de especialização entre casos de tarefa

Um caso de tarefa básico pode ser estendido com base em outro, descrevendo um caso particular de interação. O resultado da relação de extensão será a interação básica permitindo situações particulares, de acordo

com a figura 3.6, onde o caso de tarefa básico “Mudar Imagem” em um documento qualquer é estendido pelo caso de tarefa particular “Procurar Imagem”.

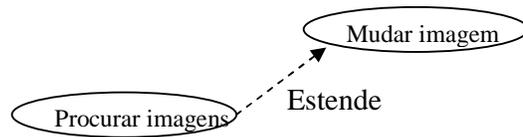


Figura 3.6 - Exemplo de extensão entre casos de tarefa

O caso de tarefa “Procurar Imagens” poderia ser útil em situações nas quais os arquivos disponíveis não sejam os que o usuário quer. Este pode ser usado como extensão do caso de tarefa “Mudar Imagem”, o que simplifica sua narrativa. A extensão de um caso de tarefa representa, assim, uma intenção alternativa e opcional, que nem sempre é realizada quando o caso de tarefa se instala.

Na relação de composição, uma interação representada por um **supercaso** é realizada por meio do emprego de interações definidas dentro de componentes de subcasos de tarefas. No exemplo que segue, o **supercaso** de uso “Fazer Primeiro Acesso” é formado a partir de “Registrar” e “Obter Acesso”.

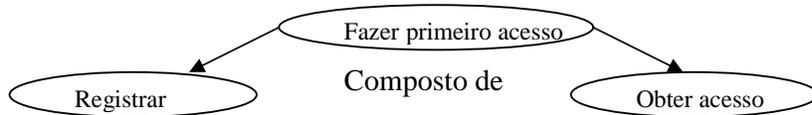


Figura 3.7 - Exemplo de composição entre casos de tarefa

Em uma modelagem sempre existe similaridade entre casos de uso que não são totalmente claras. Elas são modeladas como relações de afinidades e são especialmente úteis nas fases iniciais de projeto, quando as coisas não estão ainda bem definidas. Nos diagramas, as classes com afinidades aparecem levemente sobrepostas ou ligadas por linhas tracejadas. Em contraposição, as classes que não se assemelham aparecem distanciadas.

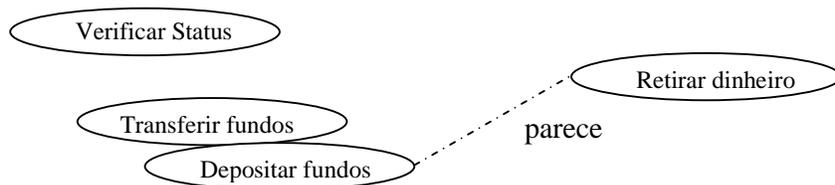


Figura 3.8 - Exemplo de afinidade entre casos de tarefa

3.2.3 Modelo de conteúdo da interface

Define uma arquitetura da interface capaz de apoiar usuários quando estiverem desempenhando seus papéis nos casos de tarefa definidos no projeto. Essa estrutura é composta de contextos de interação e mapa de navegação entre contextos.

3.2.3.1 Contextos de Interação

“Espaços” que devem estar equipados com as ferramentas e os materiais de que os usuários necessitam para a realização de suas tarefas. As regras gerais de mapeamentos são as seguintes:

- Os casos de tarefa correspondem a espaços de trabalho que serão associados a uma janela ou caixa de diálogo;
- Ferramentas representam as funções que processam dados e informações;
- Materiais representam esses dados e informações.

As descrições nessa etapa da modelagem devem se referir aos componentes de interface abstratos. Assim, os autores propõem que nessa etapa se faça a construção de protótipos abstratos, espécies de maquetes feitas de componentes abstratos.

O processo de modelagem inicia-se pela definição aproximativa de um contexto ou espaço de interação para cada caso de tarefa. Tal regra funciona bem para casos de tarefa que são bem diferentes. Entretanto, casos de tarefa parecidos podem ser realizados em um só espaço de interação.

O próximo passo envolve a análise das narrativas de casos de tarefa, linha por linha, de modo a identificar quais ferramentas e materiais terão de ser fornecidos para que o usuário seja capaz de alcançar seus objetivos na tarefa. Inicialmente são definidos os materiais necessários. Na seqüência, os casos de tarefa são revistos em busca das ferramentas necessárias. Um espaço de interação será descrito por uma lista de materiais abstratos e de ferramentas abstratas necessárias para que o usuário realize a tarefa.

Para os autores, Constantine e Lockwood, não é necessário, nem prático, tentar definir soluções de layout nessa etapa, o que será feito mais adiante, na etapa de projeto detalhado.

3.2.3.2 Mapa de Navegação

Um mapa de navegação completa a representação do modelo de conteúdos de interface, mostrando as transições entre esses espaços abstratos e os eventos que as disparam.

É representado por um diagrama de transição de estados, onde os espaços de interação são representados por retângulos e as transições são representadas por flechas conectando espaços. A figura 3.9 mostra as notações para os mapas de navegação.

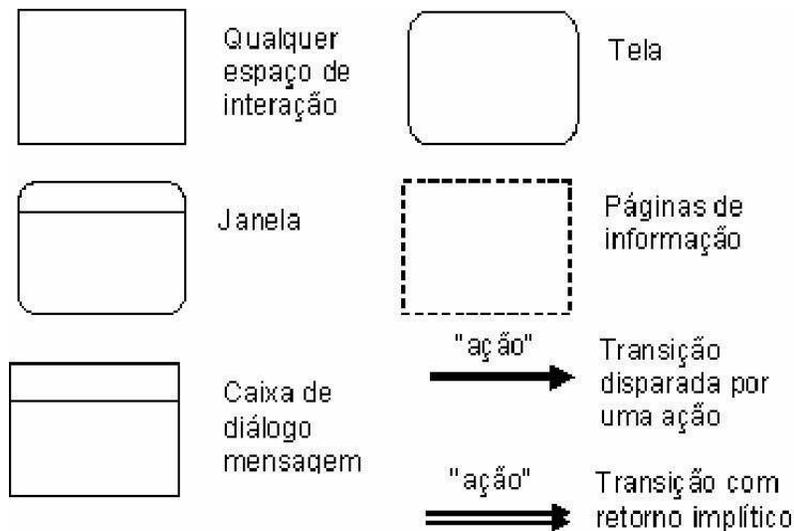


Figura 3.9 - Notações para os mapas de navegação

3.2.4 Maquete/protótipo da interface

Essa etapa envolve a construção de um protótipo/maquete da interface. Para a concepção do protótipo deve-se pensar de forma precoce e isolada antes de fazê-lo, com o objetivo de subdividir (simplificar) as tomadas de decisões.

3.2.5 Crítica da Abordagem

Pode-se concluir que é uma abordagem bastante completa, estruturada e detalhada, onde os casos de tarefa são simples e conhecidos dos projetistas. Além disso, a natureza essencial das representações ajuda a prevenir definições precoces. Por outro lado, não há uma atividade para definir uma metáfora de interface. Uma das desvantagens dessa abordagem é que ela é bastante trabalhosa.

3.3 Abordagem *The Brigde*

Tom Dayton e seus colegas da Bellcore propuseram uma abordagem para o projeto de IHC, com base em sua longa experiência no envolvimento de usuários nesse tipo de atividade.

Essa abordagem baseia-se em uma seqüência de sessões de projeto participativo, envolvendo usuários, engenheiros de usabilidade, engenheiros de software, programadores. De preferência todo o trabalho previsto deve ser realizado em um ou dois dias de reuniões em salas equipadas apenas de mesas, cadeiras, lápis, papel, cartões adesivos e desenhos de janelas pré-impressos. Nelas, projetistas e usuários irão realizar a passagem (“The Bridge”) entre os requisitos de usabilidade e o projeto de uma interface.

A abordagem procura ligar requisitos dos usuários aos objetos de interface e envolve 3 etapas principais, que serão descritas a seguir.

3.3.1 Etapa 1 – Fluxo de tarefas

Expressa os requisitos do usuário em termos de um fluxo de tarefas. As descrições nesta etapa se fazem em um alto nível de abstração, já que elas se referem aos objetivos dos usuários e não entram em detalhes sobre o sistema ou a interface.

Nessa primeira etapa os projetistas e usuários realizam a definição de um novo fluxo de trabalho para o usuário de novo sistema. Este é descrito por um fluxograma apresentando blocos para o início, os processos e decisões e o

resultado esperado do trabalho. As descrições nos blocos devem conter nomes, associados aos objetos e atributos manipulados pelos usuários, e verbos, associados às ações realizadas pelos usuários sobre tais objetos. A seguir um exemplo de fluxo de tarefa para o trabalho de registro em um hotel.

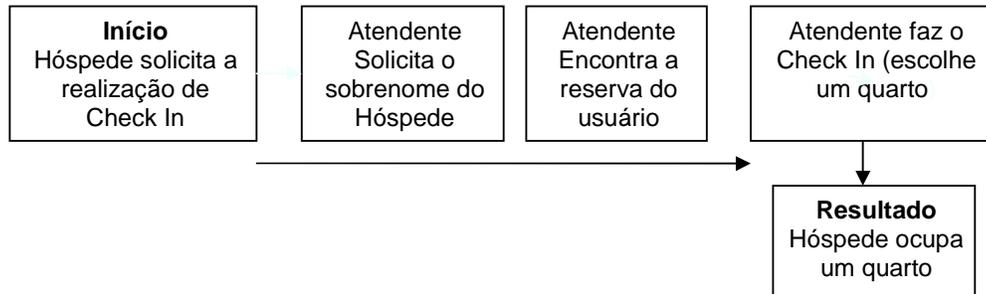


Figura 3.10 – Exemplo de fluxo de tarefa

Essa etapa pode envolver uma atividade de reengenharia, quando já existe um sistema apoiando o trabalho dos usuários. Nestes casos, o fluxo atual deve ser analisado, descrito e nele o projetista deve identificar os principais problemas e gargalos na estrutura do trabalho atual. A partir dessas informações, o processo de reengenharia se desenvolve, de forma participativa, segundo as atividades seguintes:

- Para cada tarefa problemática, definir diversas maneiras alternativas para realizá-la, de modo a eliminar os problemas identificados;
- Definir uma ordem de prioridade para as diferentes alternativas geradas, levando em consideração o desejo do usuário e a viabilidade técnica, definida pelos projetistas;
- Definir uma nova estrutura da tarefa que represente as alternativas desejadas e viáveis para a sua realização.

3.3.2 Etapa 2 – Mapear os fluxos da tarefa em objetos da tarefa

Nessa etapa deve-se identificar objetos de tarefa nos fluxos de tarefas.

Os objetos de tarefa são as entidades com as quais os usuários irão realizar uma tarefa com o sistema (independentemente de sua interface).

Uma vez validados, os fluxos de tarefas devem ser analisados visando a definição de classes de objetos de tarefa. Estas correspondem a classes de

informação por meio das quais os usuários realizam suas tarefas. Elas não referem ainda a funções do novo sistema. A definição de uma classe de objetos de tarefa envolve as seguintes decisões:

- **Identificação da classe:** essa definição se dá a partir dos substantivos nas descrições dos fluxos de tarefas.
- **Propriedades:** seu conteúdo é definido a partir dos substantivos qualificadores que aparecem nas descrições de processos do fluxo de tarefas. As propriedades de uma classe de objetos podem ser de dois tipos: atributos da classe propriamente ditos (como nome e endereço da classe hotel, por exemplo) e seus componentes, que são outras classes de objetos que nela estão contidas (por exemplo, a classe quarto). Atributos e composições serão tratados de diferentes formas no mapeamento em objetos de interface (etapa seguinte).
- **Ações:** apresentam as ações que os usuários podem realizar sobre tais objetos, usando-os para realizar suas tarefas. Correspondem comumente aos verbos que aparecem nas descrições dos fluxos de tarefas.
- **Relações de agregação entre objetos:** essa definição visa discutir composições das quais uma classe faz parte e as classes de componentes que dela fazem parte. Por definição, o Desktop é a composição de mais alto nível em todas as aplicações para o ambiente Windows.

As decisões devem ser representadas em blocos de descrição como o apresentado abaixo:

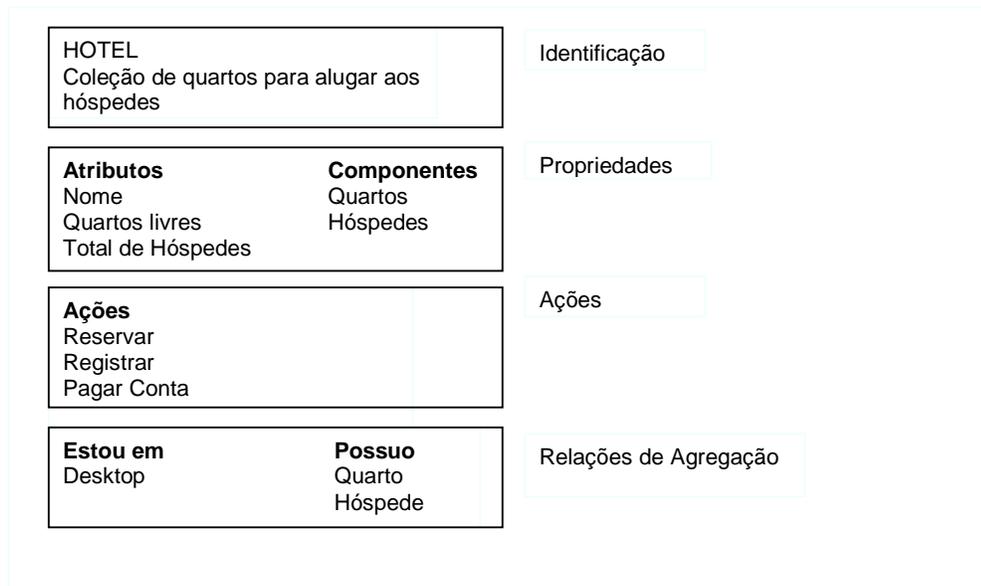


Figura 3.11 - Exemplo de classe de objeto de tarefa

Existem algumas dicas propostas pelos autores para a definição de classes de objetos de tarefa. As principais são listadas abaixo:

- Se em seu trabalho o usuário percebe unidades de dados como objetos (p. ex., catálogo telefônico), então o modelo deve considerá-los como classes de objetos.
- Se existem diversas instâncias de uma unidade de dado (p. ex., fichas de hóspedes) e, especialmente, o número de instâncias que possam existir é desconhecido, mas pode ser grande, então essa unidade de dados pode ser um objeto.
- Se os usuários criam, excluem, movem ou copiam unidades de informação em seu trabalho, tratando-as como objetos físicos, então a unidade pode ser um objeto.

Com o conjunto de classes de objetos definidos, já é possível testar sua utilidade para o usuário em relação aos fluxos de trabalho definidos na etapa anterior. Isso pode ser feito em uma reunião na qual são apresentadas aos usuários que julgam sua pertinência para a realização das tarefas.

3.3.3 Etapa 3 – Mapear objetos de tarefa em objetos de interface

Essa etapa tem como objetivo associar os objetos de tarefa com os objetos de interface, definindo-se "visões" para os objetos de tarefa e para seus atributos, bem como integrar comandos de interface.

Dessa maneira, deve-se mapear as classes de objetos de tarefas em classes de objetos gráficos de interface com o usuário, como janelas, caixa de diálogos, formulários, painéis de menu, botões etc. Um especialista em usabilidade deve participar das sessões de trabalho para ajudar a estabelecer regras para os seguintes mapeamentos:

Objeto de tarefa -> objetos de interface: Na maioria dos casos, os objetos de tarefa são mapeados como janelas primárias e secundárias, tendo a mesma identificação do objeto de tarefa. Os objetos podem também corresponder a diferentes "visões", dando origem a diferentes objetos de interface, como um item em uma lista, que, ao ser selecionado pelo usuário, aciona a apresentação de uma janela com suas informações.

Propriedades dos objetos de tarefa -> propriedades dos objetos de interface: Atributos e componentes de objetos de tarefa podem ser apresentados na forma de informação textual, figuras ou ícones nas áreas clientes das janelas de que fazem parte.

Ações de objetos de tarefa -> ações para os objetos de interface: Eles serão apresentados como opções de menu, de barras de ferramentas e de grupos de botões nas janelas principais e secundárias.

As maquetes ou os protótipos de interface definidos nesta etapa devem ter sua usabilidade testada pelos usuários participantes das sessões de projeto. Os autores da abordagem *The Bridge* propõem que detalhamentos maiores, envolvendo teclas aceleradoras, mnemônicos, bolhas de ajuda etc, sejam deixados para mais tarde, em uma etapa de projeto onde os usuários não precisam necessariamente participar.

3.3.4 Crítica da Abordagem

Essa abordagem tem um estilo popular, é rápida e prática. Realizada a partir de 3 modelagens apenas. No entanto, os fluxogramas não são adequados para a representação da dinâmica das tarefas, sendo que a passagem entre fluxo de tarefa e objetos de tarefa não é muito evidente. Além do mais, o fato de existirem poucos modelos diminui o controle sobre as transformações necessárias (passar de um modelo à outro).\

3.4 Considerações Finais do Capítulo

Pudemos observar que as duas abordagens apresentadas são bastante distintas no que se refere à própria natureza do processo de modelagem. A abordagem *The Bridge* é mais direta e econômica, propondo poucas transformações, por meio das quais o modelo conceitual é definido a partir de grandes componentes. Já a abordagem proposta por Constatntine e Lockwood, *The Usage-Centered Design*, é mais detalhada, propõe um número maior de transformações e um estilo de concepção partindo dos pequenos componentes.

4 Projeto da Interface Gráfica do Ambiente BInXS

A abordagem *The Bridge* foi escolhida para o projeto da interface do ambiente BInXS, pois se mostrou bastante objetiva, eficiente, adequada e prática para a modelagem das tarefas que requerem a intervenção do usuário durante o processo de conversão e integração de dados XML. A não escolha da abordagem *The Usage-Centered Design* deu-se pelo fato dela ser muito extensa e detalhada, o que poderia ocasionar algumas dificuldades e até mesmo a geração de resultados desnecessários para o projeto em questão.

A abordagem *The Bridge* segue as 3 etapas abaixo:

- Etapa 1: Expressar as exigências dos usuários por meio de um novo fluxo de tarefa;
- Etapa 2: Identificar objetos de tarefa nestes fluxos (entidades que são necessárias para a execução da tarefa);
- Etapa 3: Associar os objetos de tarefa com os objetos de interface.

Essas etapas são descritas adiante, nos itens 4.2 ao 4.7, conforme as tarefas já detalhadas no capítulo anterior, as quais requerem a intervenção do usuário.

4.1 Layout da Interface Gráfica

O layout da interface gráfica do BInXS foi baseado em algumas interfaces de aplicações que apresentam boa usabilidade e fácil utilização, tais como a ferramenta Jude para modelagem UML, a ferramenta BRModelo, desenvolvida no Grupo de Banco de Dados da UFSC e destinada ao ensino de modelagem de banco de dados relacional, e o Altova XMLSpy (editor XML padrão e um ambiente de desenvolvimento XML para modelagem, edição, depuração e transformação de todas as tecnologias XML).

A interface esta totalmente projetada no idioma inglês com o objetivo de atingir um público alvo maior. Posteriormente, ela poderá ser traduzida para outros idiomas.

A Figura 4.1 apresenta o layout principal da interface gráfica do ambiente BInXS.

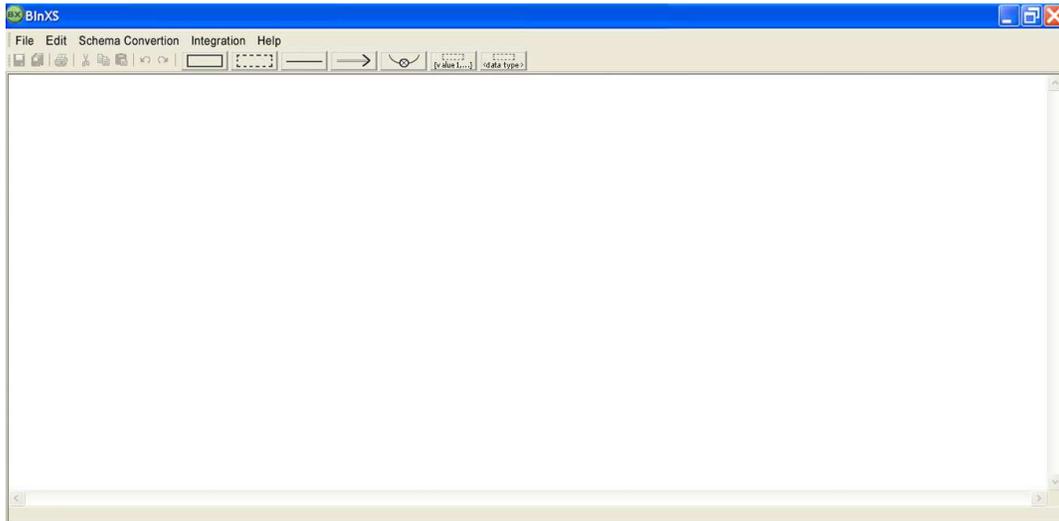


Figura 4.1 - Layout da interface do ambiente BInXS.

A barra azul superior apresenta um logo preliminar do ambiente BInXS no canto esquerdo e no canto direito as opções minimizar, restaurar e fechar.

A barra de menu, abaixo a barra superior azul, contém os seguintes itens: file, edit, schema conversion, integration e help. As Figuras 4.2, 4.3, 4.4, 4.5 e 4.6 apresentam as telas com as opções para cada um desses itens do menu, respectivamente.

Na barra abaixo do menu estão dispostos alguns ícones para agilizar e facilitar o processo, são eles: salvar, salvar como, imprimir, recortar, copiar, colar, desfazer e refazer.

Ao lado desses ícones, estão os componentes com os quais o usuário pode interagir durante algumas das suas intervenções nas tarefas que requerem a intervenção dele. Os componentes são: conceito não-léxico, conceito léxico, relacionamento de associação, relacionamento de herança, disjunção, enumeração e tipo de dado.

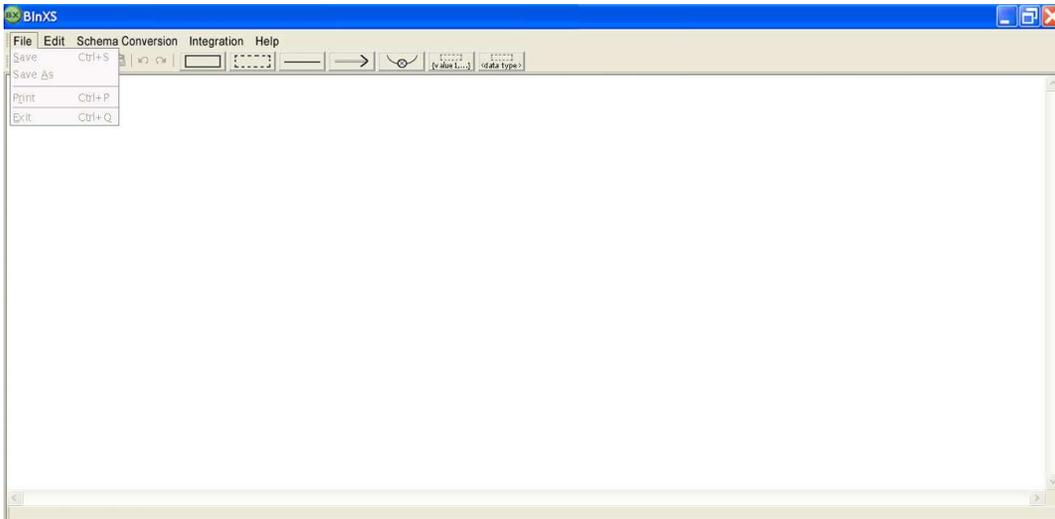


Figura 4.2 – Opções do menu File

As opções do menu File são: salvar, salvar como, imprimir e sair. Para acessar essas opções o usuário pode ainda optar pelas teclas de atalho, conforme mostra a Figura 4.2.

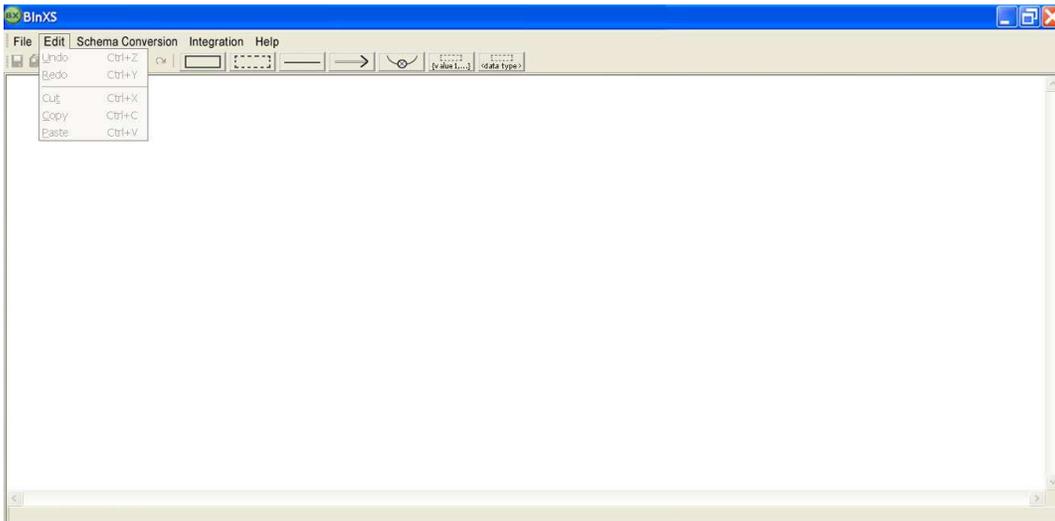


Figura 4.3 – Opções do menu Edit

As opções do menu Edit são: desfazer, refazer, recortar, copiar e colar. Essas opções também apresentam teclas de atalho, de acordo com a Figura 4.3.

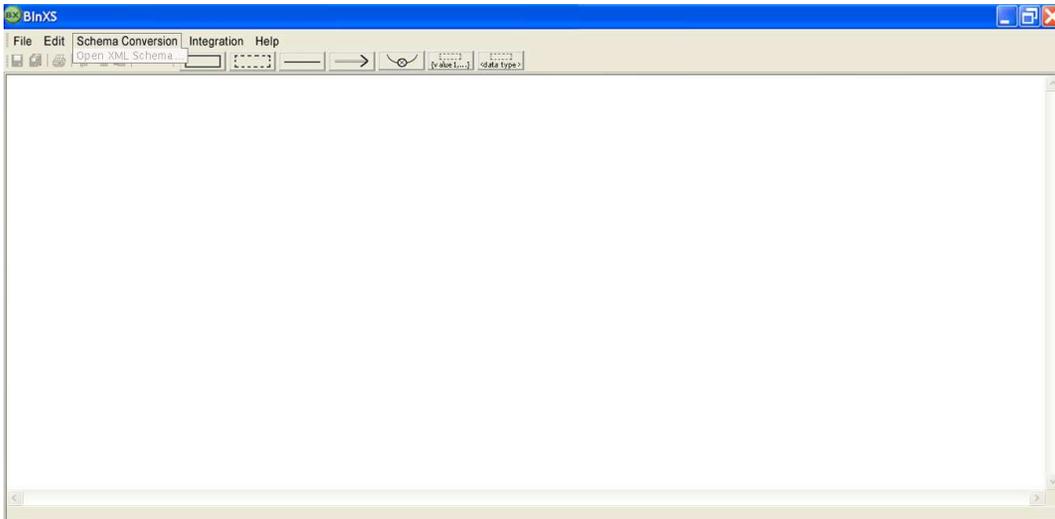


Figura 4.4 – Opções do menu Schema Conversion

O menu Schema Conversion apresenta somente uma opção, que é abrir o esquema XML, conforme a Figura 4.4. Através dessa opção é possível abrir arquivos com as seguintes extensões: .dtd, .tsd, .xdr e .xsd.

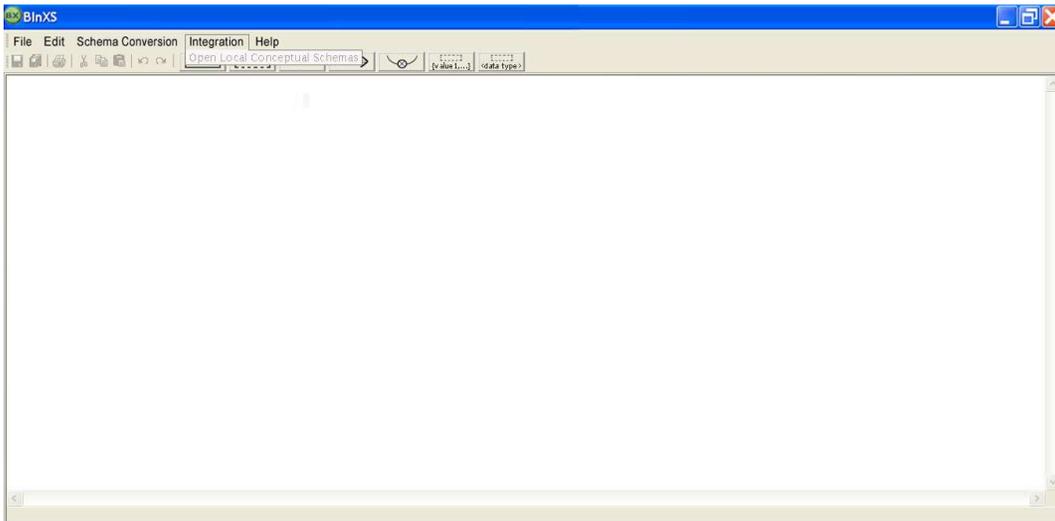


Figura 4.5 – Opções do menu Integration

O menu Integration também apresenta apenas uma opção, que é abrir os esquemas conceituais locais (os quais irão gerar o esquema global), conforme a Figura 4.5. Através dessa opção é possível abrir somente arquivos com a extensão .owl.

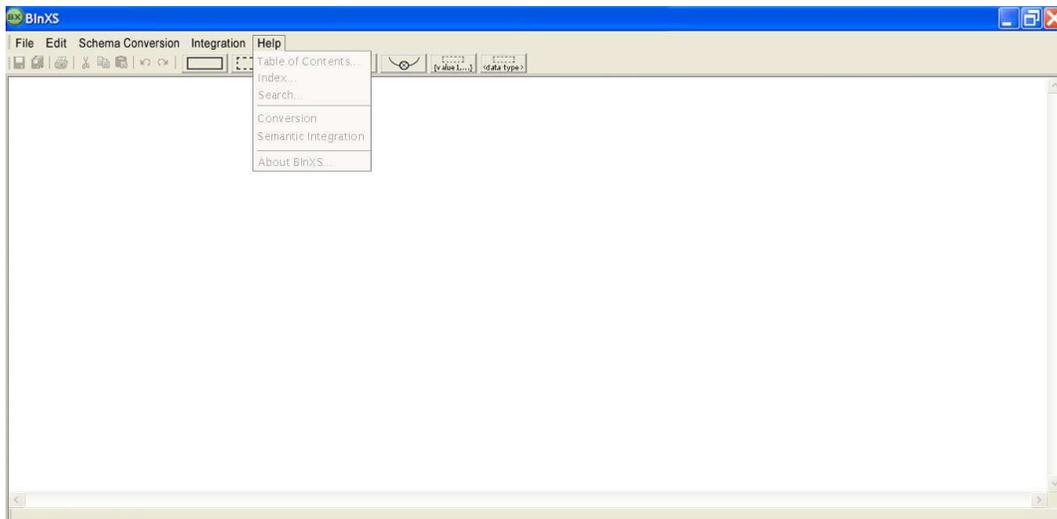


Figura 4.6 – Opções do menu Help

O menu Help apresenta seis opções: tabela de conteúdos, início, procurar, Conversão, Integração Semântica e sobre o BInXS, conforme a Figura 4.6. As cinco primeiras opções abrem a ferramenta de ajuda do BInXS. A sexta opção abre uma pequena janela contendo informações sobre a ferramenta BInXS.

4.2 Fase de Pré-processamento do Esquema

Nessa fase, o usuário pode remover do esquema XML os elementos componentes e atributos semanticamente irrelevantes e também renomear nomes de elementos e atributos.

4.2.1 Etapa 1 para as tarefas 1 e 2

A tarefa 1 refere-se à remoção dos elementos componentes e atributos sem semântica, de acordo com o fluxo de tarefa detalhado na Figura 4.7.

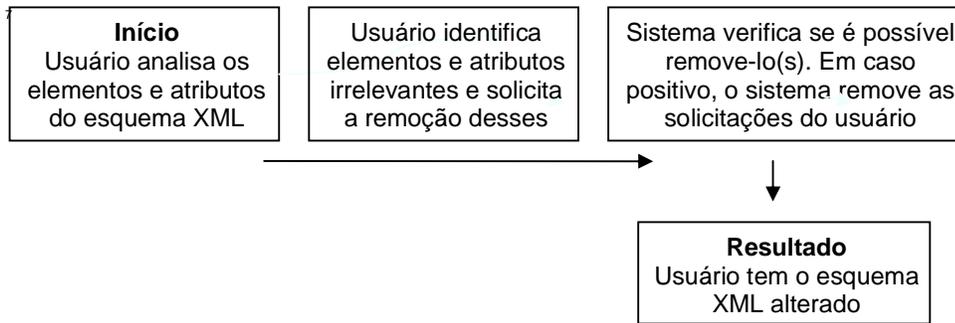


Figura 4.7 – Fluxo de tarefa para a tarefa 1.

A tarefa 2 refere-se à alteração de nomes de elementos componentes e atributos, de acordo com o fluxo de tarefa detalhado na Figura 4.8.

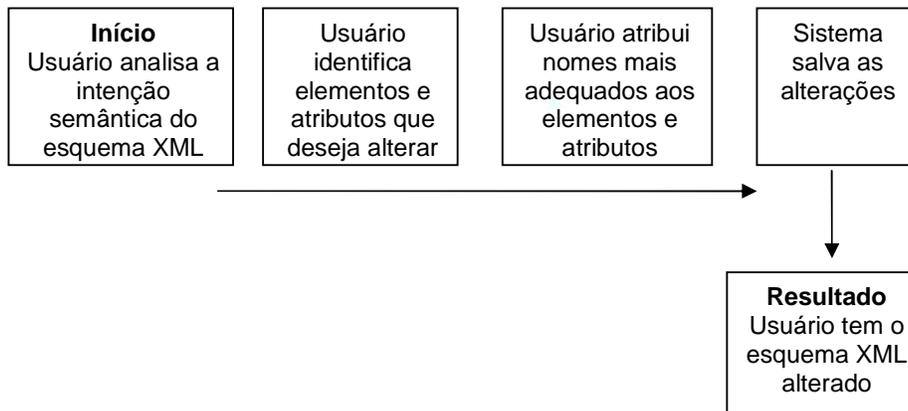


Figura 4.8 – Fluxo de tarefa para a tarefa 2.

4.2.2 Etapa 2 para as tarefas 1 e 2

A etapa 2, apresentada na Figura 4.9, é feita em conjunto para as tarefas 1 e 2, visto que o usuário pode interagir com ambas através da análise do esquema XML.

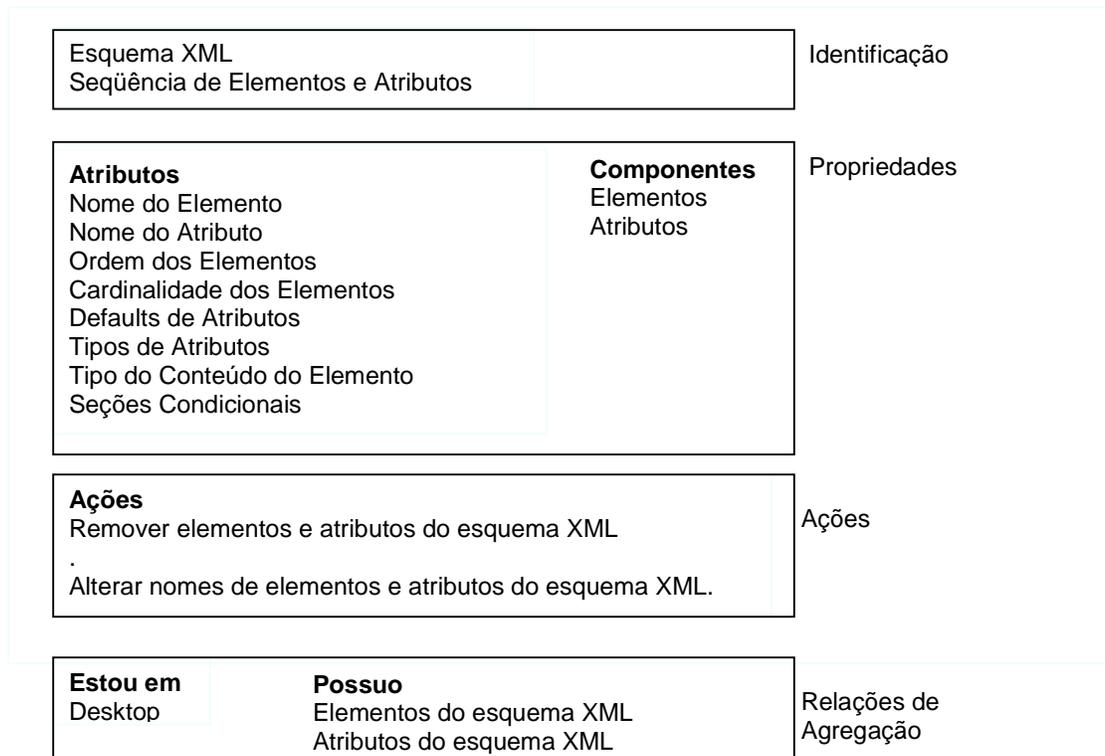


Figura 4.9 – Identificação dos objetos de tarefa para as tarefas 1 e 2.

4.2.3 Etapa 3 para as tarefas 1 e 2

Nessa etapa estão associados os objetos de tarefa (descritos no item anterior) com os objetos de interface.

Como a fase de pré-processamento é a primeira fase da conversão do esquema XML, as telas abaixo irão aparecer assim que o usuário selecionar o esquema que deseja converter, através do menu *Schema Conversion*, na barra superior do menu principal da interface gráfica do BInXS.

A interface proposta para essas duas tarefas da fase de pré-processamento é apresentada de dois modos. A primeira é o modo texto, apresentada na Figura 4.10, onde o usuário edita e/ou remove os elementos e/ou atributos que julgar necessário. Pode-se observar uma aba no lado esquerdo, que situa o usuário durante o processo de conversão do esquema XML, tornando-o mais perceptível. A etapa que estiver em curso aparece em cor destacada (no caso, em azul).

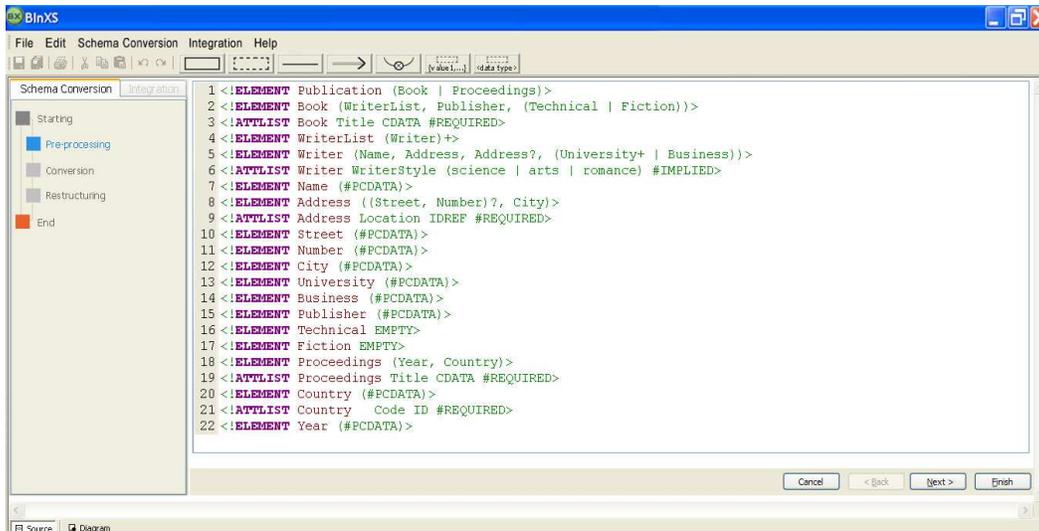


Figura 4.10 – Interface da fase de pré-processamento (modo texto).

A segunda é o modo diagrama, apresentada na Figura 4.11, que facilita a compreensão do esquema XML. Na barra inferior esquerda da interface, o usuário pode escolher o modo de exibição que preferir.

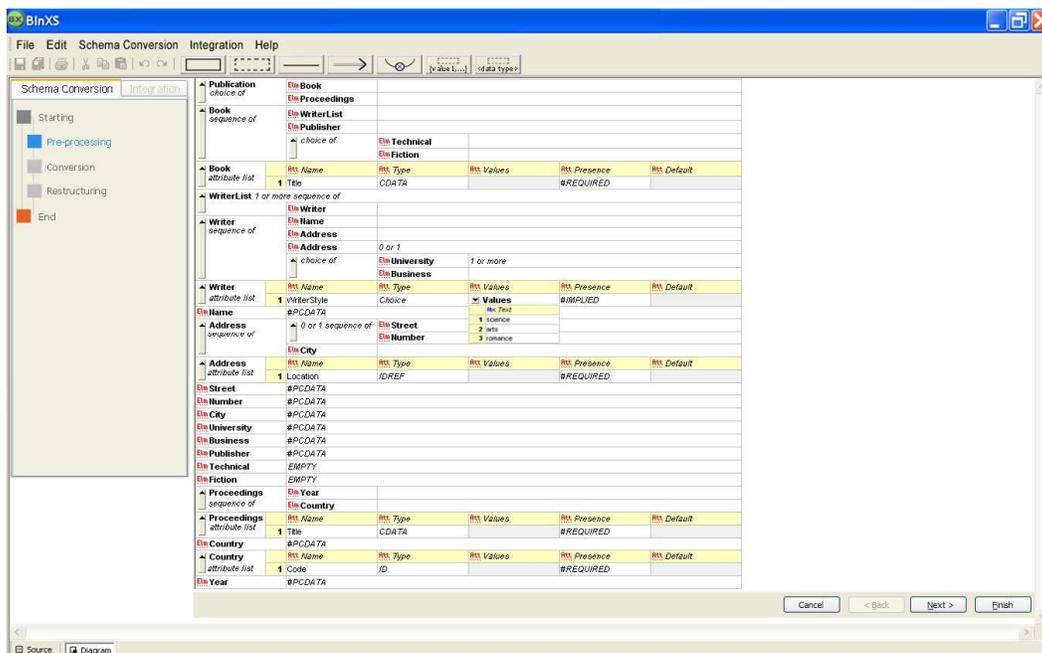


Figura 4.11 – Interface da fase de pré-processamento (modo diagrama).

Para partir para a fase seguinte o usuário deve pressionar o botão *next* na barra inferior direita. Ao pressionar este botão, o sistema, além de seguir para a próxima fase, também salva automaticamente as alterações efetuadas até o momento. Nesta mesma barra, o usuário pode cancelar a execução (ação

de fechar o arquivo) ou então finalizar a execução (ação que apresenta o esquema global definitivo).

4.3 Fase de Conversão do Esquema

Durante essa fase, o usuário deve definir, quando necessário, a intenção semântica dos elementos componentes com repetição e dos auto-relacionamentos.

4.3.1 Etapa 1 para as tarefas 3 e 4

A tarefa 3 refere-se à conversão dos elementos componentes com repetição, de acordo com o fluxo de tarefa detalhado na Figura 4.12.

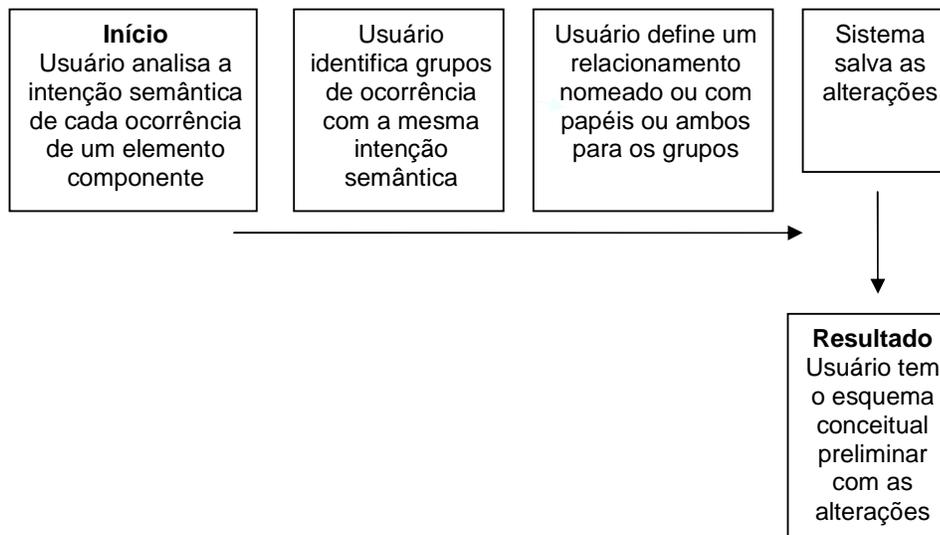


Figura 4.12 – Fluxo de tarefa para a tarefa 3.

A tarefa 4 refere-se à conversão dos elementos componentes nomeados auto-relacionamento, de acordo com o fluxo de tarefa detalhado na Figura 4.13.

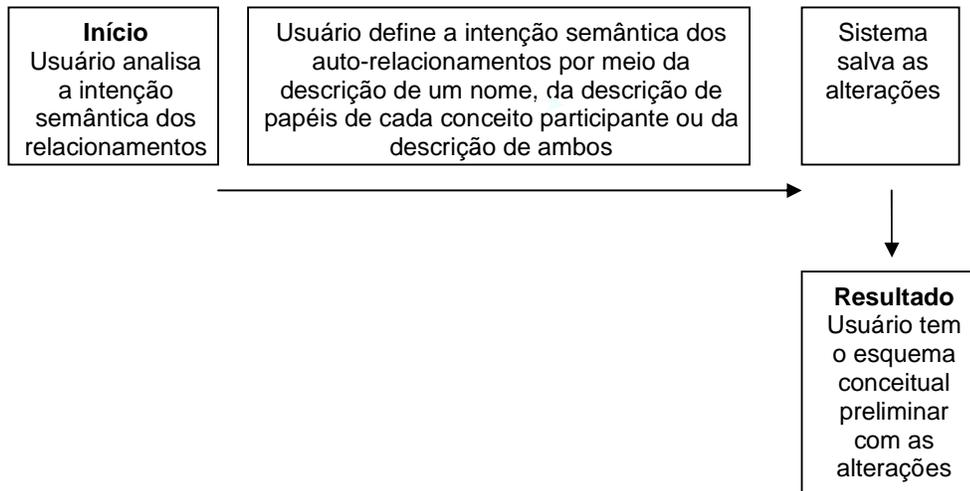


Figura 4.13 – Fluxo de tarefa para a tarefa 4.

4.3.2 Etapa 2 para as tarefas 3 e 4

A etapa 2, apresentada na Figura 4.14, é feita em conjunto para as tarefas 3 e 4, visto que o usuário pode interagir com ambas através do esquema conceitual preliminar gerado pela fase de pré-processamento.

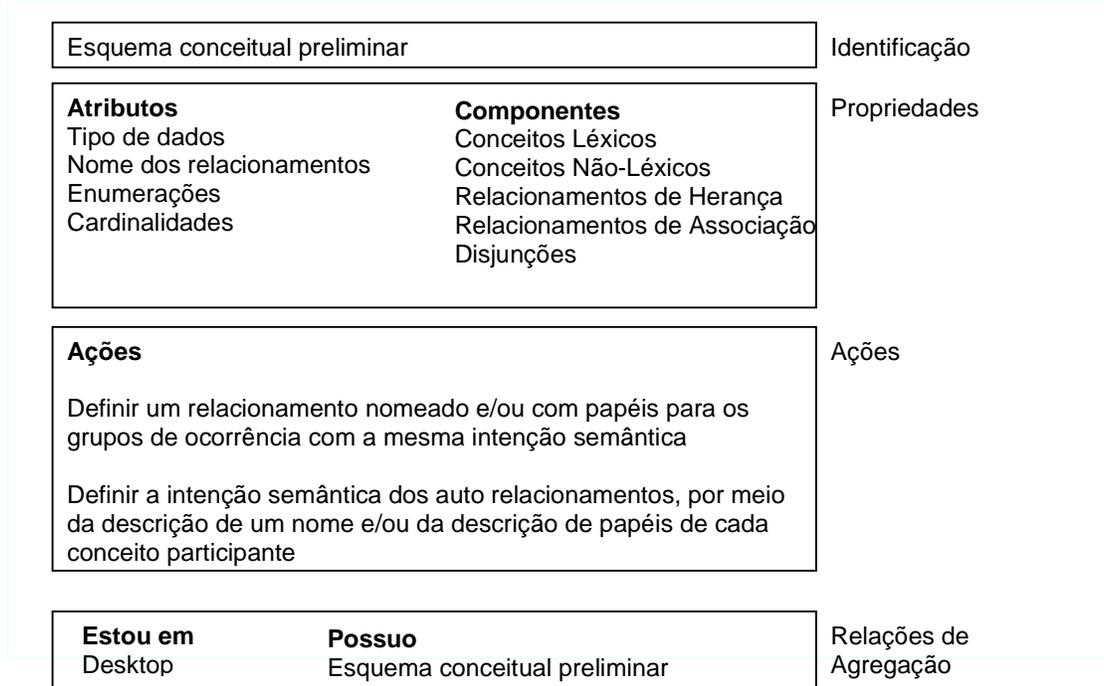


Figura 4.14 – Identificação dos objetos de tarefa para as tarefas 3 e 4.

4.3.3 Etapa 3 para as tarefas 3 e 4

Os objetos de interface para essa fase são apresentados na tela através do esquema conceitual preliminar, de acordo com as propriedades descritas no item anterior, conforme mostra a Figura 4.15.

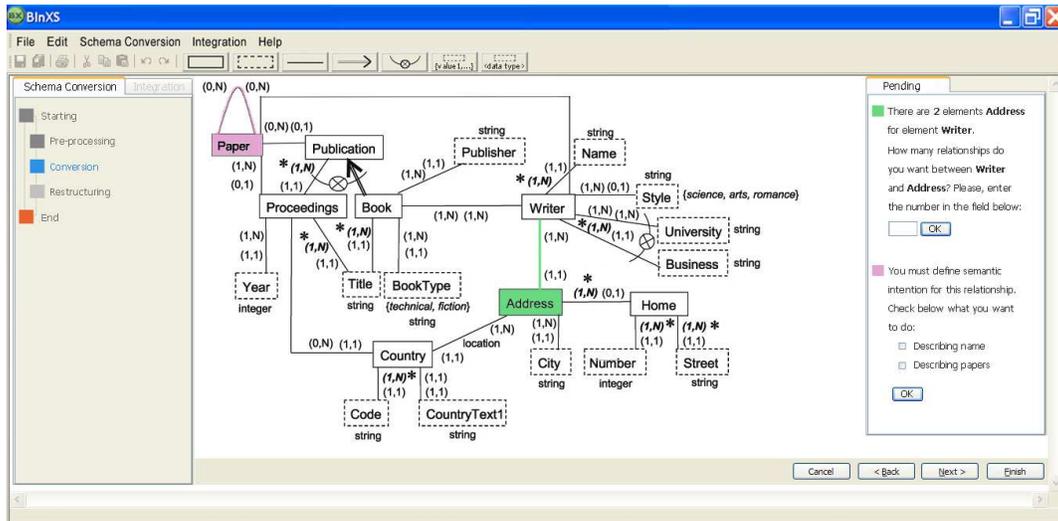


Figura 4.15 – Interface da fase de conversão do esquema XML.

A interface, durante essa fase, deve informar ao usuário os elementos que possuem pendências, destacando-os com cor diferenciada. Uma aba no lado direito é exibida contendo as informações necessárias para que o usuário possa solucionar as pendências que o sistema encontrou.

O usuário somente pode avançar para a próxima fase caso todas as pendências sejam resolvidas. Caso contrário, o sistema apresenta uma mensagem de erro, conforme a figura 4.16.



Figura 4.16 – Mensagem de erro exibida quando o usuário tenta avançar para a fase seguinte sem resolver as pendências.

4.4 Fase de Reestruturação do Esquema

Durante essa fase, o usuário deve efetuar algumas validações manuais no esquema conceitual preliminar, para que este venha a se tornar um esquema conceitual definitivo.

4.4.1 Etapa 1 para as tarefas 5, 6, 7, 8 e 9

As tarefas 5, 6, 7, 8 e 9, as quais se referem a validação do esquema conceitual preliminar, são respectivamente: atualização de nomes de conceitos, atualização de cardinalidades, atualização de tipos de dados, atualização de tipos de relacionamentos e generalização de conceitos. O fluxo de tarefa da Figura 4.17 detalha essas 5 tarefas, que são realizadas em conjunto.

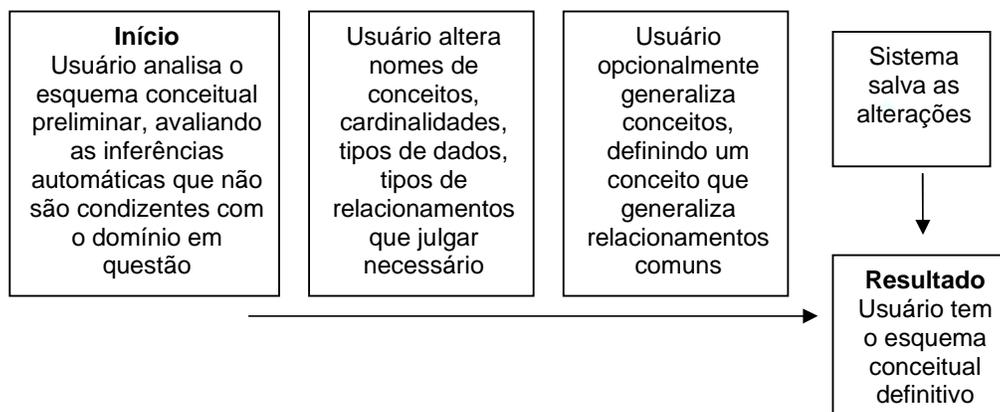


Figura 4.17 – Fluxo de tarefa para as tarefas 5 a 9.

4.4.2 Etapa 2 para as tarefas 5, 6, 7, 8 e 9

A etapa 2 é feita em conjunto para as tarefas 5 a 9, conforme apresenta a Figura 4.18, já que o usuário deve fazer as verificações na tela do esquema conceitual preliminar (gerado após a fase de conversão) para que o mesmo torne-se um esquema conceitual definitivo.

| | | |
|---|---|-----------------------|
| Esquema conceitual preliminar | | Identificação |
| Atributos Tipo de dados Nome dos relacionamentos Enumerações Cardinalidades | Componentes Conceitos Léxicos Conceitos Não-Léxicos Relacionamentos de Herança Relacionamentos de Associação Disjunções | Propriedades |
| Ações Alterar nomes de conceitos Alterar cardinalidades Alterar tipos de dados Alterar tipos de relacionamentos Generalizar conceitos (ação opcional) | | Ações |
| Estou em Desktop | Possuo Esquema conceitual preliminar | Relações de Agregação |

Figura 4.18 – Identificação dos objetos de tarefa para as tarefas 5 a 9.

4.4.3 Etapa 3 para as tarefas 5, 6, 7, 8 e 9

Nesta etapa, são apresentadas as telas para cada tarefa que o usuário deve executar durante o processo de reestruturação. O usuário pode realizá-las em uma única tela. No entanto são apresentadas várias telas, para que cada intervenção fique mais compreensível.

Nesta fase, as telas novamente apresentam o esquema conceitual preliminar, que após as modificações tornar-se-á um esquema conceitual definitivo.

A seguir, na Figura 4.19, pode-se observar como o usuário interage na tarefa 5, na qual ele pode atualizar nomes de conceitos. Essa interação pode ser através das opções que são apresentadas com o clique do botão direito do mouse ou selecionando o nome do conceito, ação esta que proporciona a ele a edição do mesmo.

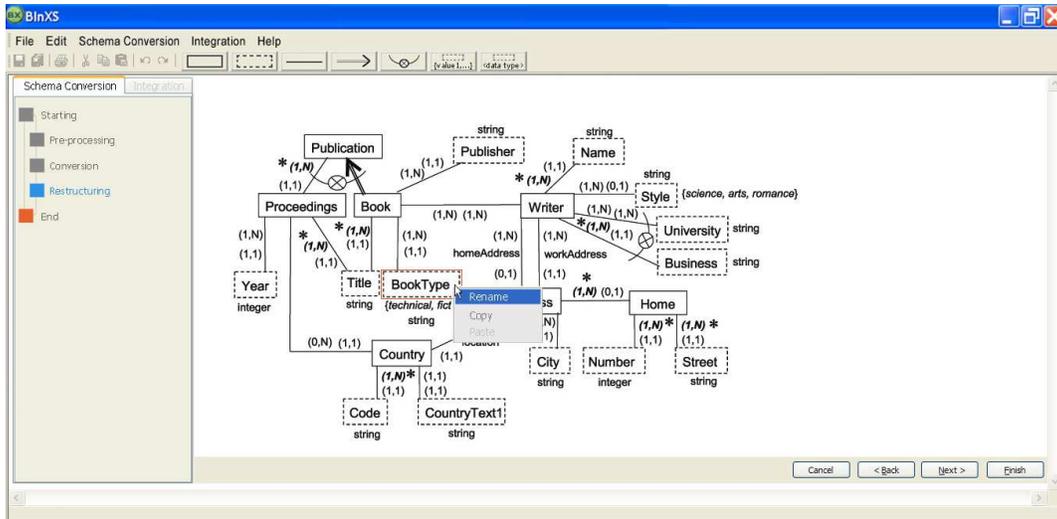


Figura 4.19 – Interface da fase de reestruturação, validação de nomes de conceitos.

Nota-se ainda que, assim que um conceito for selecionado, o mesmo aparece contornado em vermelho, para ficar claro ao usuário que ele está tratando daquele conceito naquele momento. Vale lembrar que esse contorno vermelho aparece para qualquer objeto que o usuário venha a selecionar no esquema conceitual.

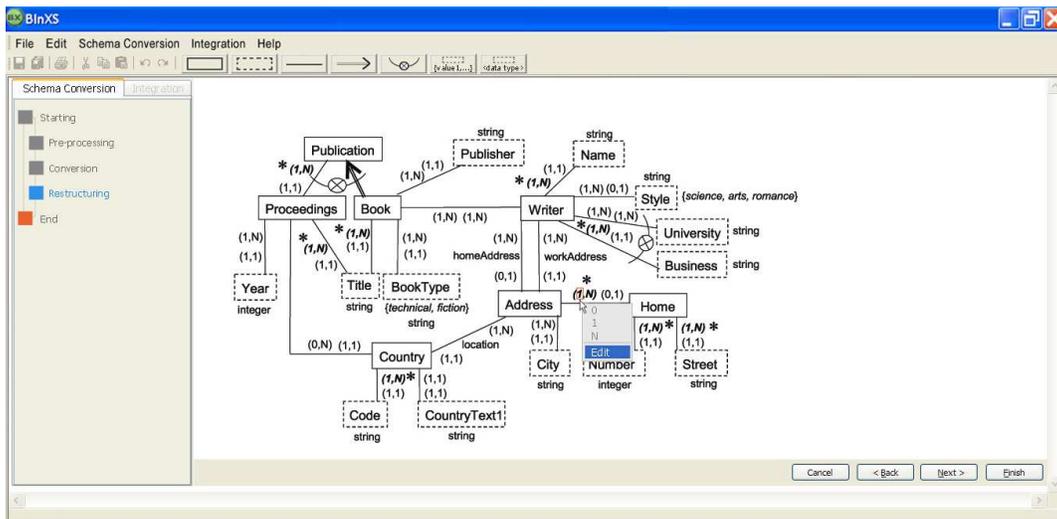


Figura 4.20 – Interface da fase de reestruturação, validação de cardinalidades

Na figura 4.20 tem-se um exemplo de como o usuário pode modificar o valor da cardinalidade a partir do clique do botão direito do mouse sobre uma dentre as opções apresentadas (0,1 ou N), ou ainda escolhendo a opção editar,

que permite a edição direta da cardinalidade selecionada (essa mesma ação ocorre se houver um clique sobre o valor).

Caso o usuário preencha os valores das cardinalidades de maneira incorreta, os quais contenham dados diferentes de números e da letra “N”, uma mensagem de erro é retornada pelo sistema assim que for pressionado o botão *next*, de acordo com a Figura 4.21.



Figura 4.21 – Mensagem de erro exibida se houver valores de cardinalidades incorretos

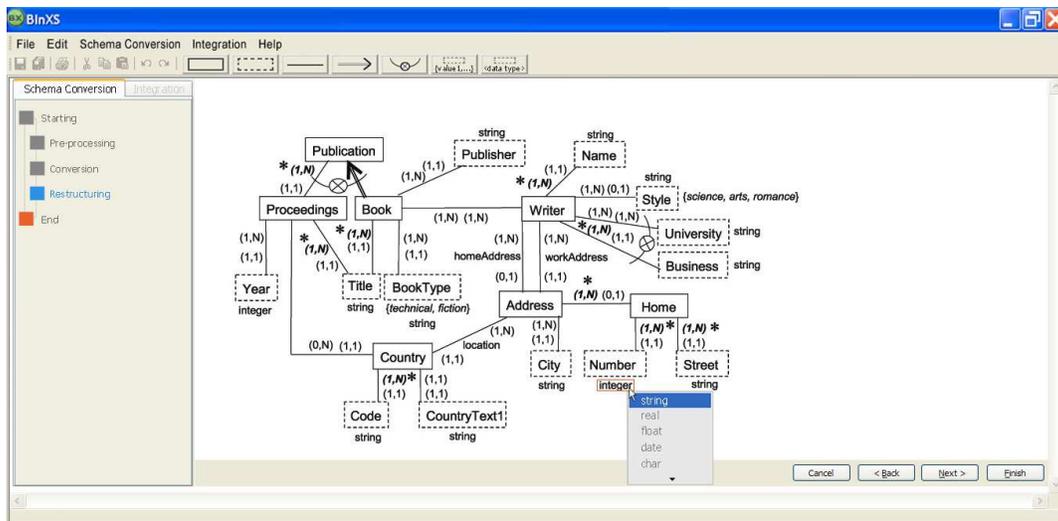


Figura 4.22 – Interface da fase de reestruturação, validação tipos de dados

A figura 4.22 exemplifica como ocorre a atualização de tipos de dados pelo usuário. Esse objeto não pode ser editado. Assim, quando o usuário clicar sobre o tipo de dado que deseja modificar, um menu *pop-up* é aberto, contendo todos os tipos de dados suportados pelo ambiente BInXS, devendo o usuário escolher o tipo que deseja. A flecha preta na parte inferior significa que há mais tipos de dados além dos apresentados, bastando o usuário passar o mouse por cima da flecha para vê-los.

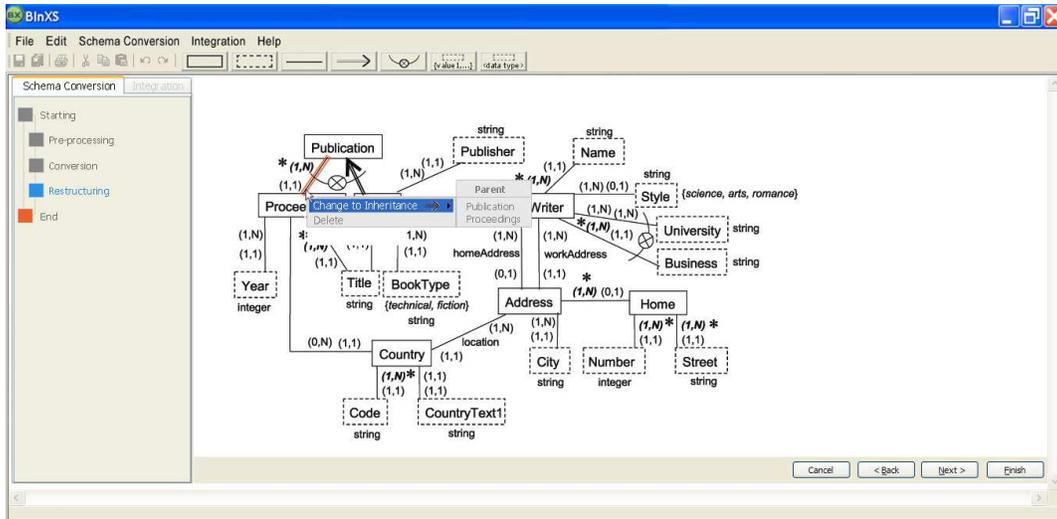


Figura 4.23 – Exemplo de interação para validação de tipos de relacionamentos.

A figura 4.23 exemplifica como ocorre a atualização de tipos de relacionamentos pelo usuário. Esse objeto pode ser removido e também incluído através dos componentes dispostos na barra superior da interface, caso necessário. Para removê-lo, basta selecioná-lo e pressionar a tecla *delete*, ou então com o botão direito do mouse, ao abrir o *pop-up*, selecionar a opção *delete*. Para incluir um novo relacionamento, basta o usuário selecioná-lo na barra de componentes e clicar sobre o conceito origem e posteriormente clicar sobre o conceito destino, devendo o mesmo sempre estar conectado a dois conceitos. Caso o usuário apenas queira mudar o tipo do relacionamento, deve-se escolher esta opção no *pop-up*. Na figura 4.23 é mostrada uma opção de troca de relacionamento, que é o relacionamento inverso àquele que foi selecionado. Caso o usuário opte por esse caminho, ele deve especificar qual é o conceito genérico (ação necessária somente para relacionamentos de herança).

A tarefa 9, onde o usuário pode generalizar conceitos, conforme exemplo da figura 2.8 apresentada no capítulo 2, ocorre através da interação do usuário com os componentes do sistema. O usuário pode adicionar ou remover conceitos, relacionamentos, disjunções, enumerações e tipos de dados.

Após as intervenções do usuário nessa etapa da reestruturação, um esquema conceitual definitivo é gerado, de acordo com a Figura 4.24, sendo finalizada a etapa de conversão do esquema.

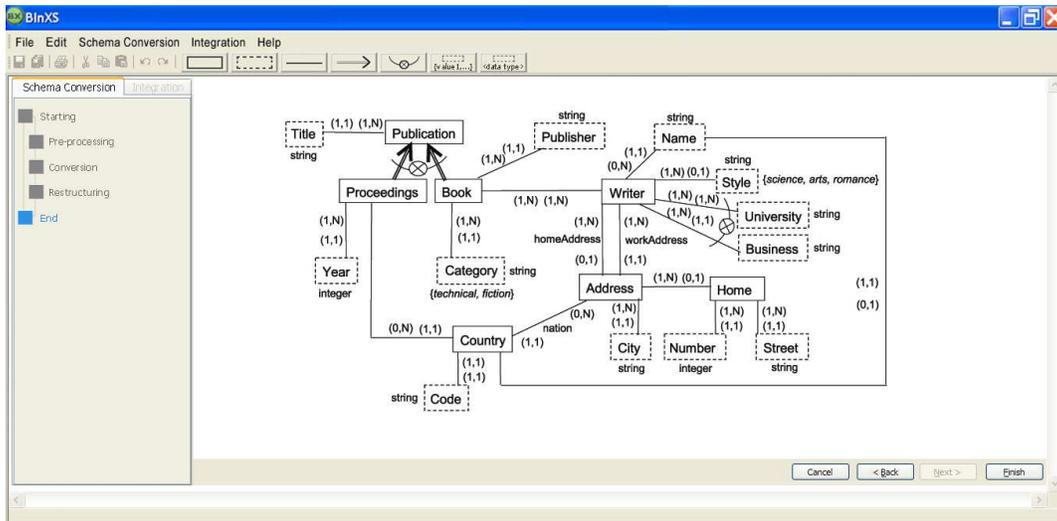


Figura 4.24 - Esquema conceitual definitivo.

4.5 Fase da Unificação

As fases daqui para adiante pertencem à etapa de Integração Semântica. Durante essa fase de unificação o usuário resolve alguns conflitos de heterogeneidade entre conceitos através de algumas regras semi-automáticas que são detalhadas a seguir.

4.5.1 Etapa 1 para as tarefas 10, 11 e 12

A tarefa 10 refere-se à Regra da Unificação de Nome, de acordo com o fluxo de tarefa detalhado na Figura 4.25.

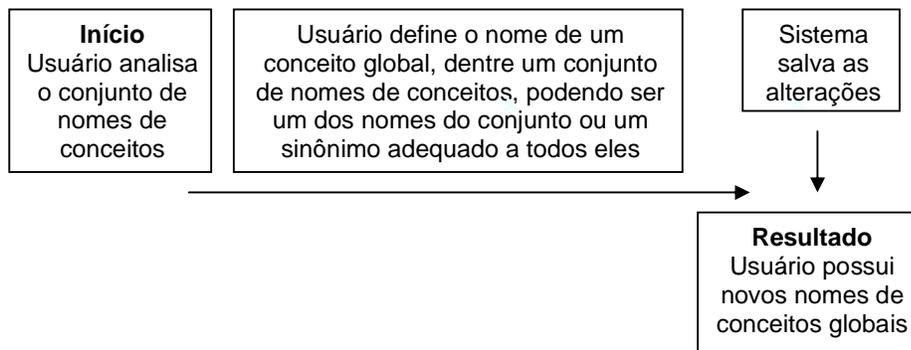


Figura 4.25 – Fluxo de tarefa para a tarefa 10.

A tarefa 11 refere-se à Regra da Unificação de Enumeração, de acordo com o fluxo de tarefa detalhado na Figura 4.26.

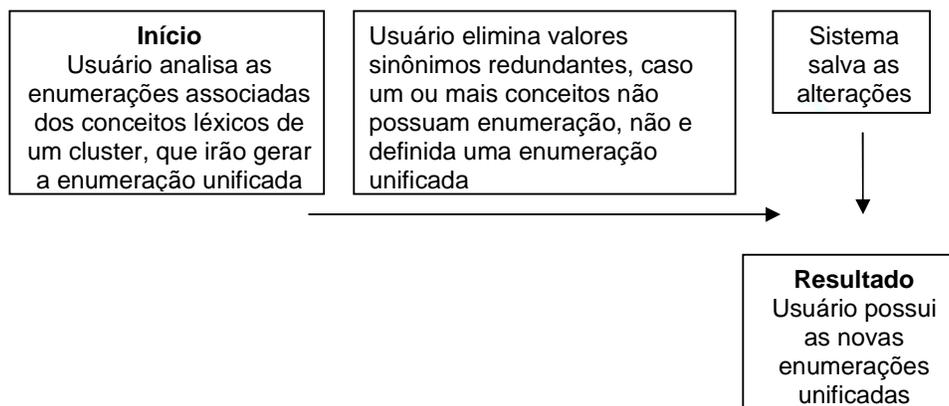


Figura 4.26 – Fluxo de tarefa para a tarefa 11.

A tarefa 12 refere-se à Regra da Unificação de Relacionamento, de acordo com o fluxo de tarefa detalhado na Figura 4.27.

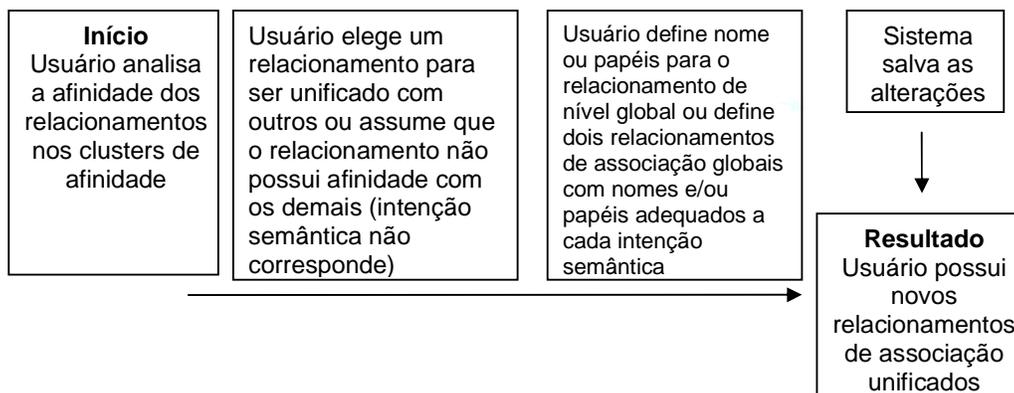


Figura 4.27 – Fluxo de tarefa para a tarefa 12.

4.5.2 Etapa 2 para as tarefas 10, 11 e 12

A etapa 2, apresentada na Figura 4.28, é feita em conjunto para as tarefas 10, 11 e 12, já que o usuário deve fazer as verificações na tela que apresenta o conjunto de clusters.

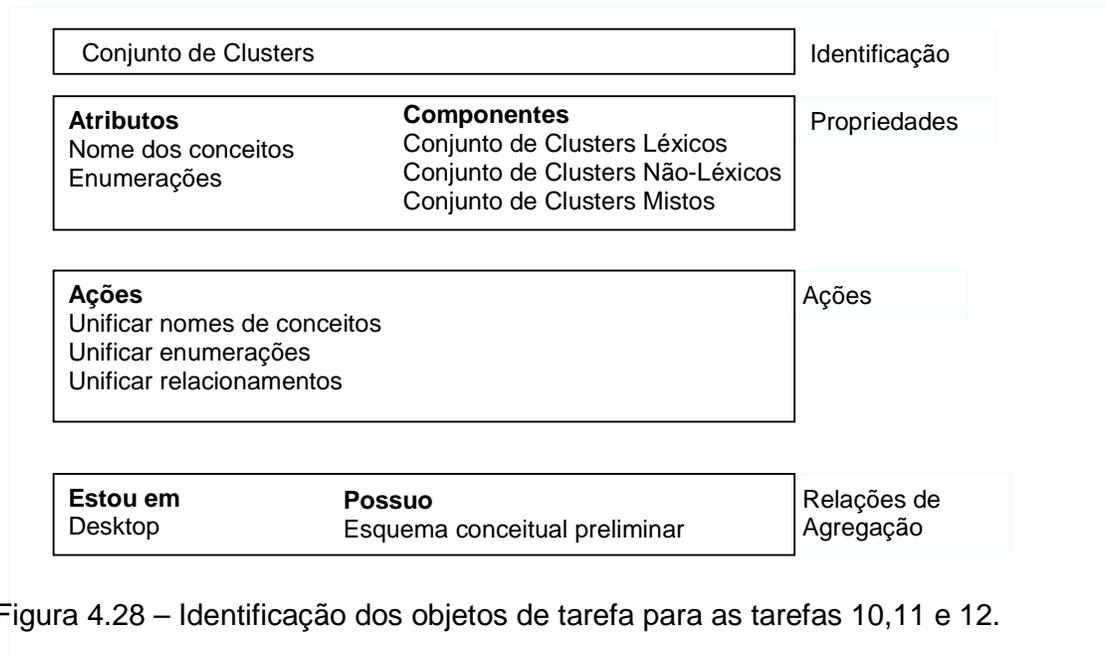


Figura 4.28 – Identificação dos objetos de tarefa para as tarefas 10,11 e 12.

4.5.3 Etapa 3 para as tarefas 10, 11 e 12

Nessa etapa estão associados os objetos de tarefa (descritos no item anterior) com os objetos de interface, de acordo com a Figura 4.29.

Como a fase de unificação é a primeira fase (que requer a intervenção do usuário) da etapa de integração semântica dos esquemas XML, as telas a seguir aparecem assim que o usuário seleciona os esquemas conceituais que deseja integrar, através do menu *Integration*, na barra de menu.

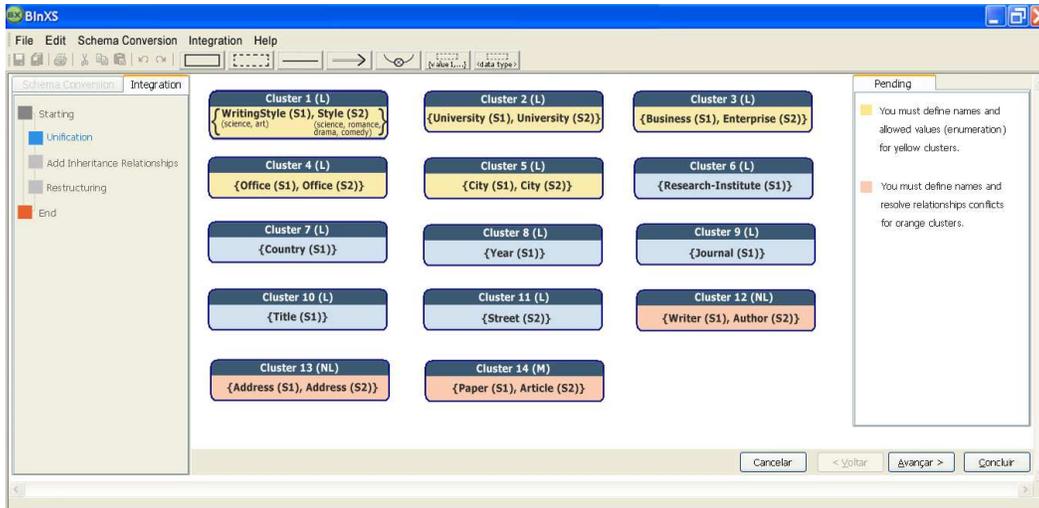


Figura 4.29 – Interface da fase de unificação

Importante lembrar que o usuário deve obrigatoriamente selecionar mais de um arquivo no menu *Integration*. Caso contrário não faz sentido a integração semântica e o sistema apresentará uma mensagem de erro, conforme a Figura 4.30.

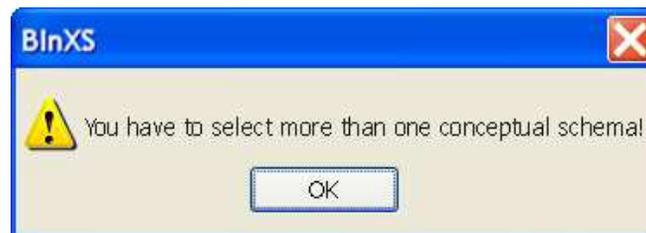


Figura 4.30 – Mensagem de erro exibida se o usuário selecionar somente um arquivo

Através da interface de unificação, o usuário pode unificar nomes de conceitos, unificar enumerações e unificar relacionamentos. Assim como na sub-etapa *Conversão* da etapa de *Conversão do Esquema*, uma aba no lado direito é exibida contendo as informações necessárias para que o usuário possa solucionar as pendências que o sistema encontrou. Caso contrário, o sistema apresenta uma mensagem de erro, semelhante a da Figura 4.16, apresentada anteriormente.

4.6 Fase da Inclusão de relações de herança

4.6.1 Etapa 1 para a tarefa 13

A tarefa 13 refere-se à inclusão de relações de herança, de acordo com o fluxo de tarefa detalhado na Figura 4.31.

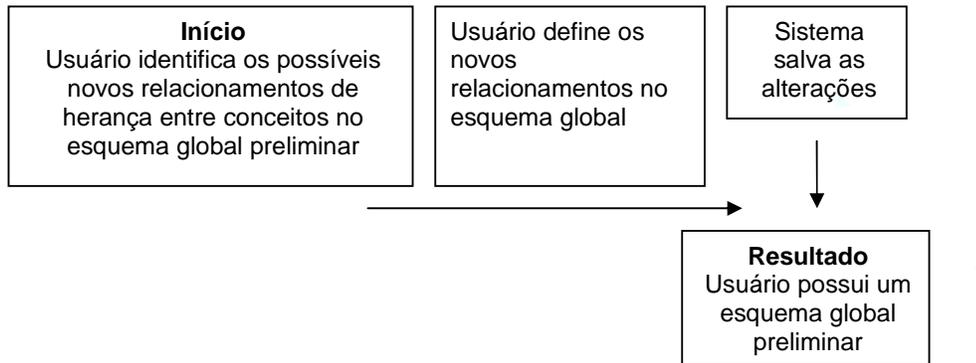


Figura 4.31 – Fluxo de tarefa para a tarefa 13.

4.6.2 Etapa 2 para a tarefa 13

A etapa 2, apresentada na Figura 4.32, tem como identificação o esquema global preliminar, no qual o usuário pode incluir relações de herança.

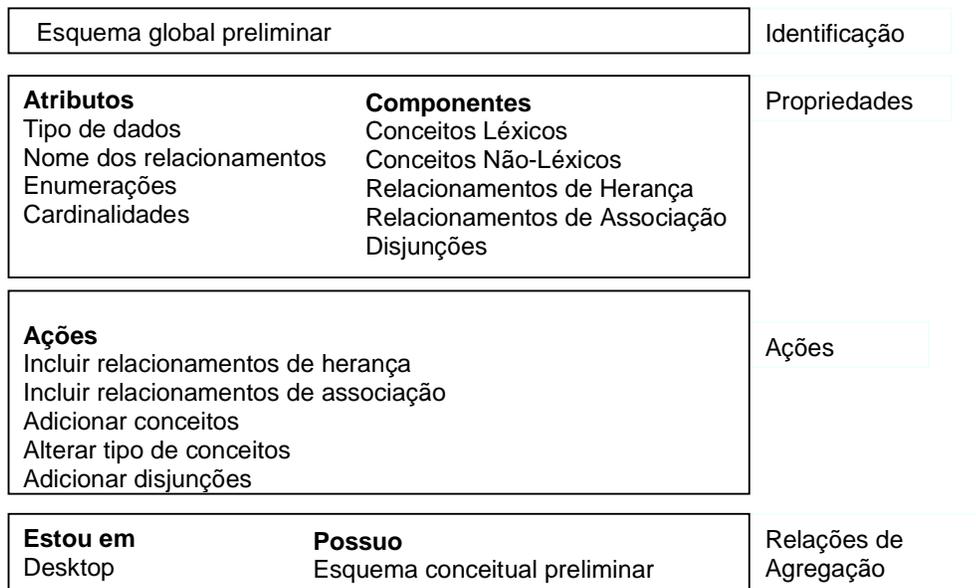


Figura 4.32 – Identificação dos objetos de tarefa para a tarefa 13.

4.6.3 Etapa 3 para a tarefa 13

Nessa etapa estão associados os objetos de tarefa (descritos no item anterior) com os objetos de interface, de acordo com a Figura 4.33. Para que seja possível a inclusão de relações de herança nesse esquema, em alguns casos é necessário que o usuário utilize outros componentes além do componente relacionamento de herança. Por exemplo, adicionar uma disjunção ou adicionar um conceito. Essa interação pode ser observada na Figura 2.19, no capítulo 2, que exemplifica a inclusão de relações de herança.

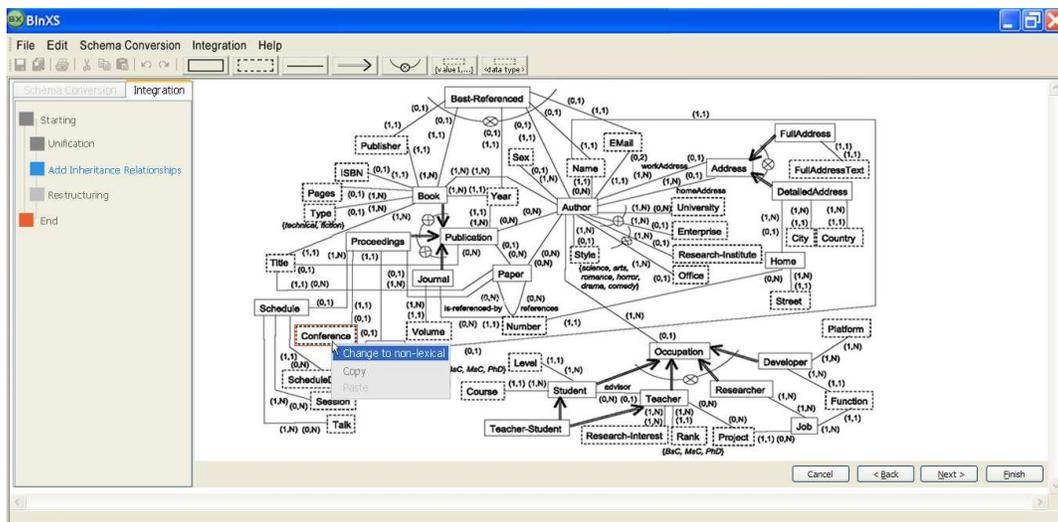


Figura 4.33 – Interface da fase de inclusão de relações de herança

4.7 Fase de Reestruturação

Esta sub-etapa realiza ajustes automáticos e manuais no esquema global preliminar a fim de gerar um esquema global definitivo. Os ajustes manuais são detalhados a seguir.

4.7.1 Etapa 1 para a tarefa 14, 15 e 16

A tarefa 14 refere-se à identificação de disjunções de relacionamentos de acordo com o fluxo de tarefa detalhado na Figura 4.34.

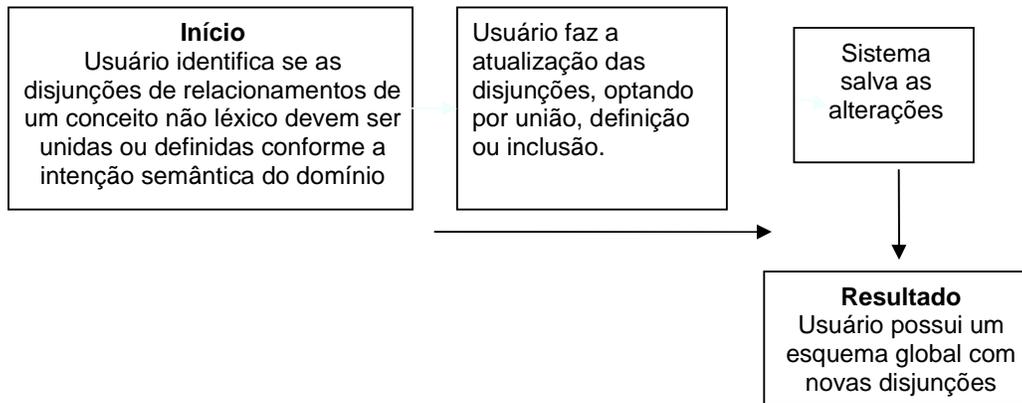


Figura 4.34 - Fluxo de tarefa para a tarefa 14.

A tarefa 15 refere-se à definição de conceitos genéricos relevantes, de acordo com o fluxo de tarefa detalhado na Figura 4.35.

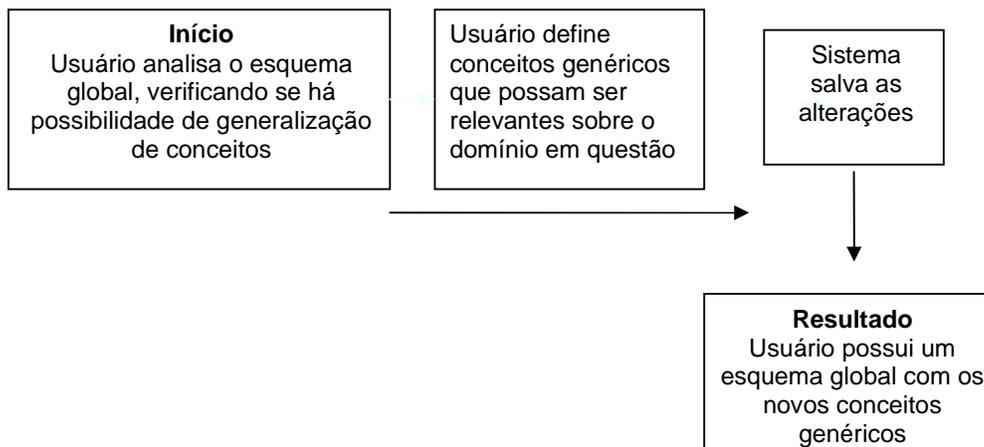


Figura 4.35 - Fluxo de tarefa para a tarefa 15.

A tarefa 16 refere-se à modificação de nomes e conceitos especiais, de acordo com o fluxo de tarefa detalhado na Figura 4.36.

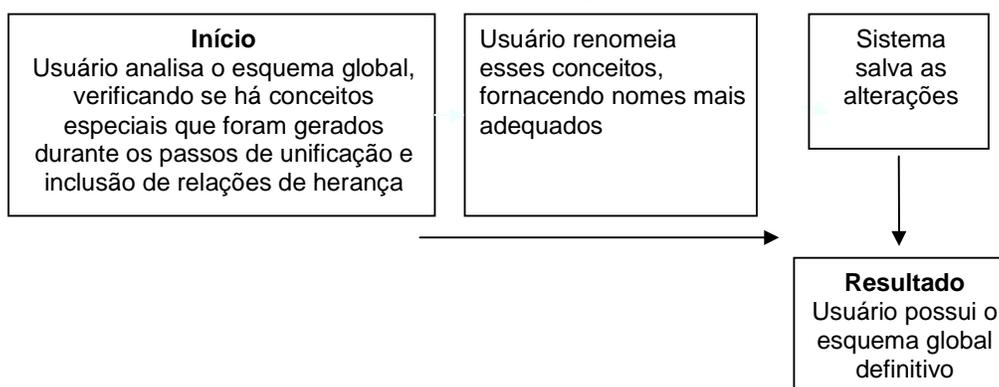


Figura 4.36 - Fluxo de tarefa para a tarefa 16.

4.7.2 Etapa 2 para as tarefas 14, 15 e 16

A etapa 2, apresentada na Figura 4.37, é feita em conjunto para as tarefas 14, 15 e 16, uma vez que o usuário pode fazer as alterações na tela que apresenta o esquema global preliminar.

| | | |
|---|---|-----------------------|
| Esquema global preliminar | Identificação | |
| Atributos Tipo de dados Nome dos relacionamentos Enumerações Cardinalidades | Componentes Conceitos Léxicos Conceitos Não-Léxicos Relacionamentos de Herança Relacionamentos de Associação Disjunções | Propriedades |
| Ações Inserir e/ou remover relacionamentos Unir, definir ou incluir disjunções Modificar nomes de conceitos especiais | Ações | |
| Estou em Desktop | Possuo Esquema conceitual preliminar | Relações de Agregação |

Figura 4.37 – Identificação dos objetos de tarefa para as tarefas 14, 15 e 16.

4.7.3 Etapa 3 para as tarefas 14, 15 e 16

Nessa etapa estão associados os objetos de tarefa (descritos no item anterior) com os objetos de interface, de acordo com a Figura 4.38.

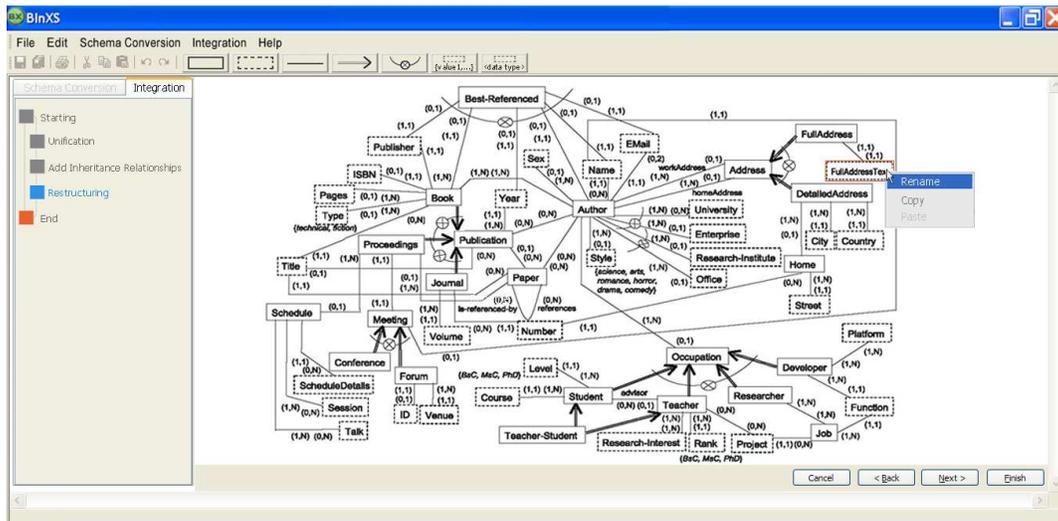


Figura 4.38 – Interface da fase de reestruturação

Nessa última tela o usuário pode renomear conceitos, incluir disjunções, inserir e/ou remover relacionamentos. Essas interações são feitas diretamente com esses componentes que estão dispostos na barras de componentes. As figuras 2.20, 2.21 e 2.22 do Capítulo 2 mostram alguns exemplos.

Após a intervenção do usuário nessa etapa de reestruturação, um esquema global definitivo é apresentado, conforme a Figura 4.39, sendo finalizado o processo de integração semântica dos esquemas conceituais representativos de cada esquema XML.

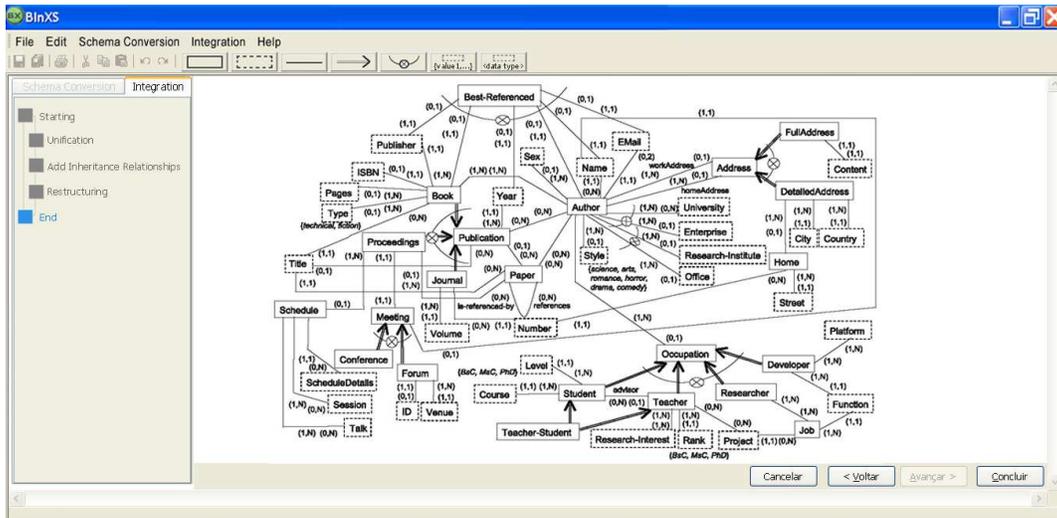


Figura 4.39 – Esquema global definitivo.

5 Conclusões e Trabalhos Futuros

Este trabalho apresenta o projeto da interface gráfica do Ambiente BInXS. Para tanto, o ambiente foi estudado, bem como algumas abordagens de projeto de interface.

Além disso, foram realizados estudos sobre usabilidade de interfaces. Uma das técnicas, *The Bridge*, foi escolhida para modelar o projeto de interface gráfica do ambiente BinXS.

Após a escolha da abordagem, as tarefas que requerem a intervenção do usuário foram modeladas de acordo com a abordagem, gerando o protótipo das telas dessa interface.

Posteriormente poderão ser desenvolvidos novos trabalhos, principalmente a implementação propriamente dita da interface projetada, bem como o melhoramento da mesma.

Como alguns módulos do BInXS já foram desenvolvidos, em outros trabalhos, as conexões desses poderão auxiliar na implementação da interface geral do ambiente BInXS.

Outro fator importante será a realização de testes com os usuários, o que irá colaborar para o aperfeiçoamento da interface.

Referências Bibliográficas

[ABI, 2000] S. Abiteboul, P. Buneman, and D. Suciu. Data on the Web: From Relations to Semistructured Data and XML. Morgan Kaufmann, San Francisco, California, 2000.

[CYBIS, 2003] CYBIS, Walter de Abreu. Engenharia de usabilidade: uma abordagem ergonômica. Florianópolis, 2003. 142 p.

[CRUZ, 2004] I. F. Cruz, H. Xiao, and F. Hsu. An Ontology-Based Framework for XML Semantic Integration. In International Database Engineering and Applications Symposium (IDEAS'04), pages 217-226, Coimbra, Portugal, 2004. IEEE.

[CXML, 2005] CXML.org. Available at: <<http://www.cxml.org>>, mar 2005.

[DBL, 2002] DBLP Bibliography. Disponível em: <<http://www.informatik.unitrier.de/ley/db/>>. Acesso em jul.2007.

[DEITEL, 2003] DEITEL, H. M.; Deitel P. J., Nieto, P. J.; Lim T.M.; Sadhu, P. XML Como Programar. Editora Bookman, 2003.

[EBisXML, 2005] EBisXML. Available at: <<http://www.basda.org>>, mar 2005.

[ISO 1997] ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs). Part 11 — Guidelines for specifying and measuring usability. Genève: International Standards Organization.

[ISO 1999]. ISO 13407: Human-centred design processes for interactive systems. Genève: International Standards Organization.

[LARRY, 1999] CONSTANTINE, LARRY L. and LOCKWOOD, LUCY A.D.. Software for Use: A Practical Guide to the Models and Methods of Usage. Addison-Wesley, 1999.

[LSCIO, 2003] B. F. Lscio and A. C. Salgado. Generating Mediation Queries for XML-based Data Integration Systems. In 18th Brazilian Symposium on Databases (SBBD'2003), pages 99-113, Manaus, AM, October 2003.

[MELLO, 2002] MELLO, Ronaldo dos Santos. Uma abordagem bottom-up para a integração semântica de esquemas XML. Porto Alegre, 2002. 145 p.

[MELLO, 2005] MELLO, R. S. and Heuser, C. A. BInXS: A Process for Integration of XML Schemata. In International Conference on Advanced Information Systems Engineering, CAISE, 17., pages 151–166, Porto, Portugal.

[REY, 2001] REYNAUD C.; SIROT, J.; VODISLAV, D. Semantic Integration of XML Heterogeneous Data Sources, In: INTERNATIONAL DATABASE ENGINEERING & APPLICATIONS SYMPOSIUM, IDEAS, 2001, Grenoble, France. Proceedings... Los Alomitos: IEEE, 2001. p.199-208.

[SIG, 1999] SIGMOD Record: xml version (v.1.0). Disponível em: <<http://www.acm.org/sigmod/record/xml>>. Acesso em ago. 2007.

[Singh et al., 2004] SINGH, Inderjeet; BRYDON, Sean; MURRAY, Greg; RAMACHANDRAM, Vijay, VIOLLEAU, Thierry; STEARNS, Beth. Designing Web Services with the J2EE 1.4 Platform JAX-RPC, SOAP, and XML Technologies. United States: Addison-Wesley, 2004.

[XPATH, 2007] XML Path Language, 2007. Disponível em: <<http://www.w3.org/TR/xpath>> . Acesso em abr. 2007.

[WOOD, 1997] WOOD, Larry E. User Interface Design: Bridging the Gap from User Requirements to Design. Boca Raton, Florida, 1997. p. 16-54.

[W3C, 2005] W3C.org. Disponível em <<http://www.w3c.org>>. Acesso em out.2007.