Geração de Ontologias subsidiada pela Engenharia de Requisitos

Carolina Howard Felicíssimo[1]
Lyrene Fernandes da Silva[1]
Karin Koogan Breitman[1]
Julio Cesar Sampaio do Prado Leite[1]

{ cfelicissimo, lyrene, karin}@inf.puc-rio.br
www.inf.puc-rio.br/~julio

1- Departamento de Informática - PUC-Rio

Resumo

Cresce a necessidade do uso de ontologias em aplicações Web devido ao fato da maioria das informações publicadas estar em linguagem natural e, portanto, serem processadas apenas por humanos. Tais informações compõem o universo de informação da aplicação (UdI) que no processo de desenvolvimento de software é elicitado, modelado e analisado pela comunidade de engenharia de requisitos. Acreditamos que com o apoio de técnicas e métodos desenvolvidos e em utilização por esta comunidade podemos apoiar o processo de geração de ontologias por não especialistas. Neste trabalho apresentamos uma ferramenta que fornece apoio semi-automático a geração de ontologias tendo como base o Léxico Ampliado da Linguagem (LAL).

1. Introdução

À medida que o volume de informações cresce na Web, pesquisadores da indústria e do mundo acadêmico vem explorando a possibilidade de criar uma Web Semântica. Central à esta idéia está a utilização de ontologias, que fornecem uma *língua franca* permitindo que máquinas processem e integrem recursos Web de maneira inteligente, possibilitando buscas mais rápidas e acuradas e facilitando a comunicação entre dispositivos heterogêneos acessíveis via Web [Berners-Lee02]. A comunidade da Web acredita que, em um futuro próximo, todo negócio na rede deverá fornecer a semântica de suas páginas, através de uma ontologia [Fensel01].

Definimos uma ontologia como *"especificação explícita e formal de um conceito compartilhado"*. [Gruber93]. Em Ciência da Computação, ontologias são desenvolvidas para facilitar o compartilhamento e reuso de informações [Fensel01]. Elas descrevem conceitos, propriedades, restrições e axiomas de um domínio usando uma organização taxonômica, i.e., baseada em generalização e especialização. Aspectos composicionais, i.e., relacionamentos do tipo parte-de/todo, são ortogonais às ontologias e devem ser representados através de funções não taxonômicas, i.e., propriedades.

Ao contrário do que vem sido pregado por pesquisadores da área de Inteligência Artifical e engenharia do conhecimento, que se concentram na criação de ontologias genéricas, e.g. WordNet [Fellbaum98] e CyC [Guha90], a Web do futuro será composta de várias ontologias pequenas e altamente contextualizadas, desenvolvidas localmente por engenheiros de software e não especialistas em ontologias [Hendler01]. Sob esta luz, a tarefa

de desenvolver uma ontologia ou reutilizar partes de ontologias existentes deve ser simples o suficiente de modo a permitir que pessoas que não são especialistas no desenvolvimento de ontologias possam realizá-las.

Uma ontologia modela os conceitos e relações de um domínio, i.e., do Universo de Informação (UdI). Sendo assim, Breitman em [Breitman03] defende que a ontologia é um produto da engenharia de requisitos, i.e., é responsabilidade do engenheiro de requisitos modelá-la: primeiro, porque é durante o processo de definição do produto que o conhecimento do UdI é descoberto (elicitado); e segundo, porque a engenharia de requisitos tem um núcleo de conhecimento sobre os processos para captura, modelagem e análise de informações relevantes e, desta forma, pode auxiliar na tarefa de construção de ontologias. Com este enfoque, Breitman propôs um processo [Breitman03] para construção de ontologias centrado em uma estratégia de elicitação denominada *Léxico Ampliado da Linguagem* (*LAL*) [Leite93].

Neste artigo, apresentamos uma implementação que demonstra como uma técnica oriunda da engenharia de requisitos pode subsidiar a geração de ontologias. Com base no processo definido em [Breitman03] desenvolvemos uma ferramenta semi-automática para geração de ontologias. Relatamos nossa experiência no seu uso. Esta ferramenta é na verdade um plug-in para uma ferramenta de edição de Cenários e *LAL*, denominada C&L [Christoph03]. C&L é um projeto coordenado por nosso grupo de pesquisa, que vem trabalhando na elaboração e disseminação de técnicas e métodos de engenharia de requisitos a um baixo custo através do paradigma de desenvolvimento de software livre.

O restante deste artigo está organizado da seguinte maneira: na Seção 2, descrevemos como a engenharia de requisitos pode guiar o desenvolvimento de ontologias; na Seção 3, apresentamos a ferramenta C&L e seu processo semi-automatizado para elaboração de ontologias a partir do *LAL*; na Seção 4, descrevemos algumas das principais ferramentas de criação de ontologias e uma análise comparativa com o nosso plug-in para o C&L; na Seção 5, traçamos nossas conclusões e os trabalhos futuros que podem ser desenvolvidos de maneira a facilitar a criação e o reuso de ontologias.

2. Engenharia de Requisitos e Ontologias

Tendo em vista a necessidade de desenvolver software cada vez mais complexo e tendo ciência da dificuldade de atender as expectativas do usuário, os pesquisadores de engenharia de requisitos concentram seus esforços em desenvolver métodos, técnicas e ferramentas para facilitar as tarefas de elicitação, modelagem e análise de requisitos. Para realizar estas tarefas é necessário entender a linguagem do domínio (UdI) e escrever as informações destes domínio de maneira que seja possível compartilhá-las com os desenvolvedores.

Um recurso utilizado para apoio à elicitação, modelagem e análise do UdI é o Léxico Ampliado da Linguagem (*LAL*). O *LAL* é um hiper-documento que descreve os símbolos de um Universo de Informação. É utilizado para facilitar a comunicação e a compreensão de palavras ou frases peculiares a um Universo de Informação entre as pessoas envolvidas no desenvolvimento de um software [Leite97]. Cada termo do léxico tem dois tipos de descrição. O primeiro tipo, **noção**, é a denotação do termo ou expressão, i.e., seu significado. O segundo tipo, **impacto** ou **resposta comportamental**, descreve a conotação do termo ou expressão, i.e., provê informação extra sobre o contexto. Dicionários e glossários, de modo geral, só

representam a denotação dos termos. Como o *LAL* representa também os relacionamentos entre os termos, através dos impactos, temos uma representação muito mais detalhada da organização do Universo de Informação. Os termos do léxico são classificados em quatro categorias: **objeto**, **sujeito**, **estado** e **verbo**.

Há dois princípios fundamentais no *LAL*. O primeiro, é maximizar o uso dos outros termos do léxico quando descrevemos a noção e o impacto de um novo termo, denominado **princípio da circularidade**. O segundo, é minimizar o uso de termos externos ao UdI, denominado de **princípio do vocabulário mínimo**. O processo de construção do *LAL* consiste em seis passos listados a seguir [Kaplan00]:

- Identificar as fontes do UdI;
- Identificar a lista de termos relevantes no UdI;
- Classificar os termos entre sujeito, objeto, verbo, e estado;
- Descrever os termos através da noção e do impacto garantindo os dois princípios do *LAL*;
- Verificar o *LAL* através de inspeção;
- Validar o *LAL* com os atores do UdI.

O *LAL* é escrito em linguagem natural. Desta forma, o próprio usuário com a ajuda do engenheiro de requisitos pode escrevê-lo e validá-lo . Isto resulta em um modelo do domínio mais próximo ao mundo real. Através do *LAL* o engenheiro de requisitos consegue compartilhar, analisar e reusar o conhecimento do UdI.

Do ponto de vista conceitual, as ontologias são comparadas ao *LAL* porque oferecem uma organização explícita dos conceitos e relacionamentos existentes em um dado Universo de Informação. Ontologias são especificações formais dos termos do UdI e as relações entre eles [Noy01]. Nos últimos anos, têm havido uma crescente necessidade em categorizar os conceitos do UdI no contexto da Web. Esta necessidade é ocasionada, principalmente, devido a grande quantidade de dados disponível na Web não terem nenhuma semântica. Isto leva as pessoas a terem um trabalho extra de seleção das informações relevantes e as máquinas a armazenarem e tratarem os dados redundantes. A categorização taxonômica oferecida pelas ontologias surgiram para mudar este cenário.

Como citado por Noy em [Noy01], as principais razões para se construir ontologias são: compartilhar o entendimento da estrutura da informação entre pessoas e agentes de software; possibilitar o reuso de conhecimento do domínio; tornar as verdades absolutas do domínio explícitas; separar o conhecimento do domínio do conhecimento operacional; e analisar o conhecimento do domínio.

Na literatura há várias definições para ontologias. Neste trabalho adotamos a definição proposta por Maedche [Maedche02]. Nesta estrutura, uma ontologia é descrita pela tupla $O := \{C, R, H^C, rel, A^O\}$:

- C (conceitos) e R (relações) são dois conjuntos disjuntos;
- H^C é uma relação direcionada $H^C \subseteq C \times C$ que é chamada hierarquia de conceitos ou taxonomia. Por exemplo, $H^C (C1, C2)$ significa que C1 é um subconjunto de C2;
- *rel* é uma função $rel : R \rightarrow C \times C$ que relaciona os conceitos não taxonomicamente;
- A^O é um conjunto de axiomas expresso em linguagem lógica apropriada.

Em nossa visão, a modelagem de ontologias deve ser realizada durante a fase de requisitos. Para apoiar o desenvolvimento de ontologias, que já foi apontado como uma arte ao invés de ciência, propomos um processo baseado no Léxico Ampliado da Linguagem. A maior vantagem deste enfoque é poder contar com um método maduro para auxiliar na tarefa de elicitação, modelagem e validação dos conceitos e relacionamentos do Universo de Informação. O processo de construção do Léxico é estruturado e segue princípios sólidos de engenharia de Software e técnicas já estabelecidas para a captura, modelagem e posterior validação da informação modelada [Kaplan00]. O *LAL* provê a linguagem comum para a comunicação informal entre os interessados no processo de desenvolvimento de software, i.e., clientes, usuários e desenvolvedores, enquanto que ontologias fornecem esta linguagem de modo mais formal, permitindo o compartilhamento de informações entre máquinas e agentes de software. Na próxima seção, detalhamos a ferramenta que fornece apoio ao processo que serve como ponte entre a representação informal do *LAL* e ontologias.

3. Processo Semi-automático para Construção de Ontologias

Por utilizar o *LAL*, o processo proposto em [Breitman03] leva em conta as tarefas de elicitação, modelagem e análise para explicitar e comunicar o conhecimento do UdI. Este processo mapeia os termos do *LAL* nos elementos da ontologia: os termos do tipo **objeto** e **sujeito** são mapeados em **conceitos**; os termos do tipo **verbo** são mapeados em **propriedades**; os termos do tipo **estado** são mapeados em **conceitos** ou **propriedades**; a **noção** de cada termo é mapeada na **descrição** do respectivo **conceito**; e através da lista de **impactos** de cada termo do léxico mapeia-se o **verbo** em **propriedades** e o **predicado** em **conceitos** ou **restrições dos conceitos**. Na Tabela 1, resumimos o conjunto de mapeamentos entre termos do *LAL* e elementos da ontologia.

Elementos do *LAL*		**Elementos da Ontologia**
Termo (ou símbolo)		
- Tipo		
- Objeto e Sujeito	→	Conceitos
- Estado	→	Conceito ou axioma
- Verbo	→	Propriedade
- Noção	→	Descrição
- Impacto		
- Verbo	→	Propriedade
- Predicado (termos)	→	Conceitos ou Axiomas

Tabela 1 – Mapeamento entre os elementos do *LAL* e os da Ontologia

De modo a prover apoio semi-automático ao processo de construção de ontologias a partir do *LAL*, desenvolvemos um plug-in para a ferramenta C&L [Christoph03]. C&L é uma ferramenta de apoio à engenharia de requisitos e tem como objetivo principal a edição de Cenários e *LAL*. As funcionalidades oferecidas pelo C&L estão resumidas na Tabela 2. O plug-in desenvolvido para geração semi-automática de ontologias utiliza como dados de entrada o léxico de um projeto já editado e, gera como saída, uma ontologia em um arquivo do tipo *daml*, padrão [W3Consortium03].

Funcionalidades Gerais:
- Criar projeto e seu administrador; - Cadastrar usuário no projeto; - Verificar e aprovar ou rejeitar pedidos de alterações nos cenários; - Verificar e aprovar ou rejeitar pedidos de alterações nos termos do léxico; - Verificar e aprovar ou rejeitar pedidos de alterações nos conceitos; - Verificar e aprovar ou rejeitar pedidos de alterações nos relações; - ***Gerar Ontologia do projeto.***
Funcionalidades de edição do *LAL:*
- Edição: criar, alterar ou remover; - Marcação automática dos termos do *LAL*, seus sinônimos e nomes dos cenários; - Verificação de consistência em consequência da remoção de termos.
Funcionalidades de edição dos Cenários:
- Edição: criar, alterar ou remover; - Marcação automática dos termos do *LAL*, seus sinônimos e nomes dos cenários; - Verificação de consistência em consequência da remoção de cenários.
Funcionalidades de Entradas e Saídas:
- Gerar XML do projeto; - Recuperar XML do projeto; - **Gerar DAML da ontologia do projeto;** - **Histórico em DAML da ontologia do projeto.**

Tabela 2 – Principais funcionalidades da ferramenta C&L

A seguir, apresentamos o processo ilustrado na Figura 1 intercalado com descrições de como suas fases foram implementadas no plug-in de ontologias. Tal plug-in automatiza uma quantidade razoável das tarefas de geração de ontologias e oferece sugestões para o usuário quando é necessária a sua intervenção. Como demonstração, utilizamos um exemplo no domínio de sobremesas.

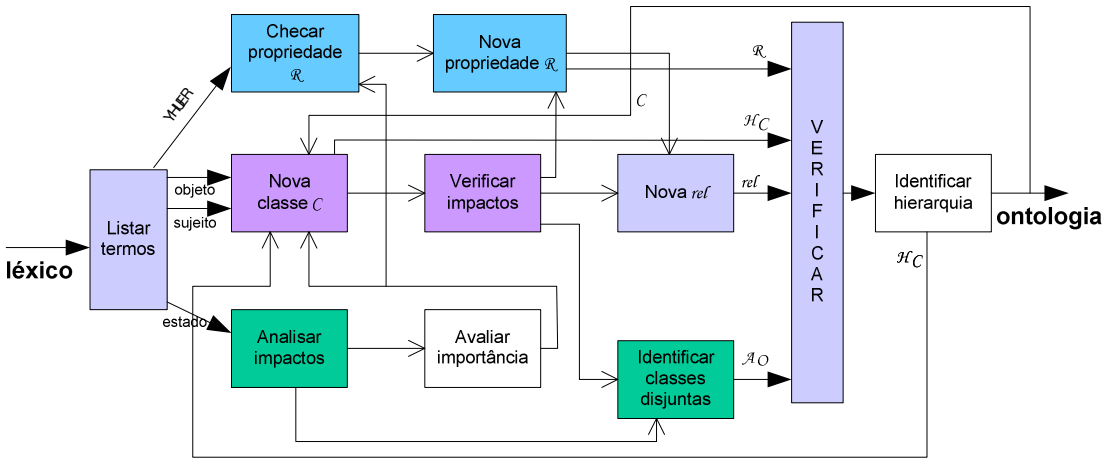

Figura 1 - Processo de construção de ontologias baseado no *LAL* [Breitman03]

Início do processo:

1. ***Listar os termos alfabeticamente de acordo com seu tipo (verbo, objeto, sujeito ou estado)***
2. ***Fazer 3 listas: conceito (classe) (C), propriedade (R) e axiomas (AO). Na lista de classes cada entrada terá um nome, descrição (linguagem natural) e uma lista contendo zero ou mais rel (função que relaciona o conceito em questão a outros, de maneira não taxonômica). As entradas na lista de axiomas terão nomes (labels) somente.***
3. ***Utilizando a lista de símbolos do léxico classificados como sujeito ou objeto, para cada termo:***
   3.1. Adicione uma nova classe a lista de classes. O nome da classe é o símbolo do léxico propriamente dito. A descrição da classe é a noção do termo.

Os itens 1, 2 e 3 foram totalmente automatizados, sendo desnecessária a intervenção do usuário. Os itens 1 e 2 representam a fase inicial do algoritmo em que são criadas as listas de conceitos, propriedades e axiomas onde são inseridos os respectivos elementos da ontologia durante o restante do processo. No item 3, inicia-se o processo de identificação dos elementos da ontologia. Os termos do *LAL* classificados como **objeto** e **sujeito** são inseridos na lista de **conceitos** e suas respectivas **noções** são inseridas como **descrições** de tais conceitos.

O algoritmo inicia a verificação dos **impactos** de cada termo do *LAL*:

3.1.1. Para cada impacto,
3.1.1.1. Checar se já faz parte da lista de propriedades da ontologia.
3.1.1.2. Caso não faça parte da lista (a propriedade ainda não existe), adicione uma nova propriedade na lista (de propriedades). O nome da propriedade deve ser baseado no verbo utilizado para descrever o impacto
3.1.1.2.1. Verificar consistência.
3.1.1.3. Na lista de classes adicione uma nova rel para a classe em questão. A rel é formada pela classe + a propriedade (definida em 3.1.1.1) + a classe relacionada (esta classe é o objeto direto/indireto do verbo utilizado no impacto do símbolo do léxico. Usualmente é um termo do próprio léxico e aparece sublinhado).

O **verbo** de cada um dos impactos do termo do léxico deve ser cadastrado na lista de **propriedades** da ontologia e o usuário deve identificar se no restante do impacto há **conceitos** (ainda não cadastrados) que devem fazer parte dessa ontologia, fazendo corretamente a identificação desses elementos.

Na Figura 2, ilustramos como o plug-in identifica cada elemento do impacto. Inicialmente, é identificado o **conceito** e sugerido que o restante do impacto é a nova **propriedade**. Em seguida, tenta-se encontrar qual é o elemento referente à esta **propriedade** e qual é o referente ao **conceito** do relacionamento. Para isto, é verificado na lista de propriedades da ontologia se o primeiro elemento dessa nova propriedade já está cadastrado. Em caso afirmativo, a própria ferramenta seleciona esse elemento cadastrado. Caso contrário, o cadastro deve ser realizado para a continuação do processo.

Para a identificação do outro conceito do relacionamento, a ferramenta seleciona o elemento quando o encontra (ou parte dele) cadastrado na lista de **conceitos** da ontologias ou na lista dos termos do *LAL* do tipo **sujeito** ou **objeto**. Não encontrando esse elemento (ou parte dele) cadastrado, a nova inserção é necessária.

É importante ressaltar que cabe ao usuário aceitar ou não as sugestões da ferramenta e conduzir o processo da forma correta. Isto é, se existir um conceito composto (conceito formado por mais de uma palavra) em que apenas parte dele já foi cadastrada, o algoritmo guia o usuário a cadastrar apenas a sua parte faltante e não o conceito composto. Por exemplo, se apenas o conceito *massa* estiver cadastrado no *LAL* e existir um impacto *base de biscoito tem massa de biscoito*, então, o algoritmo sugere que apenas o termo *massa* é o novo conceito e não *massa de biscoito* como seria o correto. Inferências errôneas (sugestões errôneas) podem ser evitadas quando se utiliza as heurísticas definidas em [Leite93] para escrever o *LAL*.

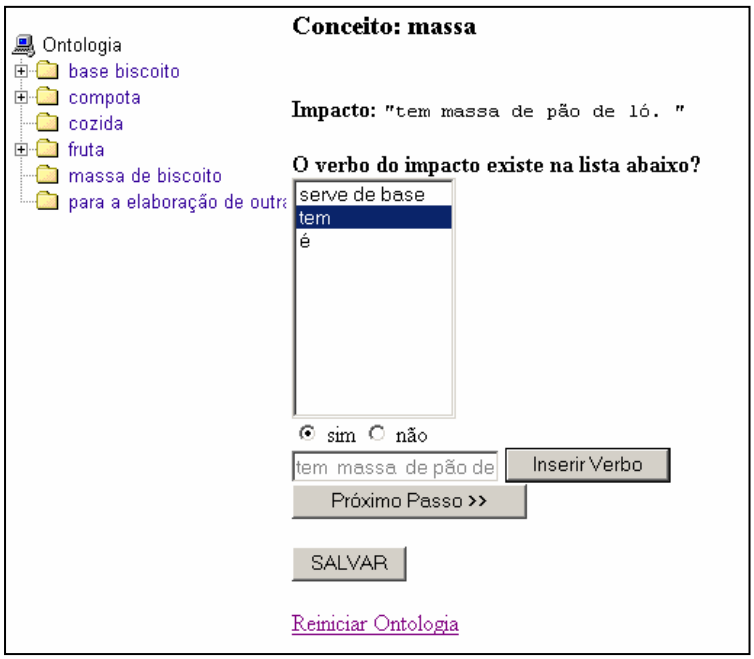

Figura 2 – Interface do C&L quanto à extração dos elementos da ontologias do Impacto dos termos do *LAL* – identificando conceitos e relações

O algoritmo verifica a existência de termos disjuntos:

3.1.1.4.Checar se existem indicativos de negação no vocabulário mínimo que relacionem duas ou mais classes. Verificar se estas classes possuem um relacionamento do tipo disjuntas (exemplo macho e fêmea).

3.1.1.4.1. Se verdadeiro, adicionar o disjoint a lista de axiomas.

3.2. Verificar consistência.

Nos itens 3.1.1.4, 3.1.1.4.1 e 3.2, é descrito como o algoritmo procede na análise de algum termo disjunto do conceito em análise. Caso exista, o usuário pode selecioná-lo na lista de

conceitos da ontologia (*namespace* próprio) ou referenciá-lo de uma outra ontologia existente, bastando para isto, informar seu *namespace* de referência. Ilustramos tais itens na Figura 3.

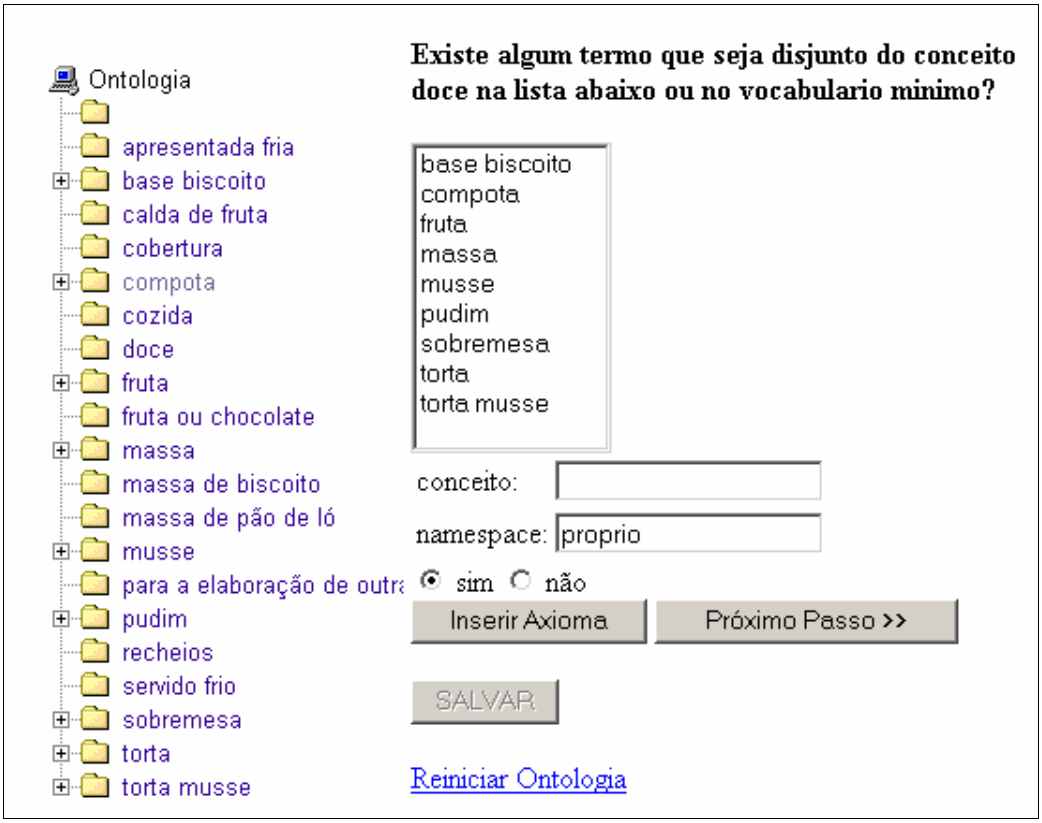

Figura 3 – Interface do C&L quanto à extração dos elementos da ontologias do Impacto dos termos do *LAL* – identificando axiomas

O algoritmo passa a classificar os termos do *LAL* do tipo **verbo**:

4. Utilizando a lista de símbolos classificados como tipo verbo, para cada termo:
4.1.1. Checar se já faz parte da lista de propriedades da ontologia.
4.1.1.1.Caso não faça parte da lista (a propriedade não existe), adicione uma nova propriedade na lista (de propriedades). O nome da propriedade é o símbolo do léxico propriamente dito.
4.1.1.1.1. Verificar consistência.

Os termos do *LAL* do tipo **verbo** são classificados como **propriedades** na ontologia gerada. No entanto, é necessário a intervenção do usuário caso a propriedade ainda não esteja cadastrada. A Figura 4 ilustra essa situação. Será preciso que o usuário clique em "Inserir Verbo" para o novo cadastro e para prosseguir no processo.

Figura 4 – Interface do C&L quanto à classificação dos termos do *LAL* do tipo verbo

O algoritmo passa a classificar os termos do *LAL* do tipo **estado**:

5. Utilizando a lista de símbolos classificados como tipo estado, para cada termo:

5.1.1. Para cada impacto

5.1.1.1.Tentar identificar a importância relativa do termo para a ontologia. Esta estratégia é similar a utilização de questões de competência proposta em [Gruninger95]. Estas questões são obtidas através do refraseamento dos impactos de cada símbolo em perguntas inciadas por quando, onde, o quê, quem, porque, e como.

5.1.1.2.Checar se existem indicativos de negação no vocabulário mínimo que relacionem duas ou mais classes. Verificar se estas classes possuem um relacionamento do tipo disjunto (exemplo macho e fêmea)

5.1.1.2.1. Se verdadeiro, adicionar o disjoint a lista de axiomas.

5.1.2. Caso o termo seja central a ontologia, classifique-o como classe (C)).

5.1.3. Caso contrário (o termo não é central para a ontologia) classifique-o como propriedade (R).

5.1.4. Verificar consistência.

Como dito nos itens 5.1.2 e 5.1.3, cabe ao usuário classificar o termo do *LAL* do tipo **estado** como **conceito** ou **propriedade**. Caso a classificação seja **conceito**, então, a ferramenta o conduzirá da mesma forma se o termo fosse originalmente **sujeito** ou **objeto**, isto é, a ferramenta realizará os passos descritos no item 3 do processo; caso a classificação

seja **propriedade**, a ferramenta o conduzirá para o passo 4. Na Figura 5, ilustramos o passo classificatório.

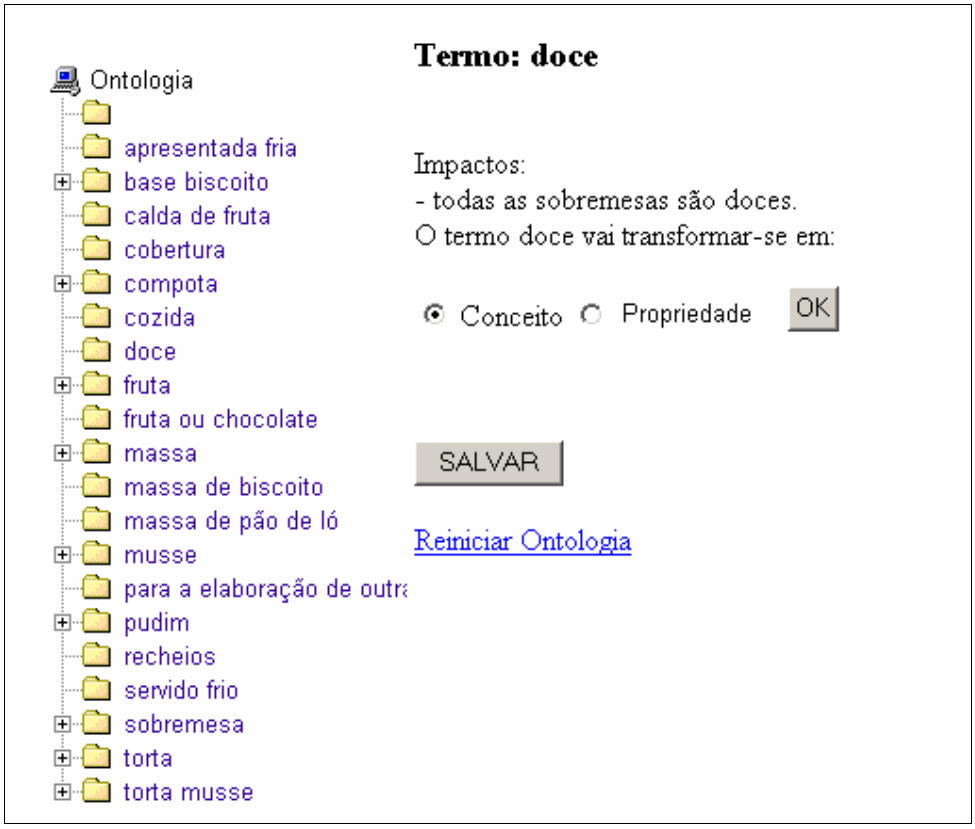

Figura 5 – Interface do C&L quanto à classificação dos termos do *LAL* do tipo Estado

Finalização do algoritmo:

6. Quando todos os termos tiverem sido adicionados à ontologia,
6.1. Checar se existe conjuntos de conceitos que compartilham rel idênticos
6.1.1. Para cada conjunto de conceito que compartilha rel, construir uma lista de conceitos separados
6.1.2. Buscar na ontologia conceitos que fazem referência a todos os membros desta lista
6.1.2.1.Se não forem encontrados, busca na noção e no impacto de cada membro da lista de conceitos tentando identificar um termo comum do vocabulário mínimo
6.1.3. Construir uma hierarquia de conceitos onde todos os membros da lista de conceitos é um sub-conceito do conceito encontrado em 6.1.2.
6.1.4. Verificar a consistência.

Concluídos os passos de identificação dos conceitos, propriedades e axiomas da ontologia a partir do *LAL*, é necessário que o usuário construa a hierarquia desses conceitos. Na figura 6, ilustramos a interface do C&L para criação de tal hierarquia. Observamos duas listas idênticas contendo todos os conceitos cadastrados na ontologia. O usuário deve

relacionar cada conceito do tipo *pai* (mais abstrato) da lista a esquerda com os conceitos do tipo *filho* (mais específico) na lista a direita.

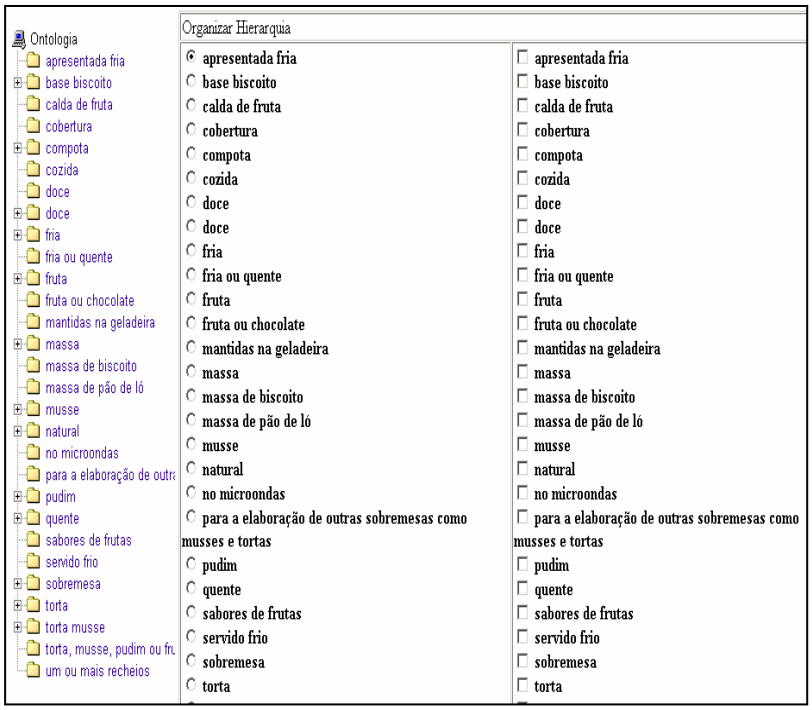

Figura 6 – Interface do C&L quanto à criação da hierarquia

Conforme descrito acima, a ferramenta foi construída de modo a minimizar o número de intervenções do usuário. Isto acontece porque a medida em que o processo vai avançando, novas informações são cadastradas e sugeridas nos passos posteriores. Mesmo na fase inicial, quando ainda não há conceitos nem propriedades cadastradas, a ferramenta consegue fazer algumas sugestões a partir do *LAL* e informações já obtidas. Por exemplo, na identificação de novos conceitos e sua propriedade, a ferramenta sugere como nova propriedade o restante do predicado sem o conceito já identificado. Sugestões mais precisas são realizadas com o avanço do processo. É responsabilidade do usuário apenas cadastrar a parte da propriedade sugerida referente à nova propriedade e o restante como o novo conceito do relacionamento.

4. Trabalhos Relacionados

Existe hoje no mercado uma série de ferramentas para a edição de ontologias, no entanto, não encontramos nenhuma que forneça apoio a um processo completo de forma que pessoas que não são especialistas possam desenvolver suas ontologias. A maioria das ferramentas disponíveis guiam o usuário apenas na modelagem do conhecimento do domínio esquecendo que é necessário antes elicitar este conhecimento. Tratam-se, basicamente, de editores de ontologias. A seguir, descrevemos brevemente algumas destas ferramentas e, em seguida, apresentamos uma análise do plug-in de ontologia desenvolvido para o C&L.

O OilEd [OilEd03] é um simples editor, o "NotePad" dos editores de ontologias. Ele utiliza as linguagens *DAML+OIL*. Sua intenção inicial é prover um simples editor que facilite o uso e estimule o interesse na linguagem *OIL*. Sua versão atual não provê um ambiente

completo de desenvolvimento de ontologias – não suporta o desenvolvimento destas em larga escala, migração e integração, versionamento, argumentação e muitas outras tarefas envolvidas no seu processo de construção. Simplesmente, permite ao usuário escrever ontologias e demonstra como usar o verificador FACT para checagem delas.

O OntoEdit [OntoEdit03] é um ambiente de engenharia de ontologias em que as fases de desenvolvimento são divididas da seguinte maneira: uma fase de especificação de requisitos, uma fase de refinamento e uma fase de avaliação. Na fase de especificação de requisitos, estes são coletados e devem descrever o que a ontologia dará suporte. Por natureza, essa tarefa é realizada pelos especialistas do domínio acompanhados pelos especialistas da modelagem. Essa fase também deverá gerar os subsídios que guiarão o engenheiro de ontologia na decisão sobre os conceitos relevantes e sua estrutura hierárquica na ontologia. Na fase de refinamento, uma ontologia madura é produzida e orientada à aplicação de acordo com a especificação dada na fase anterior. O engenheiro de ontologia pode desenvolver a hierarquia dos conceitos, propriedades e axiomas tão independente quanto seja possível da linguagem de representação concreta.

No OntoEdit é armazenado o modelo conceitual da ontologia de forma que seja possível fazer a transformação dessa representação conceitual para a maioria das linguagens de representação de ontologias como RDF(s), XML, DAML+OIL ou F-Logic. A fase de avaliação serve para provar a utilidade do desenvolvimento de ontologias e de seu ambiente de software associado. Nela, o engenheiro de ontologia checa se a ontologia corresponde às especificações do documento de requisitos.

O Protègè-2000 é um ambiente de plataforma independente e extensível para criação e edição de ontologias e bases de conhecimento. Permite a construção de ontologias de domínio, formulários de entrada de dados customizados e a entrada de dados. Possui uma biblioteca onde outros aplicativos podem acessar sua bases de conhecimento.

O Chimaera [Chimaera03] é um sistema de software que apoia o usuário na criação e manutenção de ontologias distribuídas na Web. Suas duas principais funções são: combinação de múltiplas ontologias juntas e diagnóstico de ontologias individuais ou múltiplas. Apoia o usuário nas tarefas de carregamento de bases de conhecimento em diferentes formatos, reorganização taxionômica, resolução de conflitos com nomes, busca de ontologias, edição de termos, etc. Sua intenção original era combinar fragmentos de bases de conhecimento.

No ambiente de desenvolvimento C&L, temos o *LAL* como uma base madura para o processo proposto em [Breitman03] de forma que uma quantidade razoável das tarefas de geração de ontologias são automatizadas, tendo a intervenção do usuário apenas naquelas em que haja absoluta necessidade. No entanto, ainda falta nesse ambiente uma interface gráfica amigável para a visualização das ontologias geradas e exportadores para as demais linguagens de representação. Hoje, o ambiente apenas fornece um link para o arquivo gerado em *DAML+OIL* que é interpretado pelo navegador em forma de *tags* estruturadas. Na Figura 7, mostramos parte de tal arquivo do exemplo no domínio de sobremesas.

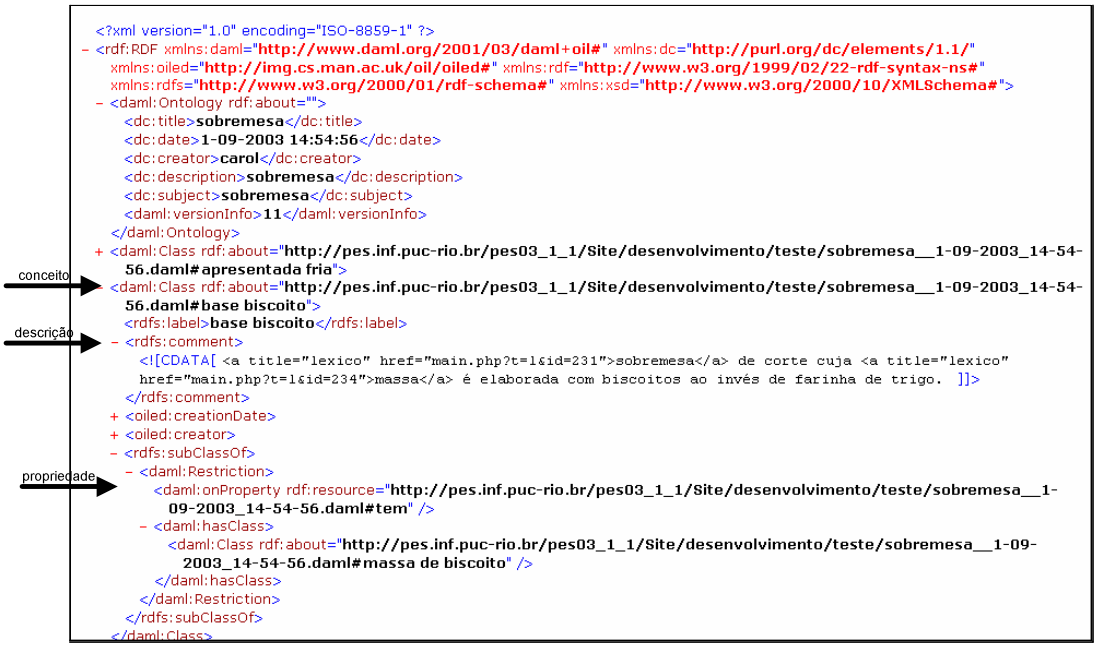

Figura 7 – Parte do arquivo em *DAML+OIL* no domínio sobremesa.

5. Conclusões e Trabalhos Futuros

A principal aplicação e benefício das ontologias consiste em prover semântica à informação disponibilizada através da Internet. Sua maior motivação é transformar os dados e aplicativos em elementos úteis, legíveis e compreensíveis para o software, ou, mais exatamente, para os agentes inteligentes, de forma a facilitar-lhes a comunicação dinâmica, a cooperação e o comércio eletrônico entre empresas (business-to-business ou b2b).

Neste artigo discutimos como a engenharia de requisitos pode auxiliar o processo de criação de ontologias. Introduzimos uma ferramenta que automatiza o processo de geração de ontologias baseadas no LAL, proposto em [Breitman03]. Esta ferramenta está integrada no projeto de software livre mantido pelo grupo de engenharia de requisitos da PUC-Rio. Neste projeto, estamos investigando técnicas de engenharia de requisitos segundo as premissas de desenvolvimento de software livre. Através dos softwares desenvolvidos pelo grupo, pretendemos disponibilizar e divulgar métodos, técnicas e ferramentas de qualidade a baixo custo.

Por ser uma ferramenta semi-automática, o plug-in desenvolvido permite que usuários de um domínio escrevam suas ontologias sem, necessariamente, conhecer as linguagens da tecnologia que lhes dá suporte. A tradução para o formato *daml* é automática e transparente para o usuário, que lida todo o tempo com uma interface gráfica, em linguagem natural.

Enfrentamos algumas dificuldades na implementação do plug-in devido, principalmente, ao tratamento da linguagem natural utilizada pelo *LAL*. Em nossa prática, notamos que é comum uma falta de consistência no vocabulário utilizado para capturar descrições em linguagem natural. Tipicamente, a utilização de tempos verbais distintos (gerúndio e infinitivo), singular versus plural e polisemia (utilização de uma palavra para designar mais de um conceito) foram observados. Este fato atrapalha o processo de geração de

ontologias e, em casos extremos, uma verificação seguida da uniformização dos termos do léxico foi necessária de modo a viabilizar o processo de geração de ontologias.

Também notamos que os usuários ao utilizarem a ferramenta tiveram uma certa dificuldade em identificar conceitos disjuntos entre termos do léxico. Acreditamos que esta dificuldade seja resultado da falta de familiaridade de alguns desses usuários com conceitos de lógica de primeira ordem. Acreditamos poder facilitar este aspecto através de exemplos concretos.

O plug-in desenvolvido continua em evolução. Estamos validando a ferramenta através de alguns projetos práticos. É necessário incluir novas funções de edição e exportação e um novo módulo para fornecer suporte automatizado ao processo de alinhamento de ontologias no contexto da Web Semântica. Nosso objetivo é apoiar a interoperabilidade de agentes de software que atuam em ambientes heterogêneos, regulados por um grande número de ontologias. Através do processo de alinhamento, esperamos prover ontologias que sirvam no compartilhamento de dados permitindo transações entre os agentes de software no contexto dessa Web Semântica. Acreditamos que o fato de estarmos utilizando o Léxico Ampliado da Linguagem, como base no processo de geração de ontologias, possa fornecer subsídios que facilitem a identificação de similaridades e diferenças entre ontologias que disponibilizem esta representação.

Agradecimentos

O projeto C&L é resultado do trabalho colaborativo entre: Grupo de desenvolvedores da **primeira versão do C&L** (grupo TôSobrando) – Alexandre Albuquerque, Alexandre Florio, Bernardo Franco, Fabio Souza, Felipe Nogueira, Leone Masiero, Marcelo Paes, Olavo Castro, Omar Paranaiba. Grupo de desenvolvedores da **segunda versão do C&L** – Carolina Howard Felicissimo, Cláudio Sant'Anna, Diva de Souza, Geórgia Gomes, Kleyna Moore, Leonardo Almaraz, Lyrene Fernandes, Miriam Sayão, Paulo Bastos, Reubem Alexandre, Richard Werneck, Roberto Christoph, William Oliveira. Grupo de desenvolvedores da **terceira versão do C&L** – Alex Campos, Fabio Nakamura, Fabio Guerra, Gabriel Bustamante, Guilherme Alvarenga, Gustavo Wagner, Jeronimo Venetillo, Pedro Fernandes, Rafael Rizzato. Os professores – Karin Breitman e Julio Cesar Leite.

Referências Bibliográficas

[Berners-Lee02] – Berners-Lee, T.; Lassila, O. Hendler, J. – The Semantic Web – Scientific American - http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html

[Breitman03] - Breitman, K.K,; Leite, J.C.S.P.- Lexicon Based Ontology Construction - 2nd. International Workshop on Software Engineering for Large Scale Multi Agent Systems - SELMAS - ACM computer Press, Portland Oregon, 2003.

[Chimaera03] – Disponível em <http://www.ksl.stanford.edu/software/chimaera>. Acesso em 17 de Setembro de 2003.

[Christoph03] – Christoph, R. H.; Felicíssimo, H. C.; Leite, J.C - C&L: Uma Ferramenta de Edição e Visualização de Cenários e Léxicos – Submetido para a seção de ferramentas do 17º Simpósio Brasileiro de engenharia de Software – Manaus, Outubro, 2003 – Disponível em <http://www.er.les.inf.puc-rio.br/grupoer/trabalhos/C&L.pdf>. Acesso em 03 de Setembro de 2003.

[Fellbaum98] - Fellbaum, C.; ed - WordNet: An electronic Lexical Database - Cambridge, MA - MIT Press - 1998.

[Fensel01] – Fensel, D. – Ontologie: a silver bullet for knowledge management and electronic commerce – Springer, 2001

[Gruber93] – Gruber, T.R. – A translation approach to portable ontology specifications – Knowledge Acquisition – 5: 199-220

[Gruninger95] – Gruninger, M.; Fox, M. – Methodology for the Design and Evaluation of Ontologies: Proceedings of the Workshop on basic Ontological Issues in Knowledge Sharing – IJCAI-95, Montreal, Canada, 1995.

[Guha90] - Guha, R. V., D. B. Lenat, K. Pittman, D. Pratt, and M. Shepherd. "Cyc: A Midterm Report." *Communications of the ACM* Vol.33 , No. 8 - August, 1990.

[Hendler01] - Hendler, J. – Agents and the Semantic Web – IEEE Intelligent Systems – March/April - 2001. pp.30-37

[Kaplan00] – Kaplan, G.; Hadad, G.; Doorn, J.; Leite, J.C.S.P. –Inspección del Lexico Extendido del Lenguaje– In Proceedings of the Workshop de engenharia de requisitos – WER'00 – Rio de Janeiro, Brazil – 2000.

[Leite93] Leite, J.C.S.P. and Franco, A.P.M. A Strategy for Conceptual Model Acquisition. First International Symposium on Requirements Engineering. **Proceedings.** IEEE Computer Society Press, 1993. pp. 243-246.

[Leite97] Leite, J.C.S.P, Rossi, G., Balaguer, F., Maiorana, V., Enhancing a requirements baseline with scenarios. In: Third IEEE International Symposium on Requirements Engineering - RE97, **Proceedings**. IEEE Computer Society Press, January, 1997, pp 44-53.

[Maedche02] – Maedche, A. – Ontology Learning for the Sematic Web – Kluwer Academic Publishers – 2002.

[Noy01] – Noy, N.; McGuiness, D. – Ontology Development 101 – A guide to creating your first ontology – KSL Technical Report, Standford University, 2001.

[OilED03] - Disponível em <http://oiled.man.ac.uk/>. Acesso em 17 de Setembro de 2003.

[OntoEdit03] - Disponível em <http://www.ontoprise.de/products/ontoedit_en>. Acesso em 17 de Setembro de 2003.

[Protègè2000] - Disponível em <http://protege.stanford.edu/>. Acesso em 17 de Setembro de 2003.

[W3Consortium03] – Disponível em <http://www.w3.org>. Acesso em 17 de Setembro de 2003.