4. Ferramenta Universo-I

No capítulo anterior foi apresentada a estratégia para identificação das fontes de informação de Leite et al (2007). Uma vez que havia tarefas repetitivas que envolviam cálculos e também a necessidade de um registro formal dos produtos gerados, decidimos construir uma ferramenta para apoiar a estratégia. Esta ferramenta foi construída sobre o conjunto de tecnologias da plataforma Eclipse. O nome Eclipse é em geral associado apenas ao seu ambiente de desenvolvimento integrado para Java (eclipse.org, 2008), mas atualmente o projeto Eclipse (Id., 2008) se define como: "*Eclipse is an open source community whose projects are focused on building an extensible development platform, runtimes and application frameworks for building, deploying and managing software across the entire software lifecycle*."

O conceito que nos levou a construção da ferramenta é comumente referenciado como CASE (Computer Aided Software Engineering) e se define como o uso de suporte baseado em computador no processo de desenvolvimento de *software* (SEI, 2008). Segundo Id. (2008) uma ferramenta CASE é um produto baseado em computador com objetivo de suportar uma ou mais atividades da engenharia de *software* dentro de um processo de desenvolvimento do mesmo. Existem diversas ferramentas CASE comerciais disponíveis na indústria para uso em projetos de *software* como o Rational Rose XDE Developer (ibm.com, 2008) e o Borland Together (borland.com, 2008), o que mostra a aceitação e uso do conceito na indústria.

No caso desta dissertação, o protótipo de ferramenta CASE tinha por objetivo dar suporte somente à estratégia apresentada aqui. Consideramos a ferramenta um protótipo por não ter ainda qualidade de nível industrial. Seu objetivo é dar suporte a um estudo de caso para auxiliar na investigação dos conceitos explorados nessa dissertação. As próximas seções descreverão os requisitos para construção da ferramenta e as técnicas que foram usadas em relação ao processo de construção. Na seção 4.2 será feita uma breve descrição das tecnologias empregadas e na seção 4.3, a arquitetura da ferramenta. Por último

serão mostradas algumas telas dessa arquitetura com alguns detalhes da sua operação.

4.1. Requisitos para a construção da ferramenta

De forma resumida, a ferramenta deveria suportar a execução da estratégia de identificação e seleção das fontes de informação, a nossa compreensão de tal suporte compreende:

- Registro das informações produzidas na execução de cada atividade: persistir os artefatos intermediários da metodologia bem como os artefatos finais da mesma em um formato recuperável e ao mesmo tempo em que facilite o intercâmbio de informações.
- A automação das tarefas: automatizar o maior número possível de atividades realizadas no contexto do método ou, onde não fosse possível automatizar por completo, auxiliar no mesmo. Por exemplo, seria difícil realizar a consolidação dos grafos de Referência automaticamente, pois tal tarefa é extremamente dependente do conhecimento do engenheiro de requisitos, mas podemos disponibilizar uma lista com os nomes de fontes que se assemelham, assim, auxiliando na tarefa de eliminação de redundâncias no momento da consolidação.

Optamos pelo uso de duas técnicas complementares para definição dos requisitos da ferramenta, o Léxico Ampliado da Linguagem (Leite e Franco, 1993) e uma técnica de construção de cenários (Leite et al., 2000).

4.2. Léxico Ampliado da Linguagem

O Léxico Ampliado da Linguagem (Leite e Franco, 1993) é uma estratégia específica de derivação de modelo conceitual a partir de um léxico, como meio de representar os conceitos da aplicação seguindo a filosofia de orientação a objetos. A proposta dos autores utiliza duas fases distintas: a construção de um léxico que descreve o vocabulário de uma aplicação e a derivação de um modelo conceitual formal através de heurísticas de aquisição.

Os autores definem o Léxico Ampliado da Linguagem como um conjunto com três diferentes entidades: sinais, noções e respostas comportamentais. O processo para construção do Léxico Ampliado possui quatro etapas:

- Identificar as principais fontes de informação: para os autores, as fontes de informação que devem ser levadas em consideração são as pessoas e os documentos do mesmo. As pessoas corretas devem ser identificadas para que estas apontem os documentos mais adequados.
- Propor uma lista de sinais relevantes no Universo de Informações: neste passo é criada uma primeira lista com os sinais que o engenheiro de requisitos identifica através de leitura ou ouvindo a fonte de informação.
- Elicitar o significado dos sinais: neste passo cada sinal do Léxico recebe uma entrada no Léxico Ampliado. Cada entrada é um conjunto de noções (denotações) e respostas comportamentais (conotações). Na descrição das denotações e conotações utiliza-se uma regra obrigatória que tenta a maximização do uso dos sinais no significado de outros sinais.
- Validar o léxico obtido: a validação é feita de duas formas: uma utiliza a checagem informal realizada com atores do Universo de Informações. E a outra por meio das estruturas do Léxico Ampliado em si, que revela sinais faltantes.

Na construção da ferramenta, cada etapa desta técnica foi aplicada na sua devida seqüência. Como resultado da primeira etapa, a fonte de informação principal a ser utilizada foi o artigo “A strategy for information source identification” (Leite et al., 2007). Tal fato era extremamente justificado uma vez que o autor da dissertação foi co-autor do artigo, e não haveria outros documentos a cerca do método proposto.

Na segunda etapa, a de identificação dos sinais, foi produzida uma lista com os 77 sinais iniciais. Na terceira etapa os 77 sinais identificados sofreram uma redução para 66 em função de sua relevância, ou seja, alguns sinais foram cortados por não justificarem o trabalho posterior sobre os mesmos. Alguns também foram cortados devido à repetição, pois se percebeu que eram sinônimos

de outros sinais ou pequenas variações de grafia. A elicitação dos significados correu em três sub-etapas distintas.

A primeira foi a classificação de cada um dos 66 sinais em uma classe de verbo, objeto, estado ou sujeito. Esta classificação nos auxiliaria na definição das conotações e das denotações a cerca do sinal. Esta técnica não estava prevista na proposição original em (Leite e Franco, 1993), mas foi utilizada para facilitar a aplicação posterior da técnica de construção de cenários (Leite et al., 2000). Tal regra auxilia na utilização do princípio da circularidade, que pretende maximizar o uso de símbolos na descrição de outros símbolos (Id., 2000). Na segunda subetapa, a cada símbolo foi atribuída ao menos uma noção e uma resposta comportamental e, na terceira etapa foi feita um checagem estrutural para aplicação do princípio da circularidade.

A etapa e a validação foram realizadas com o orientador da dissertação e gerou três correções a serem efetuadas:

- A tradução para a língua portuguesa, pois, foram utilizados os símbolos em inglês como no artigo original.
- Elaboração do vocabulário externo, ou seja, indicação no léxico dos termos que não fazem referência a outros termos. Tal indicação apareceria por contraste uma vez que os símbolos do léxico deveriam aparecer sublinhados na versão final do mesmo.
- Pequenas Correções: ainda nesta fase foram encontrados erros pelo orientador que foram corrigidos.

Para cenários e léxicos serem efetivamente utilizados, faz-se necessário o uso de boas ferramentas de edição, visualização, gerenciamento e integração com outros *softwares*, para ganho de produtividade (Felicíssimo et al., 2004). Para tal, foi utilizado o C&L (Cenários e Léxicos), é um ambiente colaborativo que auxilia a edição de cenários e léxicos descritos em linguagem natural semi-estruturada (Felicíssimo et al., 2004). Na figura 7 é mostrado o grafo do Léxico Ampliado gerado pela ferramenta. Este grafo nos dá visualmente uma noção do nível da circularidade do Léxico Ampliado.

O resultado final desta etapa foi um Léxico Ampliado, possuindo 66 entradas que se encontram em sua íntegra no apêndice A desta dissertação. Ele foi colocado na ordem em que os termos do léxico vão ocorrendo.

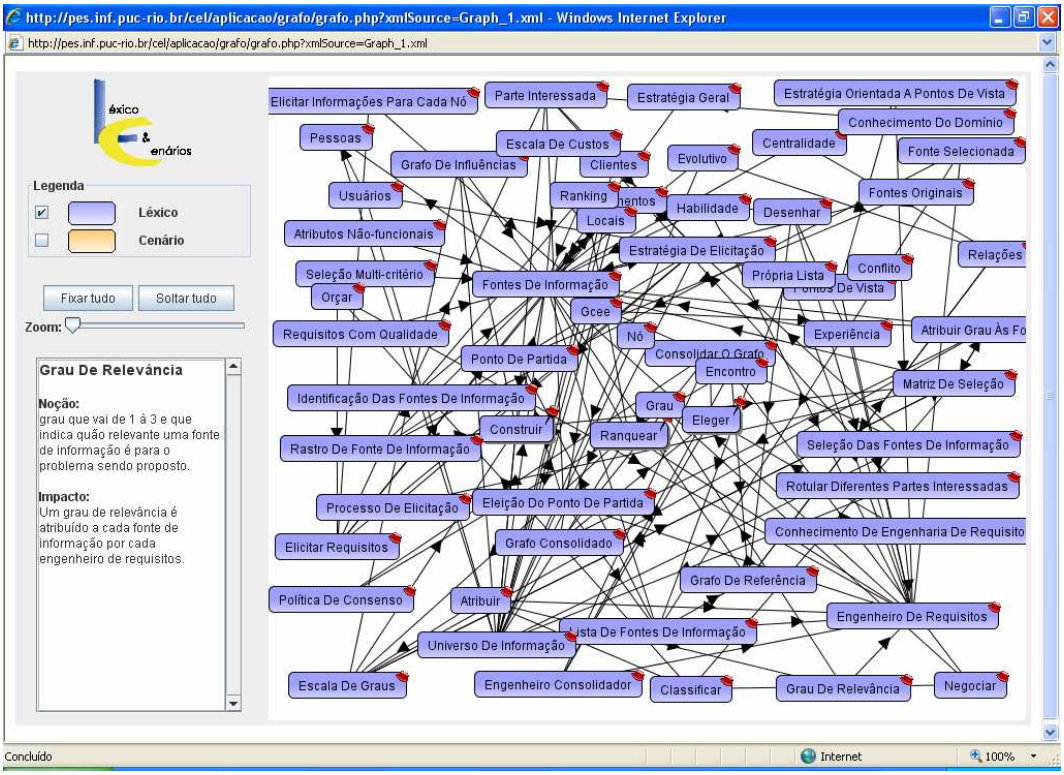

Figura 1 - Grafo do Léxico Ampliado da Linguagem gerado pela Ferramenta C&L.

4.3. Construção dos Cenários a Partir do Léxico Ampliado

A partir do Léxico Ampliado criado na etapa anterior se inicia o processo de construção de cenários de Leite et al. (2000). As cinco atividades que compõem o processo de Id. (2000) são:

1. DERIVAR (*DERIVATE*)
2. DESCREVER (*DESCRIBE*)
3. ORGANIZAR (*ORGANIZE*)
4. VERIFIAR (*VERIFY*)
5. VALIDAR (*VALIDATE*)

A figura abaixo demonstra a organização do processo utilizando um digrama SADT (Ross e Schoman, 1977). A seguir faremos uma descrição sucinta baseada em Leite et al. (2000) de como funciona o processo.

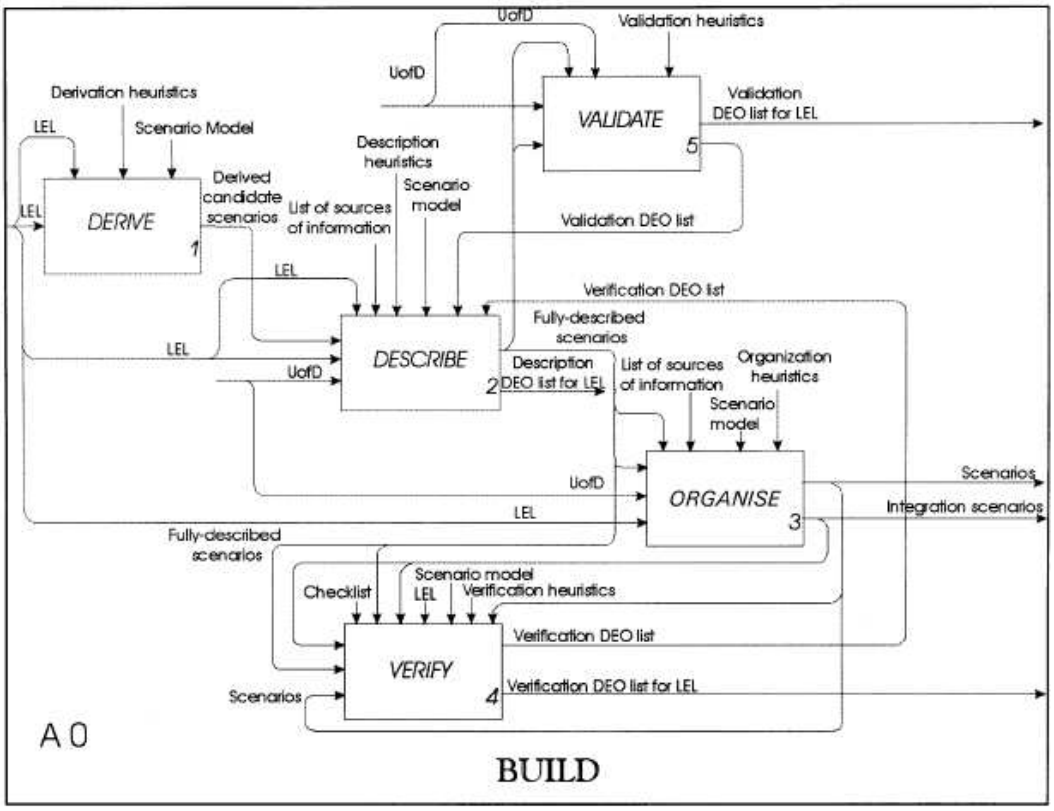

Figura 2 - Modelo SADT de contrução de Cenários (Leite et al., 2000).

A atividade DERIVAR tem por objetivo gerar os cenários candidatos a partir do Léxico Ampliado, utilizando o modelo de cenário e as heurísticas de derivação. O processo é composto de três passos:

1.1 Identificar Atores: os atores são as entradas no Léxico Ampliado que são do tipo sujeito. Os atores podem ser do tipo principal (que executa a ação) ou secundário. Este último são atores que recebem ou que dão informações, mas não tem responsabilidade nas ações executadas.

1.2 Identificar Cenários: as respostas comportamentais dos símbolos escolhidos como atores primários ou secundários são extraídos do Léxico Ampliado e incorporados à lista de cenários candidatos.

1.3 Criar: A intenção deste passo é construir o cenário com o máximo de informação possível advinda do Léxico Ampliado, aplicando as

heurísticas apropriadas, o produto desta etapa são os cenários candidatos.

A segunda atividade, DESCREVER, tem por objetivo melhorar o cenário candidato, adicionando informações do Universo de Informações usando o modelo de cenário, os símbolos do léxico nas descrições e aplicando as heurísticas de descrição. O resultado é um conjunto de cenários inteiramente descritos.

A terceira atividade, ORGANIZAR, segundo os autores é a mais complexa e mais sistematizada do processo de construção do cenário. Seu princípio é a idéia de integração dos cenários, descrições artificiais com o único propósito de fazer o conjunto de cenários mais compreensíveis e gerenciáveis. A primeira etapa desta atividade é a de REORGANIZAR. Esta subatividade consiste basicamente em colocar juntos dois ou mais cenários ou quebrar um cenário em dois ou mais. A última etapa desta atividade é a atividade DEFINIR (DEFINE), esta atividade identifica diferentes relações entre cenários de forma que se possa integrá-los. E por último a atividade INTEGRAR. Nesta atividade, os cenários são agrupados em hierarquias primeiro e depois essas hierarquias são agrupadas em seqüências.

A quarta atividade é a atividade VERIFICAR, ela é executada pelo menos duas vezes durante o processo de construção do cenário, a primeira sobre o cenário totalmente descrito e a segunda após a atividade ORGANIZAR. Esta é feita seguindo um *checklist* de heurísticas de verificação. A última atividade, VALIDAR, é realizada com os clientes/usuários geralmente através de entrevistas estruturadas ou reuniões. Durante a validação, a componente dúvida deve ser considerada com especial atenção aos cenários ainda apresentando tal problema.

Como resultado da instanciação do processo de construção dos cenários foram obtidos os seguintes produtos em cada uma das atividades do processo:

1. DERIVAR: foram identificados ao todo 17 cenários a partir do Léxico Ampliado. Como atores, temos o Engenheiro de Requisitos e uma especialização sua que é o Engenheiro Consolidador.
2. DESCREVER: cada um dos 17 cenários foi descrito. Como previsto no método empregado, foi gerado um lista do tipo DEO (Discrepâncias, Erros e Omissões), essa lista serve para registrar inconsistências que possam motivar mudanças nos cenários. O

DEO obtido possuía 8 itens. Cada um deles foi resolvido e continuamos a ter 17 cenários.

3. ORAGNIZAR: foi feita uma organização preliminar dos cenários. Uma vez que esta etapa tem que ser revisitada ao final do processo.
4. VERIFICAR: as verificações necessárias foram executadas. A sintaxe já havia sido verificada durante a criação dos cenários, portanto não foram encontrados erros posteriores. Foram verificadas as relações entre componentes e, uma vez que estas haviam sido verificadas também na etapa de criação, não foram encontrados erros posteriores. Na verificação semântica foi detectado que o verbo dos cenários deveria estar no infinitivo, para garantir uma melhor legibilidade dos cenários quando combinados com episódios. Na verificação inter-cenários foram identificados dois subcenários, cada um referente a cenários diferentes e foram, também, removidos dois cenários. Um por ser muito genérico em relação ao processo como um todo e outro por ser equivalente a um cenário já existente.
5. VALIDAR: Cenários em geral são validados com clientes ou usuários tendo em vista que não é o caso deste projeto. A validação deste projeto foi realizada com o professor orientador da dissertação, foram apontadas três correções a serem feitas. A primeira foi demonstrar os termos do léxico através de sublinhados e mostrar que princípio da circularidade estava sendo seguido. A segunda foi utilizar a sintaxe descrita no artigo para descrição do cenário. E a terceira foi um aumento do nível de detalhe dos cenários.

O resultado final da aplicação deste método foram os cenários que estão descritos abaixo. Primeiro mostramos os cenários que foram identificados e descritos.

Os cenários acima são aqueles que foram identificados a partir do Léxico Ampliado da linguagem. Já os cenários abaixo são os de integração, resultantes em grande parte da atividade ORGANIZAR.

A atividade Organizar também tem como saída a organização dos cenários em duas visões: a hierárquica e a temporal. A visão da organização temporal e da organização hierárquica estão demonstradas na figura abaixo, também mostramos o hipergrafo dos cenários gerado pela ferramenta C& L.

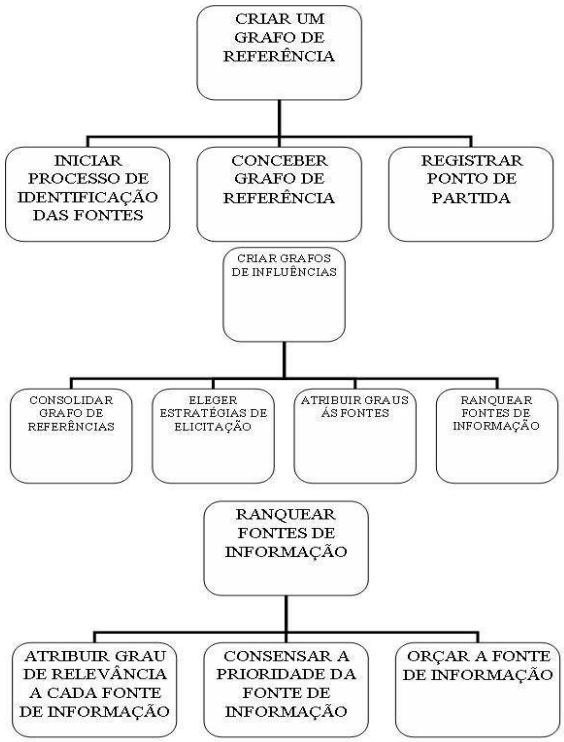

Figura 3 - Organização Hierárquica dos Cenários.

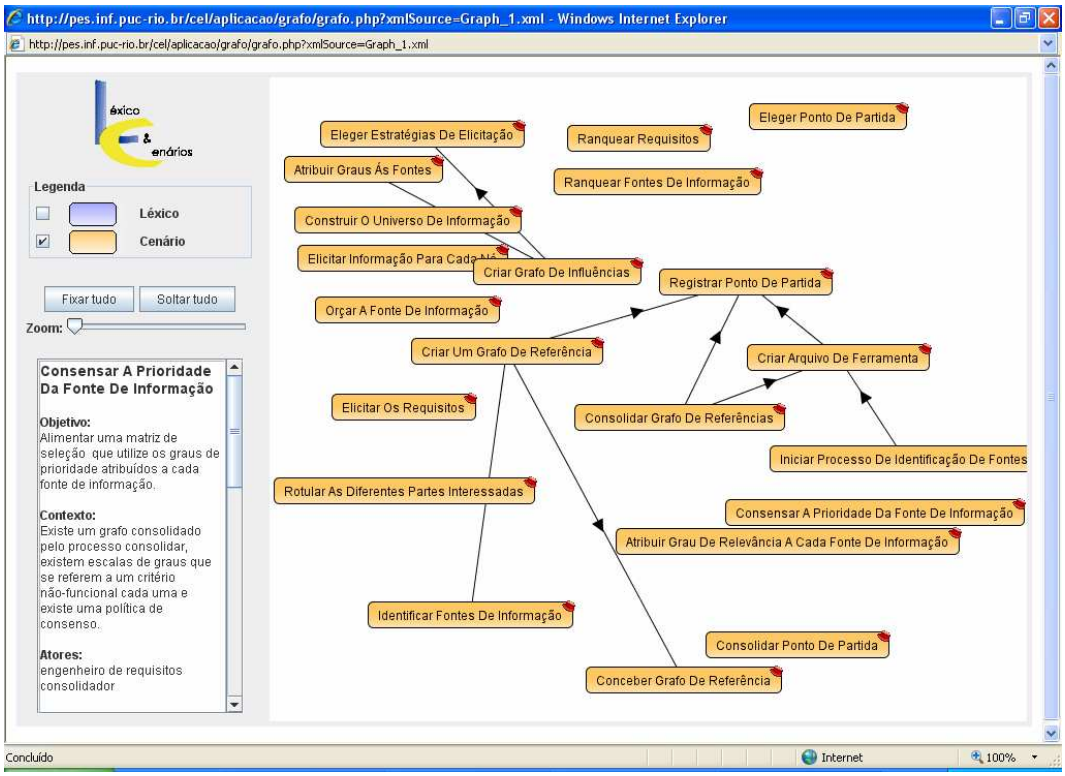

Figura 4 - Grafo dos Cenários Gerados pela Ferramenta C& L.

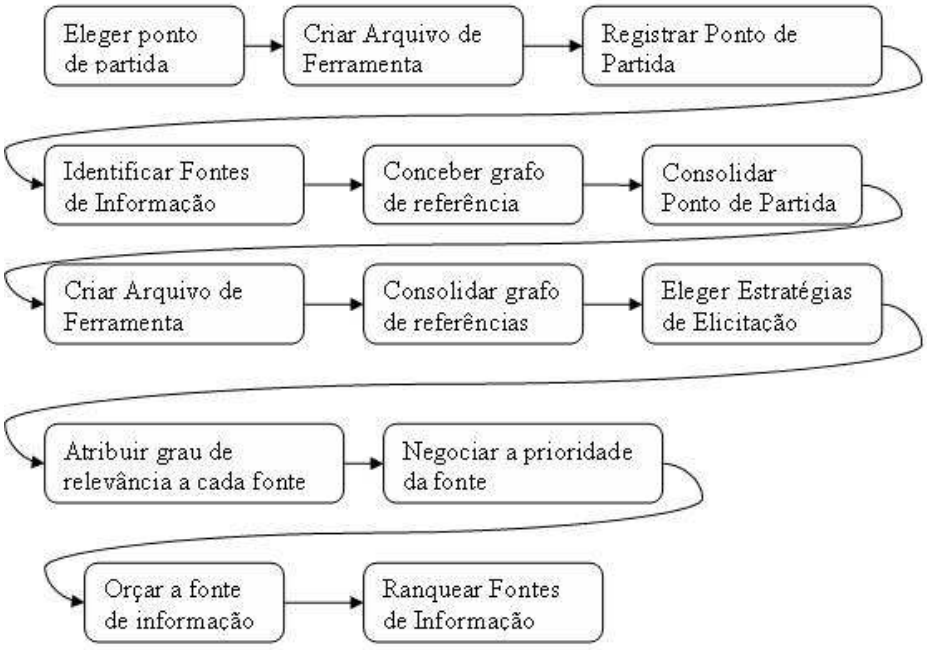

Figura 5 - Organização Temporal dos Cenários.

4.4. Tecnologias Empregadas

Como citado, o Eclipse não é apenas um IDE Java, mas também um conjunto de projetos de código aberto governados em geral pela mesma licença, a Eclipse Public License (EPL, 2008). Dentre estes muitos projetos disponíveis no Eclipse, dois foram selecionados para serem utilizados neste projeto: o *Eclipse Modeling Framework Project* (EMF) e o *Graphical Editing Framework Project* (GEF).

O EMF é um *framework* de modelagem e facilidades para geração de código fonte para construção de ferramentas e outras aplicações baseadas em modelos de dados estruturados (eclipse.org/emf, 2008). O *framework* funciona da seguinte forma, a partir da especificação de um modelo descrito em XMI (XMI, 2008), ele provê ferramentas de suporte em tempo de execução para produzir um conjunto de classes Java para o modelo, e também um conjunto de classes do tipo *adapter* para possibilitar a programação de visões, edição do modelo baseada em comandos e um editor básico (eclipse.org/emf, 2008).

O padrão XMI é um padrão criado pela Object Management Group (conhecida pela sigla, OMG) para gravação e troca de metadados. É muito comum sua utilização em ferramentas do tipo CASE para a persistência dos modelos criados. Quando nos referimos a classes *"adapter"*, programação de visões e edição baseada em comandos, estamos nos referindo respectivamente aos padrões de projeto comportamentais *Adapter* e C*omamnd* (Gamma et al., 1994) e a visão é um dos componentes do padrão arquitetural *Model-view-controller* (MVC) (Burbeck, 1992). O EMF se apóia sobre os conceitos de MDA (eclipse.org/mda 2008), padrão desenvolvido pela OMG em conjunto com a indústria. Seu objetivo de separar a lógica fundamental por trás de uma especificação dos aspectos específicos do *middleware* que o implementa.

Segundo Moore et al. (2008), a idéia por trás do conceito da especificação MDA é poder desenvolver e gerenciar o ciclo de vida completo da aplicação, colocando o foco no modelo. O modelo em si é descrito em um meta-modelo. Então, pela utilização de mapeamentos, o modelo é usado para gerar artefatos de *software* que irão implementar o sistema real. O EMF é a implementação de uma parte da especificação MDA para o Eclipse IDE.

O GEF é um *framework* que permite aos desenvolvedores criarem editores gráficos a partir de um modelo de aplicação existente (eclipse.org/gef, 2008). O princípio do GEF é disponibilizar ao desenvolvedor um conjunto de operações comuns ou especializá-las para domínios específicos. A arquitetura utilizada pelo GEF utiliza o modelo *Model-view-controller* (MVC) (Burbeck, 1992). Segundo os projetistas do GEF, além de possibilitar que pequenas mudanças sejam aplicadas a partir da visão, o *framework* é completamente neutro em relação à aplicação, provendo os fundamentos para construir qualquer tipo de aplicação. Como exemplos de aplicações que podem ser construídas os autores citam: diagramas de atividade, construtores de interfaces gráficas, diagramas de classe, máquinas de estado, dentre outros.

Em Moore et al. (2004) são elencadas diversas razões para usar os dois *frameworks* em conjunto:

- *You can use EMF's code generation facilities to produce consistent, efficient and easily customizable implementations of your model objects. If your model evolves during development, you can regenerate the code to reflect changes to the model, while preserving your customizations.*
- *The MVC architecture used by GEF relies on controllers that listen for model changes and update the view in response. If you use an EMF model, notification of model change is already in place, as all EMF model objects notify change via EMF's notification framework.*
- *The implementations generated for your model objects ensure that the model remains consistent, for example, when a reference is updated, the opposite reference is also updated.*
- *EMF provides support for persisting model instances, and the serialization format is easily customizable.*
- *Your applications can use the reflective API provided by EMF to work with any EMF model generically.*

Baseado nessas vantagens é que foi feita a escolha das duas tecnologias para a construção. Mas há ainda uma terceira tecnologia que foi adotada que precisa ser mencionada, o Rich Cliente Plataform (RCP) (eclipse.org/rcp, 2008). Uma das

características mais marcantes do IDE Eclipse é a sua capacidade de criar *plug-ins*. Um *plug-in*, assim como um objeto, no ambiente Eclipse é definido como o encapsulamento de um comportamento e/ou dados que interagem com outros *plug-ins* para formar um programa executável (eclipse.org,, 2008). As tecnologias supracitadas (EMF e GEF) foram construídas para serem utilizadas dentro do IDE Eclipse na construção de *plug-ins*. Isso traz desvantagem, pois para utilizar a ferramenta o usuário necessitaria não só ter o IDE Eclipse como um conhecimento mínimo de sua operação. O RCP é na verdade o conjunto mínimo de *plug-ins* necessário para construir uma aplicação do tipo cliente rico. Isto permite que aplicações que não sejam um IDE possam ser construídas usando um subconjunto da plataforma. Estas aplicações ricas são baseadas ainda num modelo dinâmico de *plug-ins* e a interface de usuário é construída sobre os mesmos *toolkits* e conjunto de ferramentas (eclipse.org/rcp, 2008).

Cada *framework* utilizado está dividido em uma série de *plug-ins*. De forma resumida, o resultado final do uso destas três tecnologias é a geração de uma aplicação contendo os *plug-ins* e, também , durante a exportação da aplicação é adicionada uma maquina virtual Java apara que a mesma não dependa de instalações por parte do usuário.

4.5. Arquitetura

A utilização dos *frameworks* citados na seção anterior tiveram uma grande influência na arquitetura da ferramenta. A razão é que sua extensibilidade está associada ao uso de alguns padrões de projeto que teriam que obrigatoriamente estar presentes no projeto. Em alguns casos os objetos que participavam dos padrões como clientes ou como implementações de parte do padrão.

4.6. Padrões de Projeto utilizados

O modelo da figura 12 é um Diagrama de Classe EMF. É o modelo que foi utilizado pelo EMF para gerar o código Java que foi utilizado na aplicação.

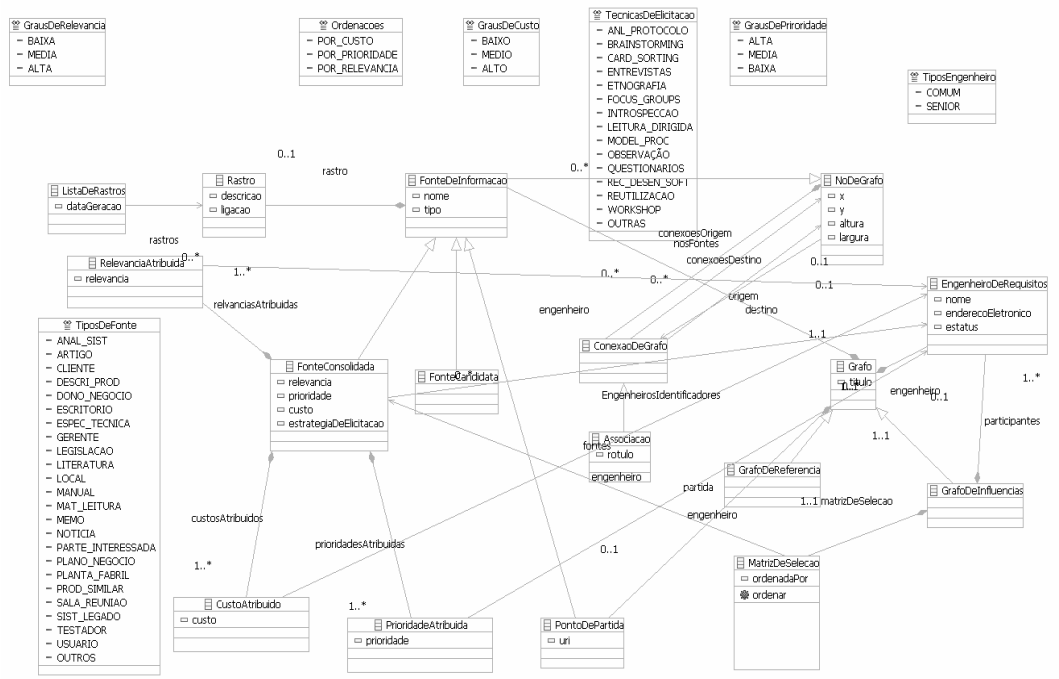

Figura 6 - Modelo EMF da ferramenta Universo-I.

O que o EMF faz:

- Gera as interfaces e as implementações das classes presentes no modelo. As implementações contendo a devida infra-estrutura de notificações de mudanças e outras facilidades descritas anteriormente.
- Uma implementação do padrão criacional *Abstract Factory* (Gamma et al., 1994). A razão para utilizar este padrão é garantir que quando da criação de uma instância de um objeto este esteja devidamente ligado ao resto do modelo. Suponhamos o exemplo da classe *FonteDeInformacao*, ao observarmos o modelo EMF da figura 9, vemos que esta configura uma composição com a classe *Grafo*. Portanto, todas as vezes que solicitarmos uma instância da classe *FonteDeInformacao* ao *Abstract Factory*, ele se encarregará de atribuir esta instância à classe *Grafo*, bem como de ligá-la ao modelo principal na forma de uma marca XML no padrão XMI, para ser persistida posteriormente.

No *framework* GEF, um diagrama é composto de um conjunto de EditParts. Cada EditPart é um elemento do diagrama, e estes são organizados em uma hierarquia de árvore. A raiz da árvore é um classe do tipo RootEditPart, as folhas

são EditParts. Mesmo as conexões são também EditParts. A edição sobre o modelo representado no GEF funciona a partir de classes do próprio *framework* que implementam o padrão *Command* (Gamma et al., 1994). Quando as modificações são executadas no modelo estas são refletidas na representação do GEF através de um mecanismo de notificação e, quando as mudanças são realizadas na representação gráfica, essas são refletidas no modelo através do uso de comandos.

Apesar de ambos possuírem uma implementação de pilha de comandos, as duas são incompatíveis entre si. Fazendo com que fosse necessário a partir dos comandos do GEF fazer as chamadas aos comandos EMF ou aos comportamentos dos objetos do modelo.

4.7. Exemplo de Uso da Ferramenta

Para demonstrar brevemente a utilização da ferramenta na execução dos cenários identificados no item 4.1, utilizaremos um subconjunto do exemplo que originalmente foi usado na concepção da estratégia. O caso das ambulâncias de Londres (Breitman et al., 1999). Seguiremos a ordem observada no diagrama da organização temporal dos cenários na figura 8.

O primeiro cenário a ser executado é o relacionado à escolha do ponto de partida (ELEGER UM PONTO DE PARTIDA). O ponto de partida escolhido foi o artigo supracitado. Os dois cenários seguintes são: CRIAR ARQUIVO DA FERRAMENTA e REGSITRAR PONTO DE PARTIDA, que estão mostrados na figura 13.

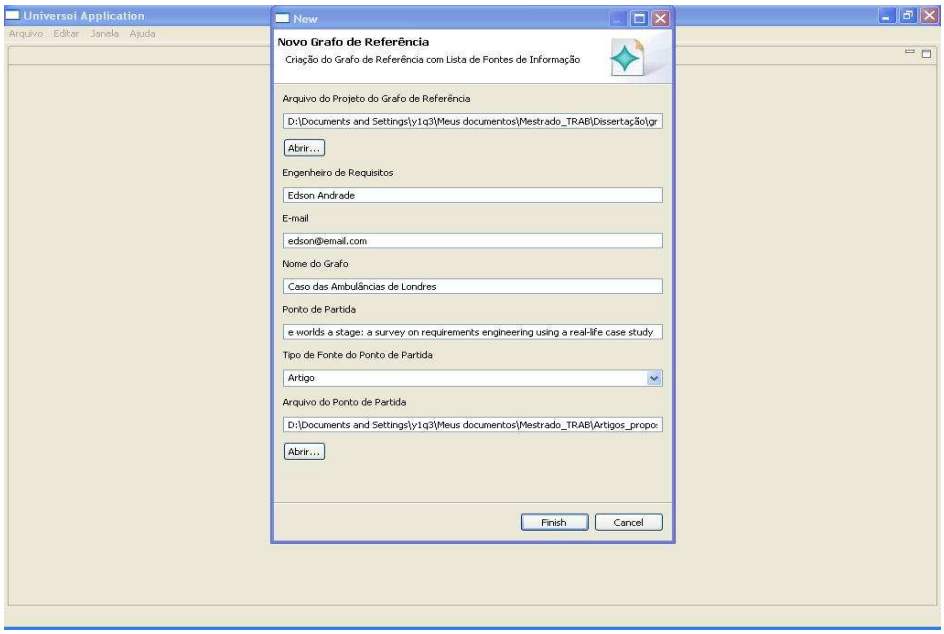

Figura 7 - Início da aplicação do método na ferramenta Universo-I.

No próximo passo, iremos executar os cenários IDENTIFICAR FONTES DE INFORMAÇÃO e CONCEBER GRAFO DE REFERÊNCIA. O registro da fonte identificada é feito fonte por fonte (figura 14) e, ao mesmo tempo, a ferramenta cria os nós a serem utilizados no Grafo de Referências. Na etapa seguinte, o engenheiro precisa apenas criar as relações entre os nós.

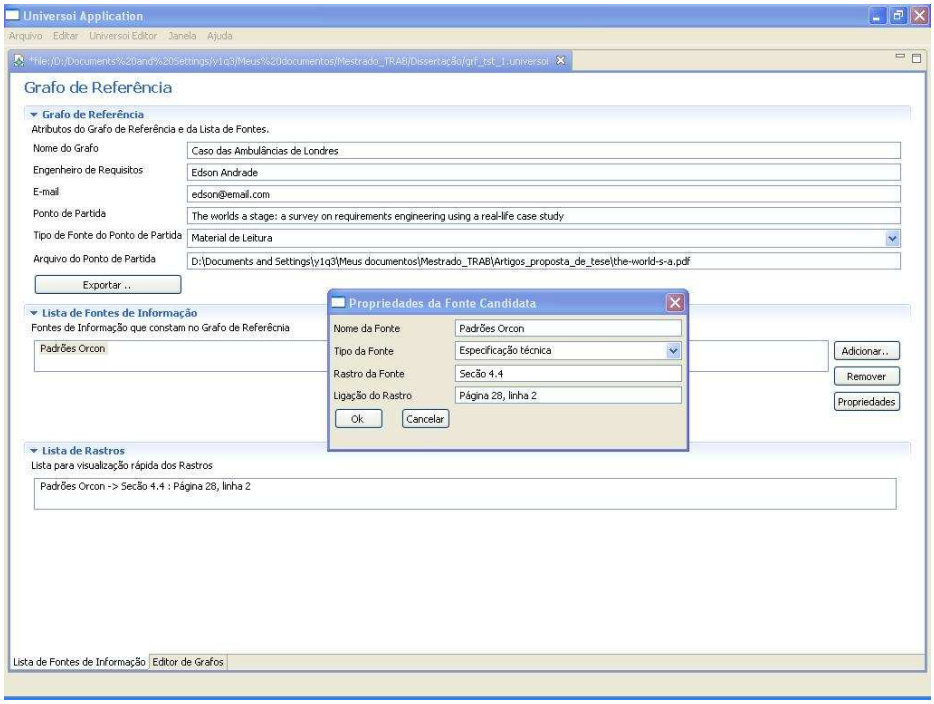

Figura 8 - Registro da Fonte de Informação.

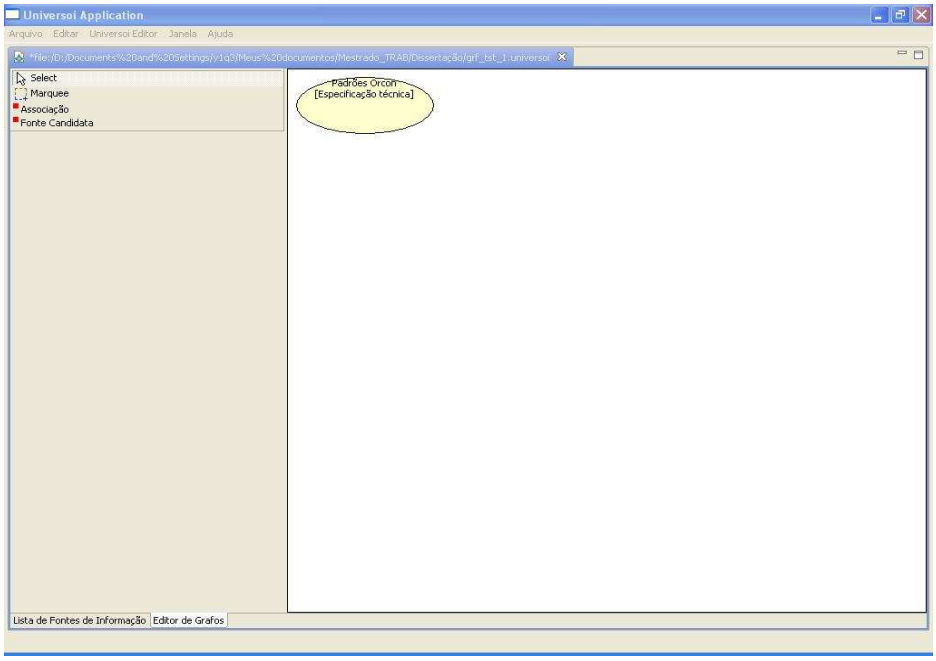

Figura 9 - Nó de Grafo a ser Utilizado.

Após cada engenheiro executar os cenários citados acima, passaremos aos cenários que são executados durante a reunião. Neste momento apenas um ator interage com a ferramenta, é o engenheiro de requisitos consolidador. O primeiro cenário a ser executado é o de CONSOLIDAR GRAFOS DE REFERÊNCIA. As telas abaixo mostram o momento em que os arquivos são consolidados, e é gerado um grafo consolidado.

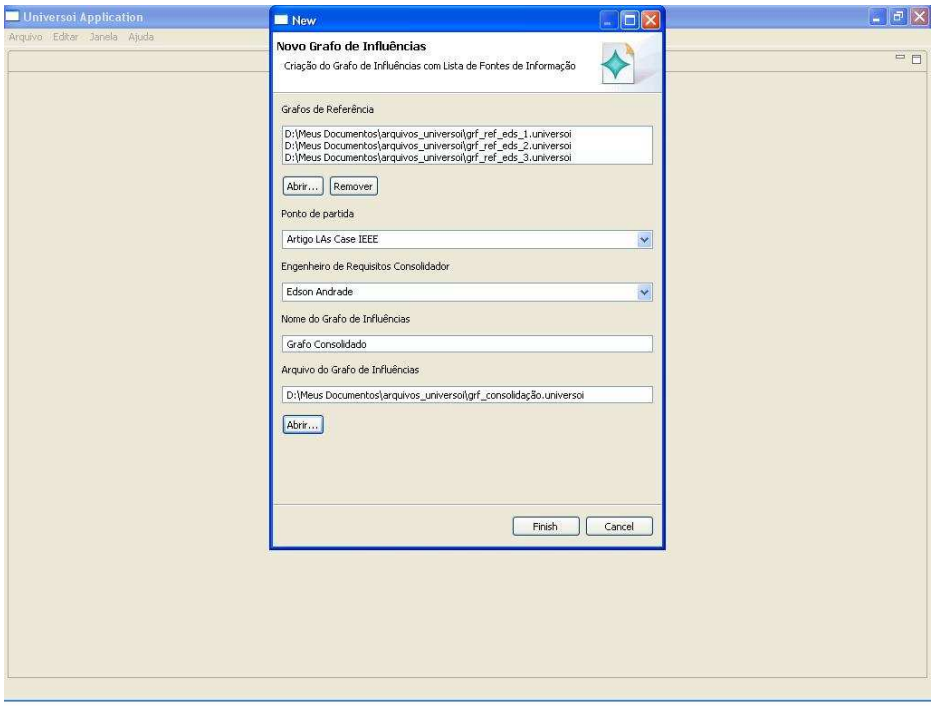

Figura 10 - Consolidação dos Grafos de Referência.

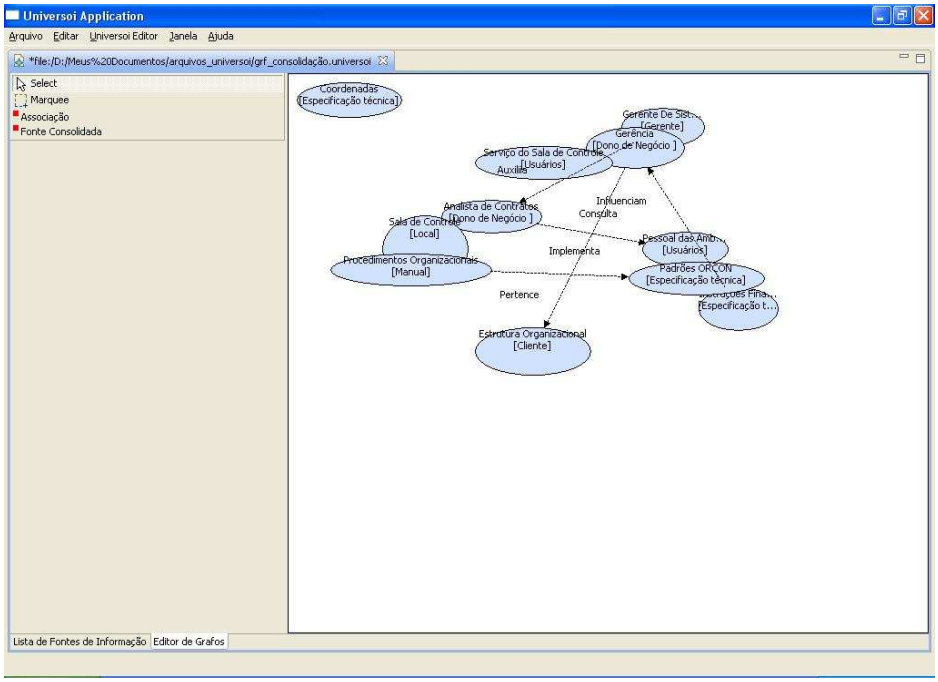

Figura 11 - Grafo Consolidado.

Na figura 18 podemos observar uma das funcionalidades da ferramenta, que é a detecção de fontes de informação com nomes semelhantes. Foi utilizada para tal uma implementação do algoritmo de Boyer-Moore (Boyer e Moore, 1977) para busca de cadeias de caracteres. O objetivo da existência da lista assinalada abaixo

é auxiliar na consolidação do grafo através da detecção de possíveis fontes repetidas.

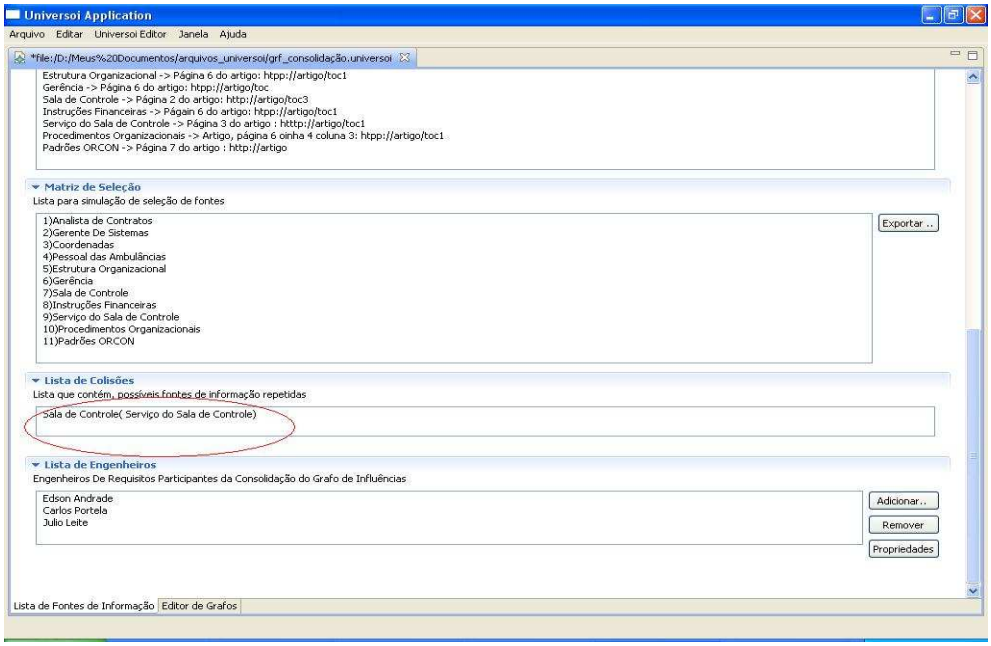

Figura 12 - Detecção de nomes semelhantes.

Os próximos cenários a serem executados são os cenários de ELEGER AS ESTRATÉGIAS DE ELICITAÇÃO e ATRIBUIR GRAUS ÀS FONTES. Como podemos ver pela tela na figura 19, a ferramenta dá apoio ao registro das estratégias de elicitação, bem como à atribuição de notas para geração posterior da planilha eletrônica que representará a Matriz de Seleção.

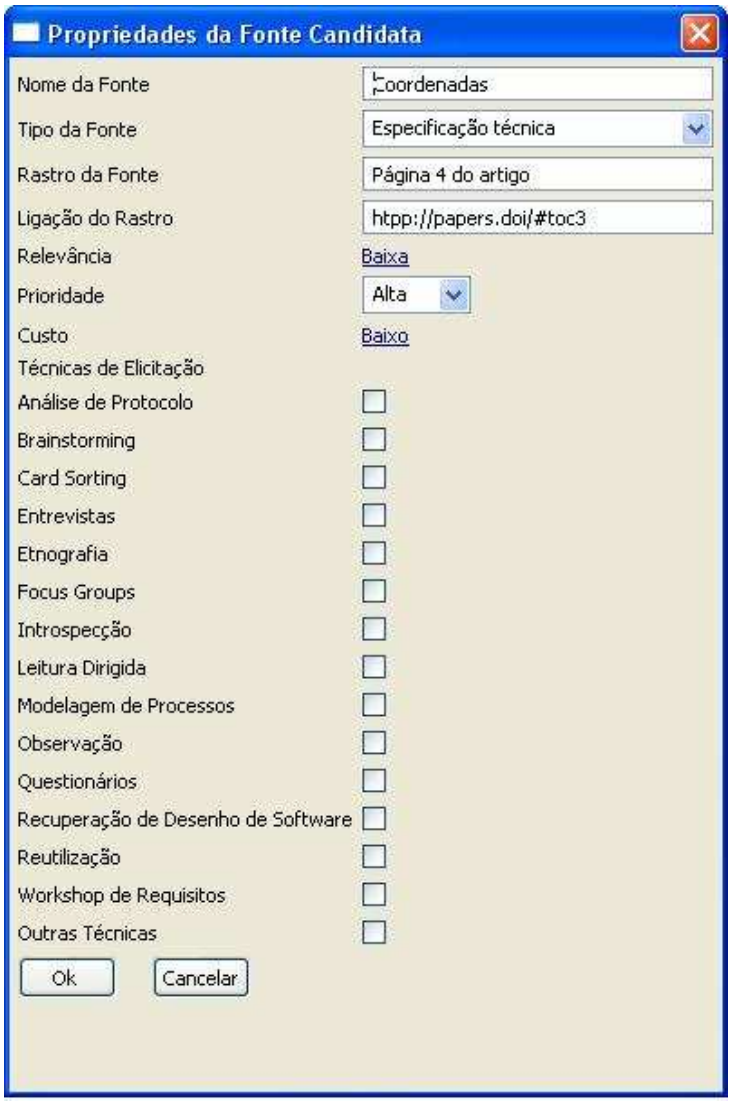

Figura 13 - Suporte ao registro das técnicas de elicitação e atribuição de notas.