

Ontologias

Internet

- Grande volume de informações desestruturadas adequadas apenas ao entendimento humano
- HTTP e HTML asseguram apenas navegação e apresentação
- Solução procedimental: Engenhos de busca
 - ▣ Análise apenas em nível léxico
 - ▣ Falta de precisão e muitos resultados irrelevantes,
- Principais ausências
 - ▣ Falta de contexto
 - ▣ Falta de semântica na definição das páginas

Soluções possíveis para o tratamento de informação na Web

- Dotar os sistemas de inteligência
 - Agentes inteligentes
 - *Manipulação cooperativa de informação*: distribuição, cooperação e comunicação sobre a *semântica* das páginas
 - Restrição de domínios
 - Dotar a Internet de inteligência: a Web Semântica
 - Linguagens e padrões para definir páginas com uma semântica clara e definida formalmente
 - Os agentes poderão raciocinar e “conversar” no contexto desta semântica
- => Ontologias desempenham um papel fundamental em ambas as soluções!

Ontologias

- Em Filosofia:
 - Ciência do ser (Aristóteles).
 - Tenta responder as questões:
 - O que é *ser* ?
 - Quais as características comuns a todos os seres?
 - estudo do que existe no mundo
- Em IA:
 - especificação das classes, objetos e relações da realidade
- Representação do vocabulário de um determinado domínio e suas conceituações.

Definições de ontologia (cont.)

- “Uma ontologia é uma especificação explícita e formal de uma conceitualização compartilhada” [Studer et al 98]
- *Especificação explícita*: definições declarativas de conceitos, instâncias, relações, restrições e axiomas
- *Formal*: declarativamente definida, sendo compreensível e *manipulável* para agentes e sistemas
- *Conceitualização*: modelo abstrato de uma área de conhecimento ou de um universo limitado de discurso
- *Compartilhada*: conhecimento consensual, seja uma terminologia comum da área modelada, ou acordada entre os desenvolvedores dos agentes que se comunicam

Define um vocabulário comum para pesquisadores que precisam compartilhar informação em um domínio.

Ontologias para quê?

- Para compartilhar conhecimento comum sobre a estrutura da informação:
 - Entre pessoas
 - Entre agentes de software
- Para permitir a reutilização do conhecimento sobre um domínio:
 - Para evitar a “reinvenção da roda”
 - Para introduzir padrões que permitam a interoperabilidade entre aplicações
- Separar conhecimento do domínio do conhecimento operacional
- Outros (análise do conhecimento do domínio, etc)

Onde Aplicar Ontologias?

- Recuperação de informações,
- Bibliotecas digitais,
- Web semântica,
- Gestão de conhecimento,
- Processamento da linguagem natural,
- Comércio eletrônico,
- Sistemas multiagentes...

Vantagens das ontologias

- Reuso massivo de conhecimento
 - ▣ Incorporação de conhecimento é facilitada, inclusive de linguagem natural
- Facilidades de acesso ao conhecimento
 - ▣ Via browser
 - ▣ Servidores
- Interoperabilidade entre formalismos
 - ▣ Tradução
 - ▣ Mapeamento
- Comunicação em nível de conhecimento

Elementos de uma ontologia

- Conceitos em um domínio de discurso (Classes)
 - ▣ Vinho, Vinho Tinto, Vinho Branco, Vinho Rosé
- Propriedades dos conceitos (slots/atributos)
 - ▣ Cor, tanino, encorpado, uva, etc
- Restrições sobre as propriedades (facetas)
 - ▣ tipo, cardinalidade,...
- Instâncias de conceitos

Ontologia + instâncias de classes = **Base de Conhecimento**

Tipos de Ontologias (quanto ao conteúdo)

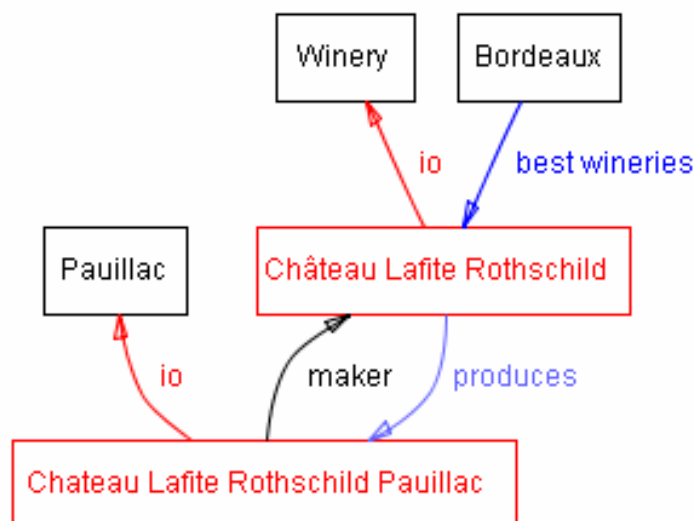
- **Ontologias de Representação:** definem as primitivas de representação - como *frames*, axiomas, atributos e outros – de forma declarativa.
- **Ontologias Gerais:** contêm definições abstratas necessárias para a compreensão de aspectos do mundo, como tempo, processos, papéis, espaço, seres, coisas, etc.
- **Ontologias genéricas:** conceitos mais genéricos e abstratos, geralmente comuns a vários campos.
- **Ontologias de domínio e de aplicação:** tratam de um domínio específico de uma área genérica de conhecimento, como direito tributário, microbiologia, etc.

Outras classificações em (Almeida, 2003)

Como construir uma ontologia

- Determinar o domínio e o alcance
- Enumerar os termos importantes
- Definir classes e hierarquias
- Definir atributos e relações
- Definir restrições (cardinalidade, tipo...)

Vinhos e Vinícolas



Processo de desenvolvimento de ontologia

Neste exemplo:



Na realidade – um processo iterativo:



Ontology Engineering versus Object-Oriented Modeling

Uma ontologia

- reflete a estrutura do mundo
- é às vezes sobre a estrutura dos conceitos
- a representação física real não é o caso

Uma estrutura de classe OO

- reflete a estrutura do dado e do código
- geralmente sobre o comportamento (métodos)
- descreve a representação física do dado (long int, char, etc.)

1) Determinar Domínio e Escopo



- Qual é o domínio que a ontologia irá cobrir?
- Para que nós estaremos usando a ontologia?
- Quais tipos de questões a informação na ontologia deve prover respostas (*questões de competência*)?
- Exemplo: Ontologia de vinhos
 - ▣ Serverá para uma aplicação que pretende verificar combinação de vinhos e comida?
 - Diferentes tipos de vinhos, principais tipos de comidas, noção de boa combinação vinho e comida.
 - ▣ Ou
 - ▣ Processamento natural de artigos em revistas especializadas em vinhos?
 - Sinônimos, part-of-speech
- *Respostas para estas questões podem mudar durante o ciclo de vida*

1.a) Questões de Competência

- Uma maneira de determinar o escopo de uma ontologia é determinar uma lista de questões que a base de conhecimento ontológica deverá responder:
 - ▣ Quais características de vinho eu devo considerar quando escolher um vinho?
 - ▣ Bordeaux é um vinho tinto ou branco?
 - ▣ Cabernet Sauvignon vai bem com frutos do mar?
 - ▣ Qual a melhor escolha de vinho para carne grelhada?
 - ▣ Quais as características de um vinho que afetam sua relação com um prato (comida)?
 - ▣ O sabor ou corpo de um específico vinho muda com o ano da safra?
 - ▣ Qual foi a melhor safra do Napa Zinfandel?
- A ontologia contém informação suficiente para responder a estas questões?
- As respostas requerem um nível particular de detalhe ou representação de uma determinada área?

2) Considerar Reuso



- Porquê reusar outras ontologias?
 - ▣ economizar esforço
 - ▣ interagir com as ferramentas que usam outras ontologias
 - ▣ usar ontologias que tenham sido validadas através do uso em aplicações
- Se a ontologia a ser reusada está representada em outro formalismo, isso não é um problema:
 - ▣ Existem várias ferramentas para importar e exportar ontologias que trabalham com diversos formalismos.

2.a) O que Reusar?

- Bibliotecas de Ontologias
 - ▣ DAML ontology library (www.daml.org/ontologies)
 - ▣ Google ☺
- Ontologias Upper
 - ▣ IEEE Standard Upper Ontology (suo.ieee.org)
 - ▣ Cyc (www.cyc.com)
- Ontologias de domínio-específico
 - ▣ Unified Medical Language System (UMLS) Semantic Net (umlsinfo.nlm.nih.gov)
 - ▣ GO (Gene Ontology) (www.geneontology.org)

2.a) O que Reusar?(II)

□ Ontologia Geral

- DMOZ (www.dmoz.org)

- WordNet (www.cogsci.princeton.edu/~wn/)

redes semântica de 70100 significados

de 95600 palavras (57000 substantivos) do **inglês**

organizados em 4 categorias sintática: substantivos, verbos, adjetivos e advérbios,

são agrupados em conjuntos de sinônimos

resultado de 15 anos de desenvolvimento manual por time de psico-lingüistas

EuroWordNet: <http://www.hum.uva.nl/~ewn/>

Resultado de um grande projeto da comunidade europeia

Versão multi-língua adicionando e inter-ligando WordNets do

Espanhol, Francês, Italiano, Alemão, Holandês, Tcheco e Estoniano ao do Inglês

3) Enumerar termos importantes

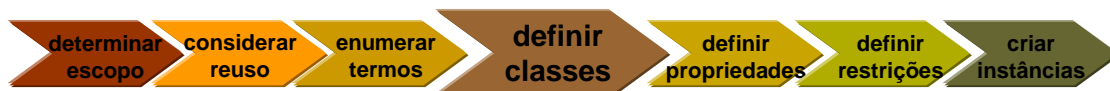


- Quais são os termos que são falados?
- Quais são as propriedades destes termos?
- O que nós queremos dizer sobre os termos?
 - Fazer lista sem se preocupar com relações ou sinônimos

Enumerando Termos – A ontologia de vinho

*vinho, uva, vinícola, localização,
cor do vinho, corpo do vinho, sabor do vinho,
quantidade de açúcar
vinho branco, vinho tinto, vinho Bordeaux
comida, frutos do mar, peixe, carne, vegetais, queijo*

4) Definir Classes e hierarquia de Classes



- Escolher termos da lista criada na etapa anterior
- Uma classe é um **conceito** no domínio
 - ▣ uma classe de vinhos
 - ▣ uma classe de vinícolas
 - ▣ uma classe de vinhos tinto
- Uma classe é uma **coleção** de elementos com propriedades similares
- **Instâncias** de classes
 - ▣ uma garrafa de vinho da Califórnia para um almoço

Herança de Classes

- Classes geralmente constituem uma **taxonomia hierárquica** (uma hierarquia de subclasse-superclasse)
- Uma hierarquia de classe é geralmente uma hierarquia IS-A:

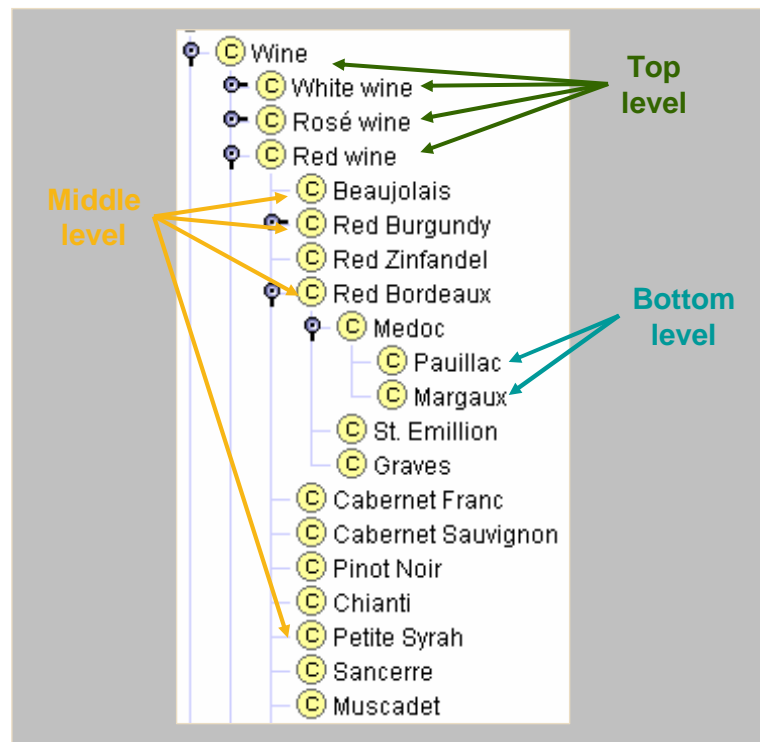
uma instância de uma subclasse é uma instância de uma superclasse

- Se você pensa uma classe como um conjunto de elementos, uma subclasse é um subconjunto

Herança de Classe - Exemplo

- Maçã é uma subclasse de Frutas
Toda maçã é uma fruta
- Vinho tinto é uma subclasse de Vinho
Todo vinho tinto é um vinho
- Vinho Chianti é uma subclasse de vinho tinto
Todo vinho Chianti é um vinho tinto

Níveis na Hierarquia



Modos de Desenvolvimento

- top-down – define os conceitos mais gerais primeiro e então os especializa
- bottom-up – define os conceitos mais específicos e então os organiza em classes mais gerais
- combinação – define os conceitos mais salientes primeiro e então generaliza e os especializa

Documentação

- Classes (e propriedades) geralmente tem documentação
 - ▣ Descrever as classes em linguagem natural
 - ▣ Listar suposições relevantes do domínio para as definições de classe
 - ▣ Listar sinônimos
- Documentar classes e propriedades é tão importante quanto documentar código em programas!

5) Definir Propriedades de Classes – Slots



- Geralmente, classes não contém informações suficientes para responder as perguntas de competência.
 - ▣ Precisamos de propriedades
- Propriedades (slot) em uma classe descreve atributos de instâncias da classe e relações a outras instâncias

Cada vinho terá cor, quantidade de açúcar, produtores, etc.

Propriedades (Slots)

- Tipos de propriedades
 - ▣ propriedade “intrínseca” : **sabor** e **cor** do vinho
 - ▣ Propriedade “extrínseca” : **nome** e **preço** do vinho
 - ▣ partes: **ingredientes** em um prato
 - ▣ relações com outros objetos: **produtores** de vinho (vinícola)
- Propriedades simples e complexas
 - ▣ propriedade simples (atributos): contém valores primitivos (strings, numbers)
 - ▣ propriedade complexas: contém (ou aponta para) outros objetos (e.g., uma instância de vinícola)

Propriedade e herança de Classe

- Uma subclasse herda todos as propriedades (slots) da superclasse

Se um vinho tem o mesmo nome e gosto, um vinho tinto também tem o mesmo nome e gosto
- Se uma classe tem múltiplas superclasses, ela herda propriedades de todas

Port é tanto um vinho suave quanto um vinho tinto. Ele herda “conteúdo de açúcar: high” do primeiro e “cor: tinto” do último

6) Definir Restrições de Propriedade (facetadas dos slots)



- Restrições de propriedade descrevem ou limitam o conjunto de valores possíveis para um slot

O nome de um vinho é uma string

O produtor de vinho é uma instância de vinícola

Uma vinícola tem uma localização explícita

Restrições comuns

- Cardinalidade da propriedade – o número de valores que uma propriedade tem
- Tipo de valor da propriedade – o tipo de valor que uma propriedade tem
- Valores mínimo e máximo – um range de valores para uma propriedade numérica
- Valor default – valor que uma propriedade tem caso não explicitamente especificado

Restrições comuns:

Cardinalidade da propriedade

- Cardinalidade
 - ▣ Cardinalidade N significa que a propriedade **deve** ter N valores
- Cardinalidade mínima
 - ▣ Cardinalidade mínima 1 significa que a propriedade deve ter um valor (requerido)
 - ▣ Cardinalidade mínima 0 significa que o valor da propriedade é **opcional**
- Cardinalidade máxima
 - ▣ Cardinalidade máxima 1 significa que a propriedade deve ter no máximo um valor (**slot de valor simples**)
 - ▣ Cardinalidade máxima maior do que 1 significa que a propriedade tem mais do que um valor (**slot de valor múltiplo**)

Restrições comuns:

Tipo de valor

- String: uma string de caracteres (“Château Lafite”)
- Número: um integer ou um float (15, 4.5)
- Boolean: uma flag true/false
- Tipo enumerado: uma lista de valores (high, medium, low)
- Tipo instância: uma instância de outra classe
 - ▣ Especifica a classe para qual as instâncias pertencem

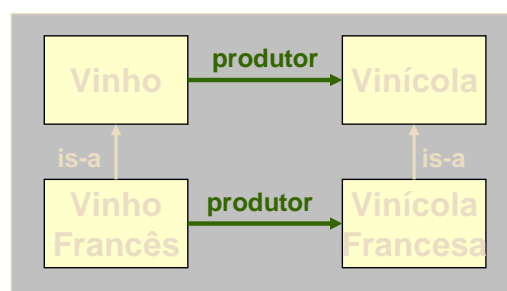
A classe vinho é o tipo de valor para a propriedade “produtores” da classe vinícola

Domínio e Range da propriedade

- **Domínio** de uma propriedade – uma classe (ou classes) que tem a propriedade
 - ▣ Mais precisamente: instâncias de classe (ou classes) da qual uma propriedade pode ter
 - *Produtor* é a classe domínio do slot *produz*.
- **Range** de uma propriedade – uma classe (ou classes) para qual valores podem pertencer a uma propriedade
 - ▣ *Vinho* é o range do slot *produz* na classe *Produtor*.

Restrições e heranças de Classe

- Uma subclasse **herda** todos as propriedades da superclasse
- Uma subclasse pode **sobrescrever** as restrições para “aproximar” a lista de valores permitidos
 - Tornar o range da cardinalidade menor
 - Substituir uma classe no range com uma subclasse



7) Criar Instâncias



- Escolher uma classe
- Criar uma instância de uma classe
 - ▣ A classe torna-se um **tipo direto** de instância
 - ▣ Qualquer superclasse do tipo direto é um **tipo** da instância
- Atribuir valores da propriedade para a instância
 - ▣ Valores das propriedades devem obedecer as restrições da mesma
 - ▣ Ferramentas de aquisição de conhecimento as vezes fazem isto

7) Criar Instâncias

Exemplo

- Chateau-Morgon-Beaujolais
 - ▣ Instância de Beaujolais, que é instância de vinho tinto, que é instância de Vinho
 - Body: Light
 - Color: Red
 - Flavor: Delicate
 - Tannin level: Low
 - Grape: Gamay (instância da classe Wine grape)
 - Maker: Chateau-Morgon (instância da classe Winery)
 - Region: Beaujolais (instância da classe Wine-Region)

7) Criar Instâncias

Exemplo

The screenshot shows a software window titled "Chateau Morgon Beaujolais (Beaujolais)". It contains several input fields and dropdown menus for defining a wine instance:

- Name:** A text field containing "Chateau Morgon Beaujolais".
- Area:** A dropdown menu showing "Beaujolais region".
- Body:** A dropdown menu showing "LIGHT".
- Color:** A dropdown menu showing "RED".
- Maker:** A dropdown menu showing "Chateau Morgon".
- Flavor:** A dropdown menu showing "DELICATE".
- Sugar:** A dropdown menu showing "DRY".
- Grape:** A dropdown menu showing "Gamay grape".
- Tannin Level:** A dropdown menu showing "LOW".

Produto Final do Projeto

- Descrição semi-formal da ontologia
- Termos, classes, hierarquias, propriedades, restrições, instâncias.
- Todos esses os elementos irão permitir o uso de uma ferramenta de software para a especificação da ontologia em alguma linguagem formal.

Protegé

- Protégé é um ambiente extensível e independente de plataforma para a criação e edição de ontologias e bases de conhecimento.
- Escrito em Java, utiliza uma máquina virtual para a execução em quase qualquer plataforma.
- Importa e exporta ontologias em diversos formatos, facilitando a reutilização e intercâmbio de ontologias.
 - ▣ Inclusive OWL

Download e Instalação

- A última versão do Protégé pode ser obtida em:
<http://protege.stanford.edu/>
- Um excelente FAQ pode ser encontrado em
<http://protege.stanford.edu/doc/faq.html>
- Para o download da versão para o Windows, recomenda-se usar o instalador automático oferecido pelo site.
- A versão para Windows, incluindo a máquina virtual Java tem mais de 50MB. É um software ainda algo “pesado” e emprega diversos plugins para obter funcionalidade.

Criando o Projeto

- Selecione “Create New Project” da Janela “Welcome” ou clique no botão “New Project” se esta já estiver fechada.
- Abre-se a janela para a seleção do tipo de arquivo. Escolha “Protégé Files (pont e pins)” e clique em “Finish”.
- Aparecem os painéis “Class Browser” à esquerda e “Class Editor” à direita.
- No box “Class Hierarchy” aparecem THING e SYSTEM_CLASS

Salvando e nomeando o Projeto

- Selecione o botão “Save Project”. Use o navegador para selecionar uma pasta e dar um nome ao Projeto (pprj).
- Os arquivos de classes (pont) e de instâncias (pins) são nomeados automaticamente.
- Tecle OK. O projeto foi salvo. Confira o nome na barra da janela. Voltam o browser e o editor.

Criando Classes

- Na hierarquia de classes selecione `SYSTEM_CLASS`.
- Clique o botão “Create Class”. Uma nova classe é criada, como classe do sistema, e automaticamente nomeada.
- Passe para o painel “Class Editor”. Modifique ou acrescente a informação que quiser. Aproveite para criar os “Template Slots”
- Use o mesmo método para criar outras classes e subclasses.

Criando Slots

- Clique na aba “Slots”. Como anteriormente, há um painel “Slot Browser” e outro “Slot Editor”.
- Na hierarquia de slots escolha o tipo de slot apropriado (p.ex: `ANNOTATED_INSTANCE`).
- Passe para a janela “Slot Editor”. Faça as modificações e as entradas desejadas.
- Use o mesmo método para criar slots de cada classe.

Criando Instâncias

- Clique na aba “Instances”. Aparecem os painéis “Class Browser”, “Instance Browser” e “Instance Editor”.
- Escolha a classe na hierarquia de classes. Somente classes concretas podem ter instâncias.
- Clique em “Create Instance”. Edite a instância para refletir a informação desejada. Repetir para as demais instâncias.

Criando Formulários

- Na aba “Forms” é possível construir formulários para a entrada de instâncias on-line.
- Selecione a classe e o slot desejados e use o editor para configurar e posicionar a correspondente região do formulário.

Criando e Salvando Consultas

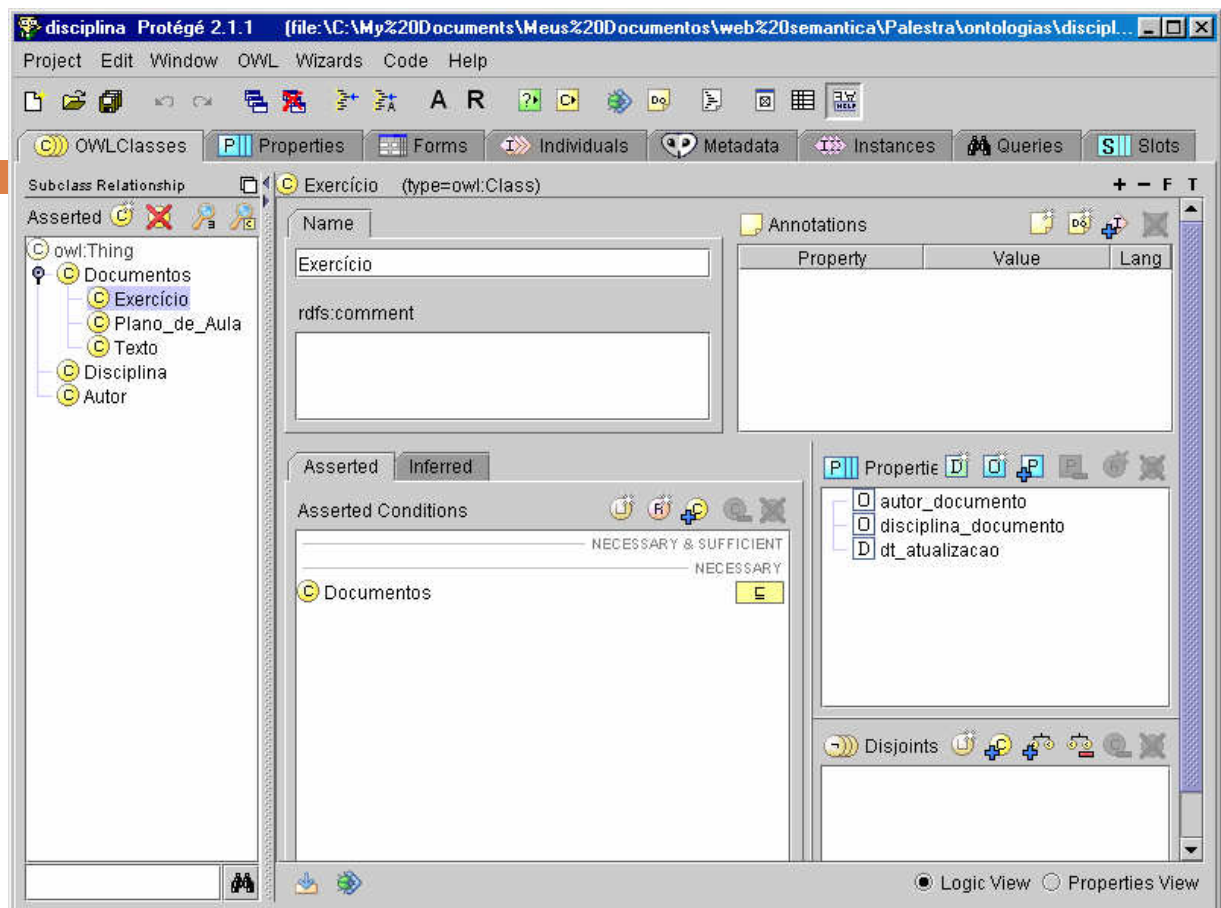
- Use a aba “Queries” para formular e salvar consultas à ontologia.
- Após a formulação e teste, a consulta pode ser salva em uma biblioteca de consultas através do botão “Add to Query Library”.

Ontologias importadas pelo Protégé

- Não deixe de consultar:
<http://protege.stanford.edu/download/ontologies.html>.
- Qualquer uma dessas ontologias pode ser lida e editada com o Protégé.
- Outras fontes:
 - DAML ontology library
<http://www.daml.org/ontologies/>
 - Ontolingua Ontology Library
<http://www.ksl.stanford.edu/software/ontolingua>
- O Protégé tem suporte nativo para .rdfs e .owl
- Outros formatos → através de plugins.

Plugins do Protégé

- ❑ O Protégé tem muitos plugins:
<http://protege.stanford.edu/download/plugins.html>.
- ❑ Qualquer um desses plugins pode ser adicionado como uma nova funcionalidade ao Protégé.
- ❑ Entre eles recursos de compatibilidade com xml, xmls, daml+oil, visualização gráfica e muitos outros.
- ❑ O Protégé tem suporte nativo para .rdfs e .owl



Exemplo de ontologia

Desenvolver uma ontologia de web semântica a ser utilizada para material didático disponibilizado na web por professores universitários.

Compartilha-se existência de..

- ▣ Tipos de Documentos:
 - Planos de Aula;
 - Exercícios;
 - Textos;
- ▣ Informações sobre o autor:
 - Nome
 - Titulação
 - Instituição com Vínculo
- ▣ Informações sobre o documento:
 - Tipo
 - Data de publicação
 - Nome da disciplina
 - Autor

1- Enumerar os termos importantes

- Documentos:
 - Planos de Aula;
 - Exercícios;
 - Textos;
- Informações sobre os documentos:
 - Data de atualização
 - Disciplina do documento
 - Autor do documento

- Autor:
- Informações sobre autor
 - Nome Autor
 - Titulação
 - Vinculo
- Disciplina
- Informações sobre a Disciplina
 - Código da disciplina
 - Nome da disciplina

2-Definir classes e hierarquias

- Existem as seguintes classes de objetos:
 - ▣ Documentos
 - ▣ Exercícios
 - ▣ Planos de Aula
 - ▣ Texto
 - ▣ Autor
 - ▣ Disciplina
- Exercícios, Planos de aula e textos são especializações da classe documentos. Autor e disciplina constituem classes separadas.

3-Definir atributos e relações

- A classe documentos e suas especializações possuem os seguintes atributos:
 - ▣ Data de atualização
 - ▣ Disciplina do documento
 - ▣ Autor do documento
- A classe Autor tem os seguintes atributos:
 - ▣ Nome Autor
 - ▣ Titulação
 - ▣ Vinculo
- A classe Disciplina tem os seguintes atributos:
 - ▣ Código da disciplina
 - ▣ Nome da disciplina

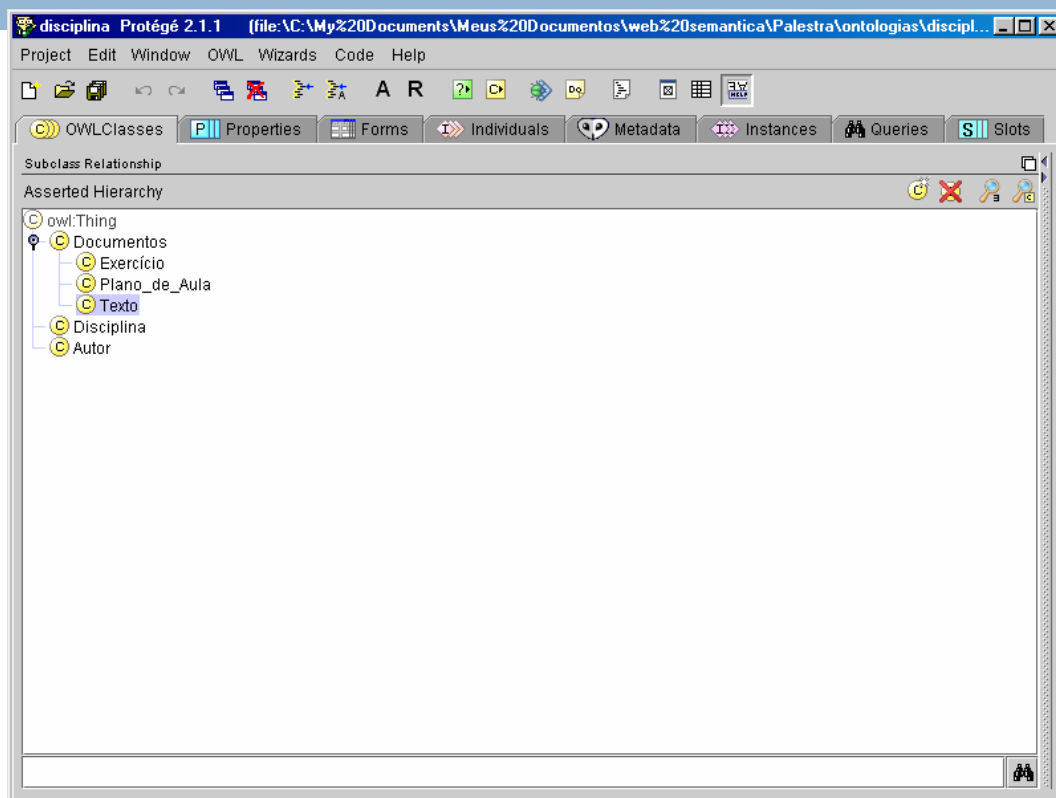
4- Definir restrições (cardinalidade, tipo...)

Atributo	Tipo	Cardinalidade
Data_atualização	String	1
Disciplina_documento	Objeto Disciplina	Múltipla
Autor_documento	Objeto Autor	1
Nome_Autor	String	1
Titulação	String	1
Vinculo	String	1
Codigo_disciplina	Inteiro	1
Nome_disciplina	String	1
inverso_disciplina_documento	subclasses de documentos	Múltipla

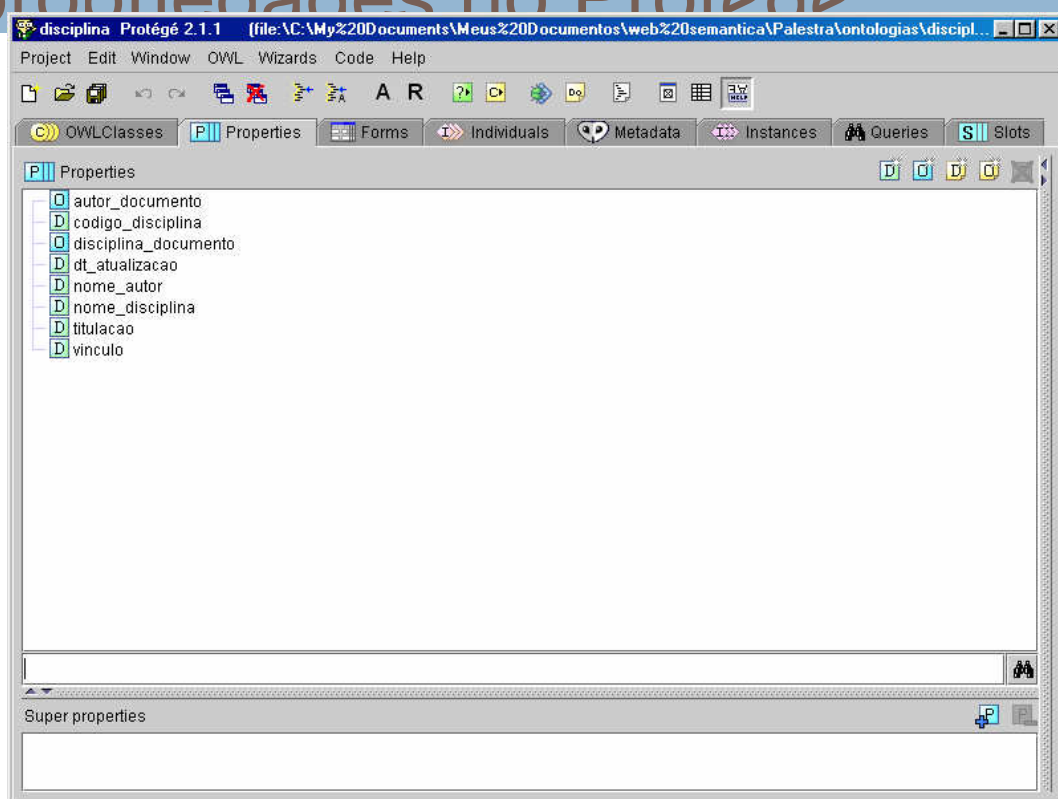
Desenvolvimento no
Protégé



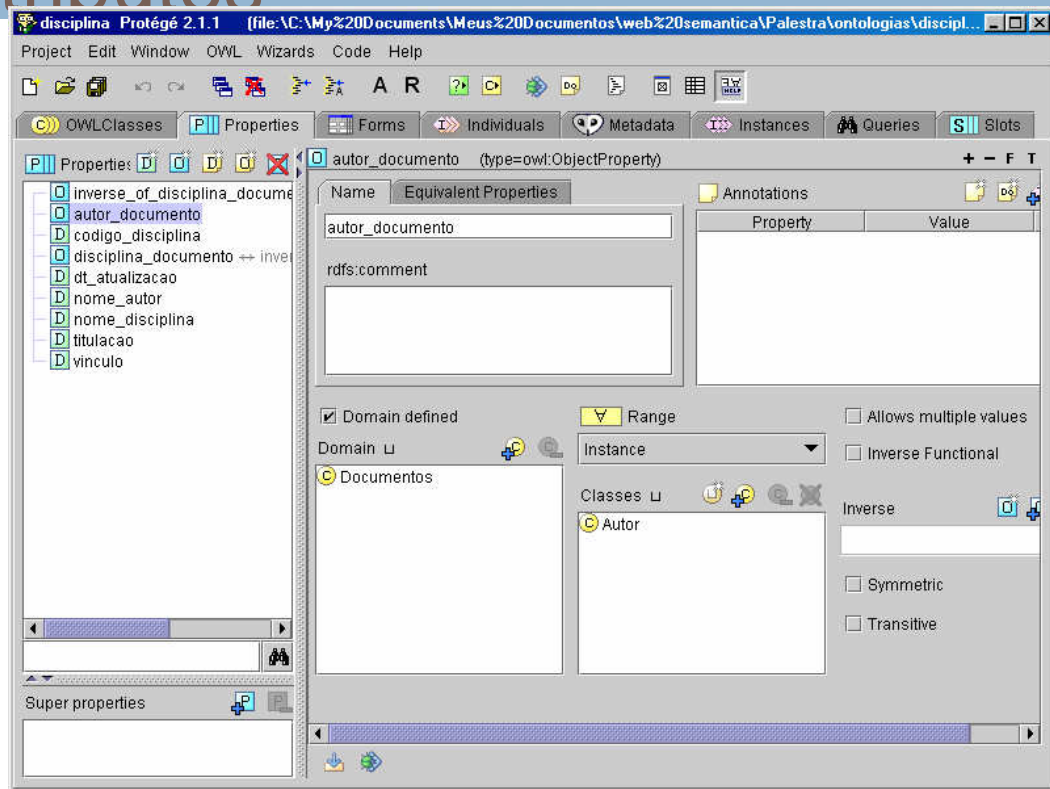
Criando as classes:



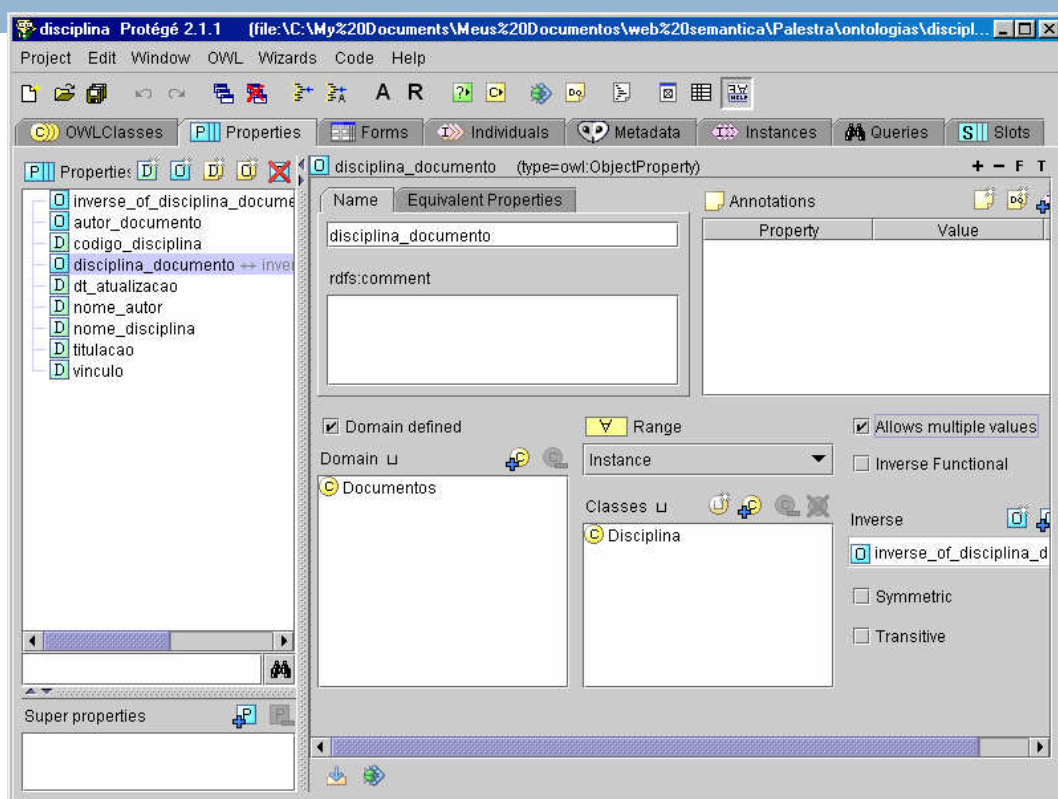
Lista dos atributos, chamados de propriedades no Protégé



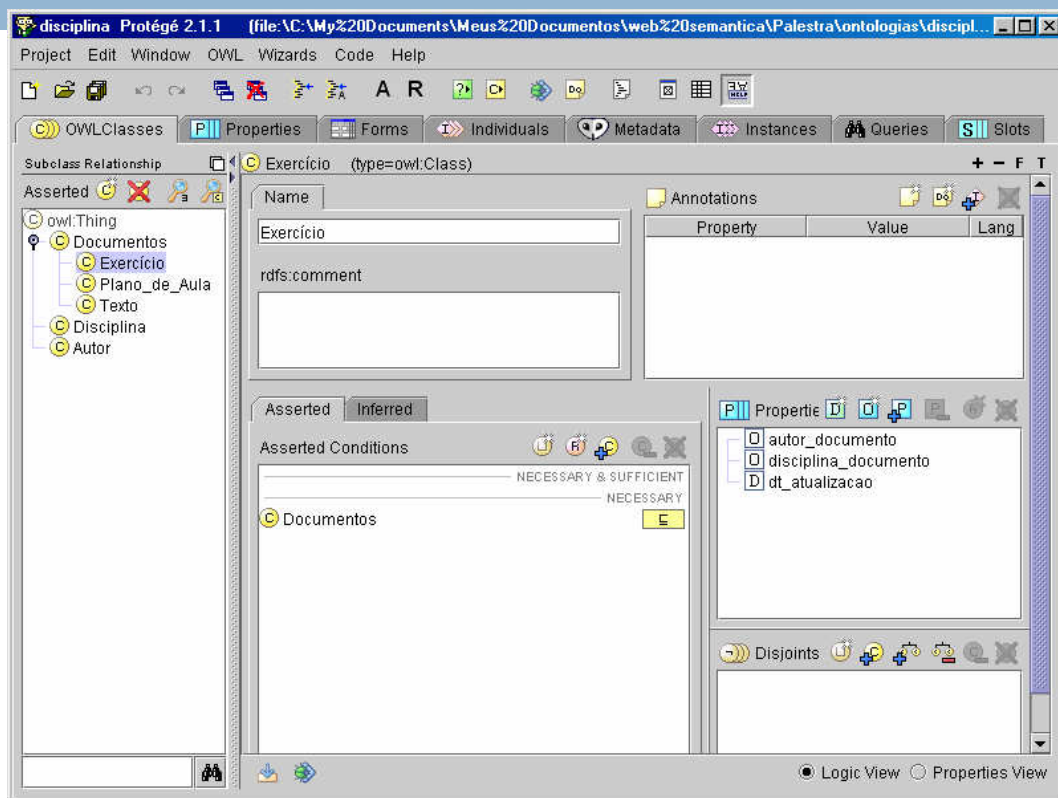
Cardinalidade e tipos dos atributos



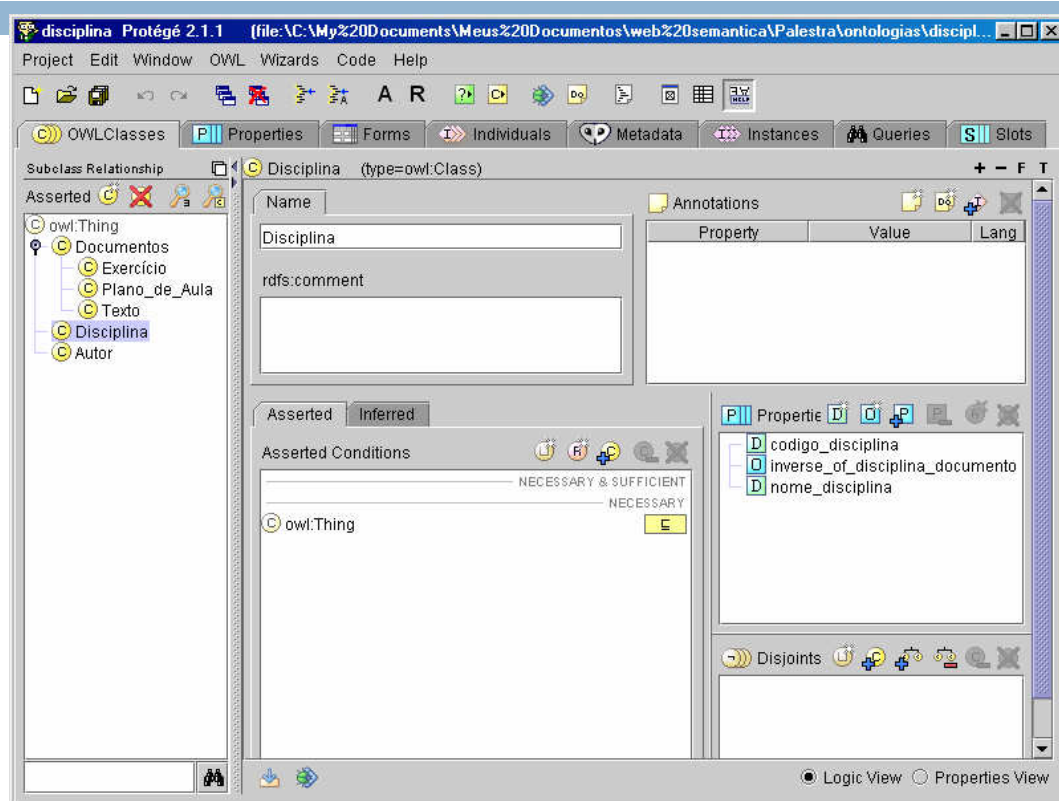
Atributo com inverso:



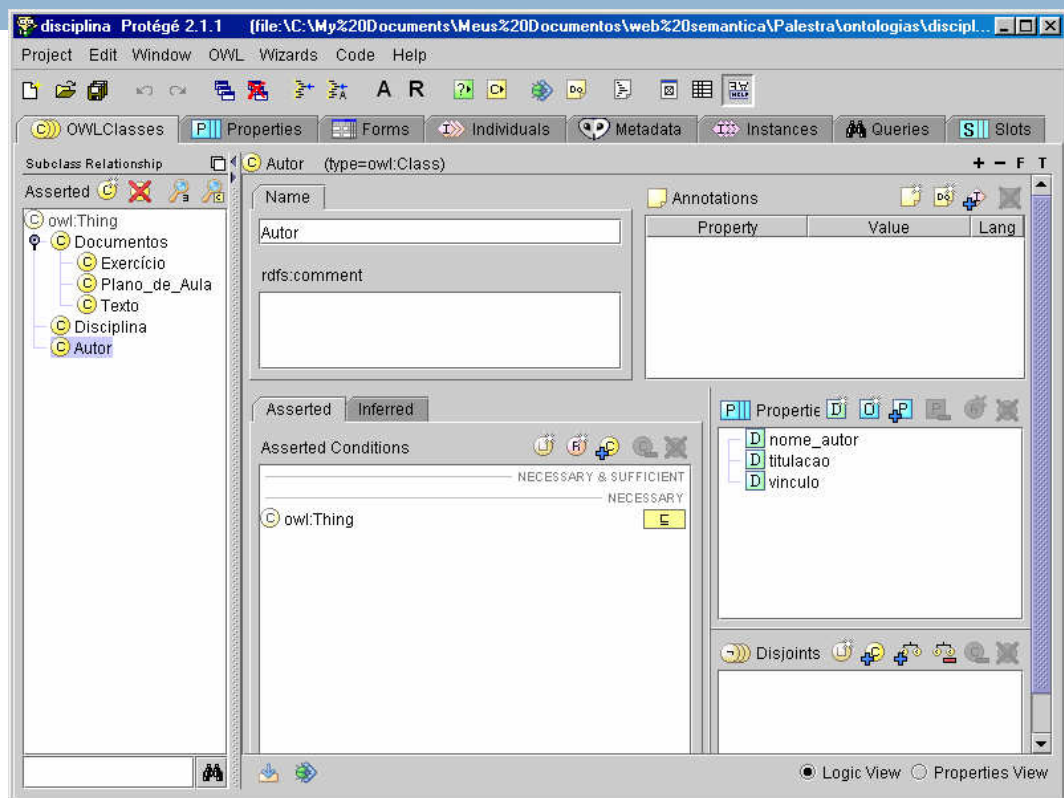
Incluindo os atributos nas classes



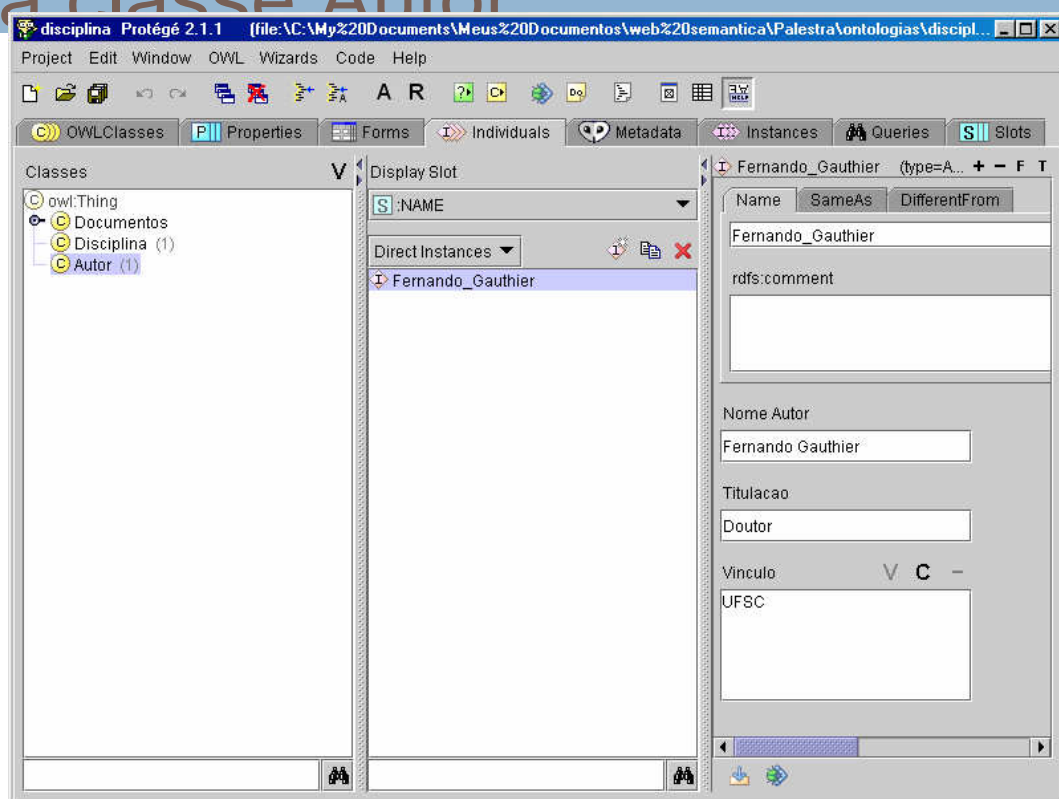
Incluindo os atributos nas classes



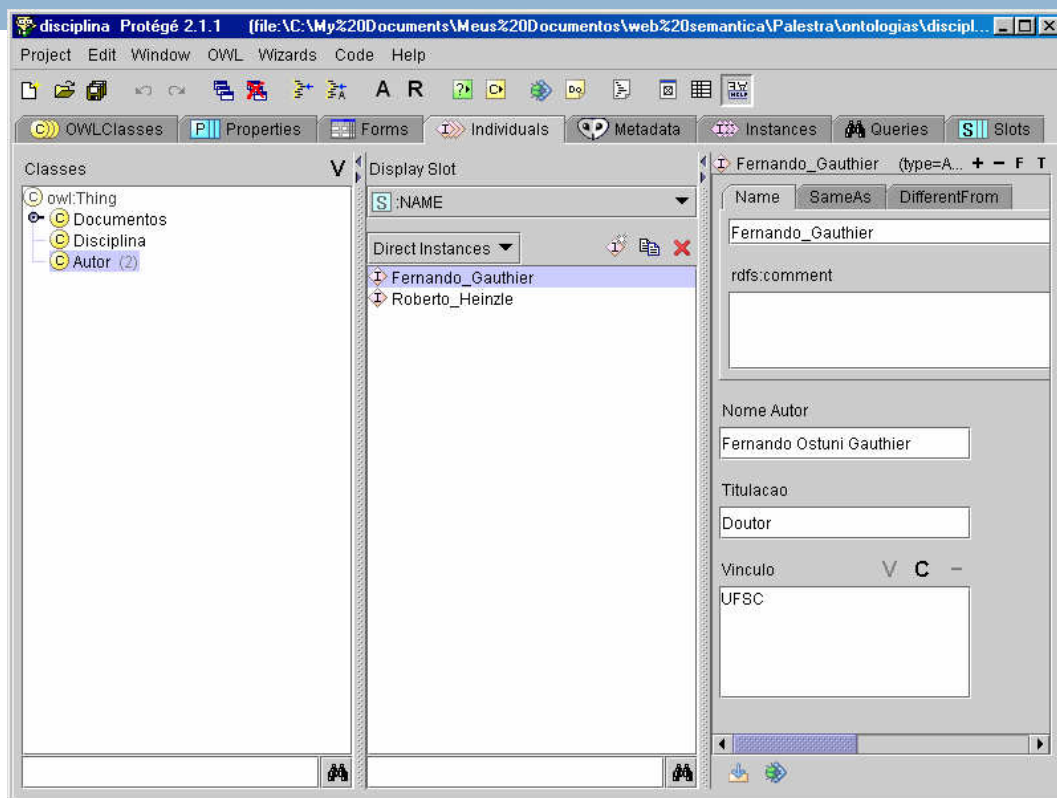
Incluindo os atributos nas classes



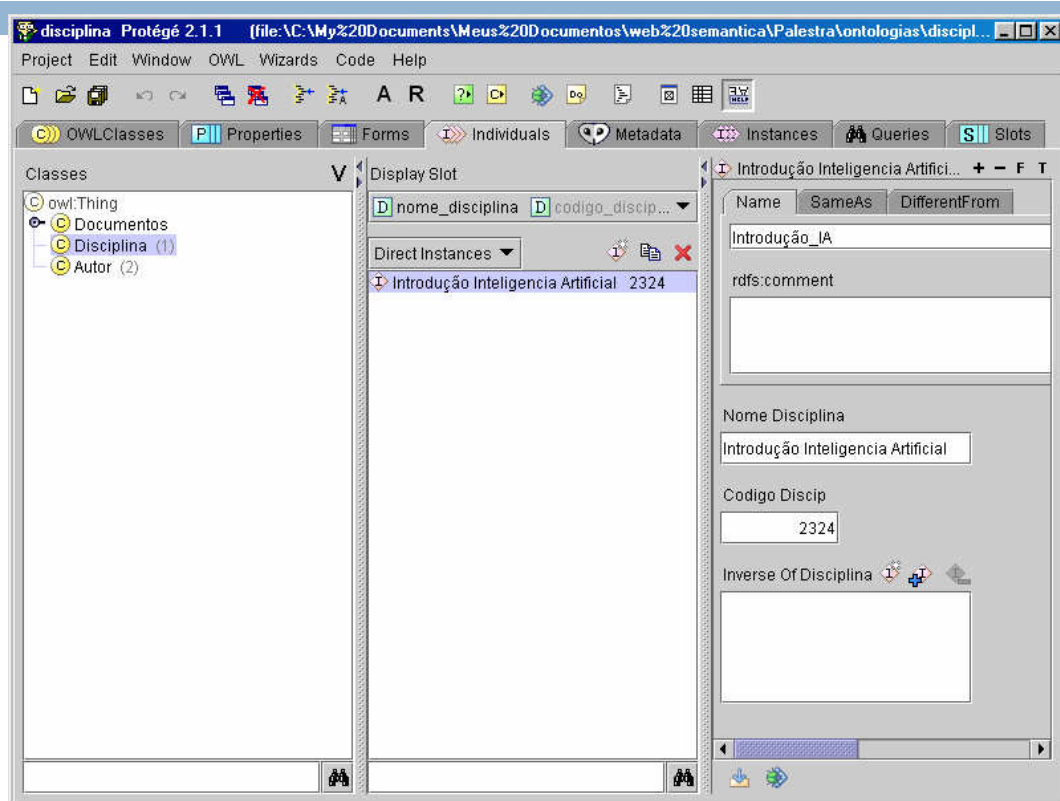
Criando instancias ou indivíduos da classe Autor



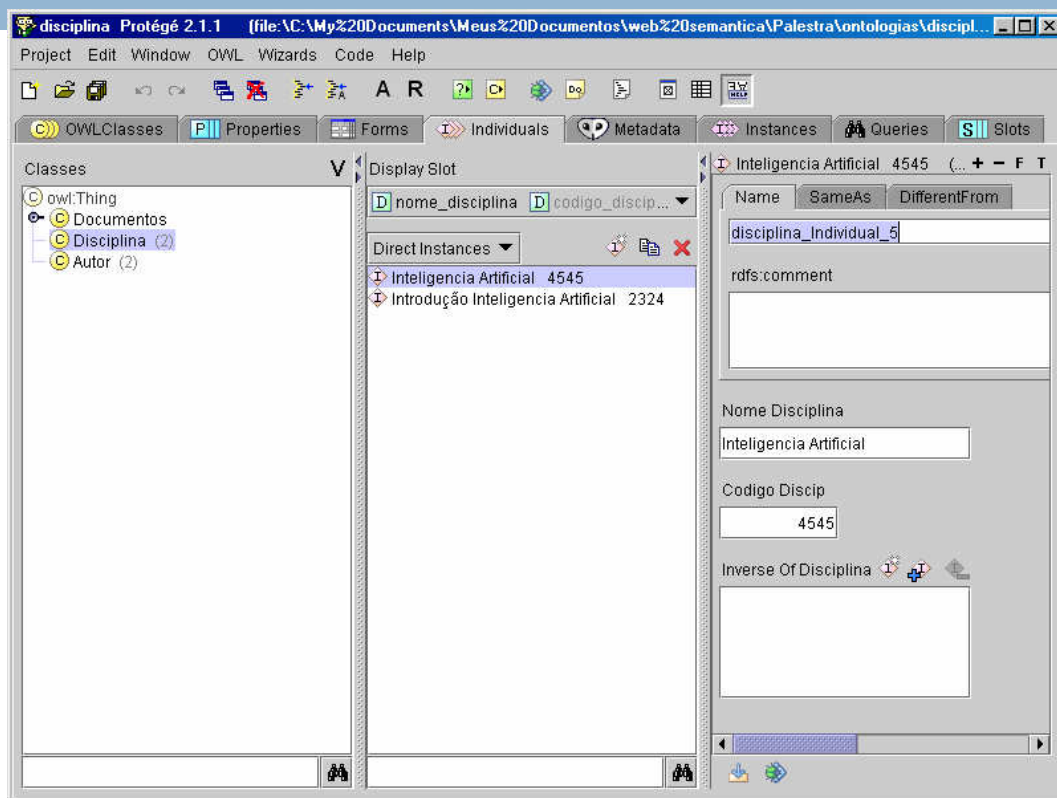
Mais uma instância de autor



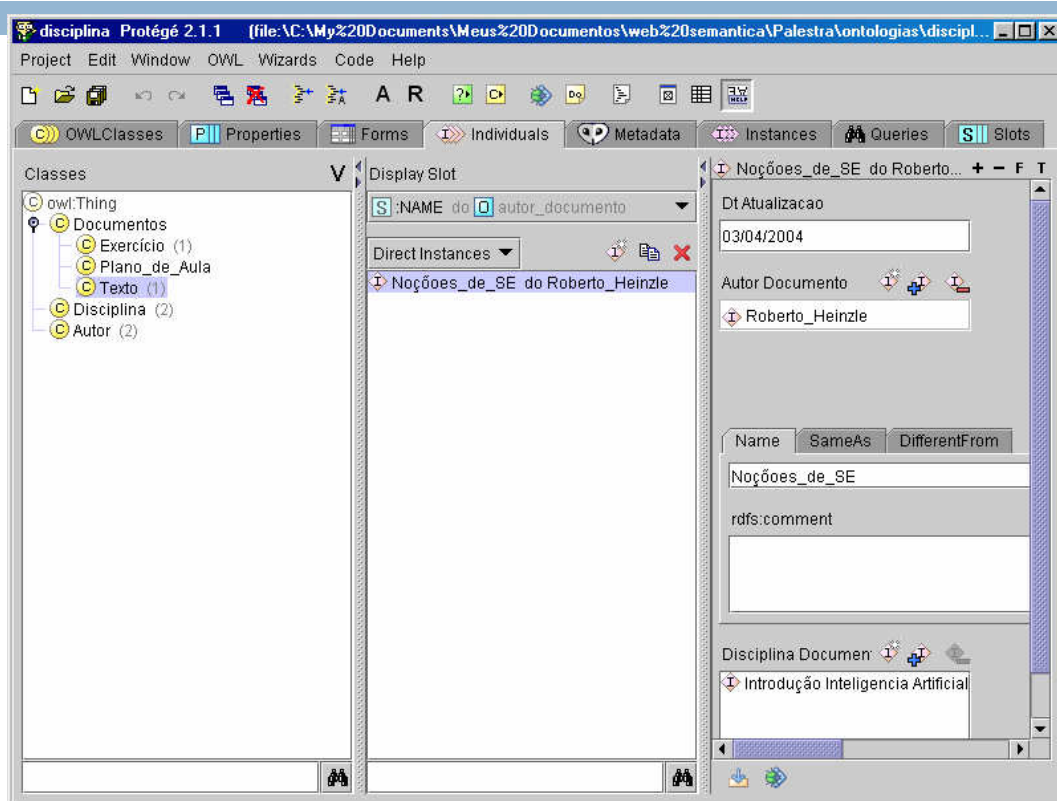
Criando uma disciplina



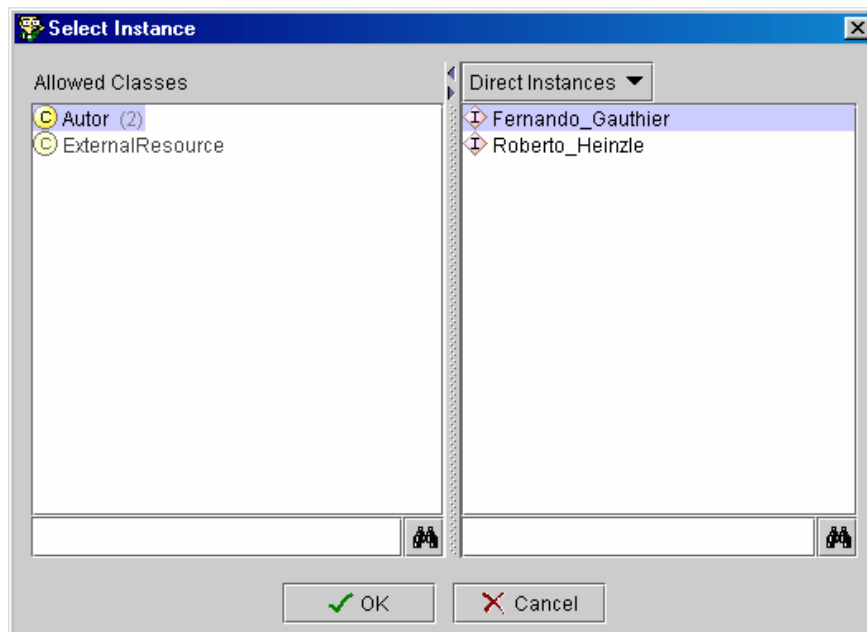
Criando outra disciplina



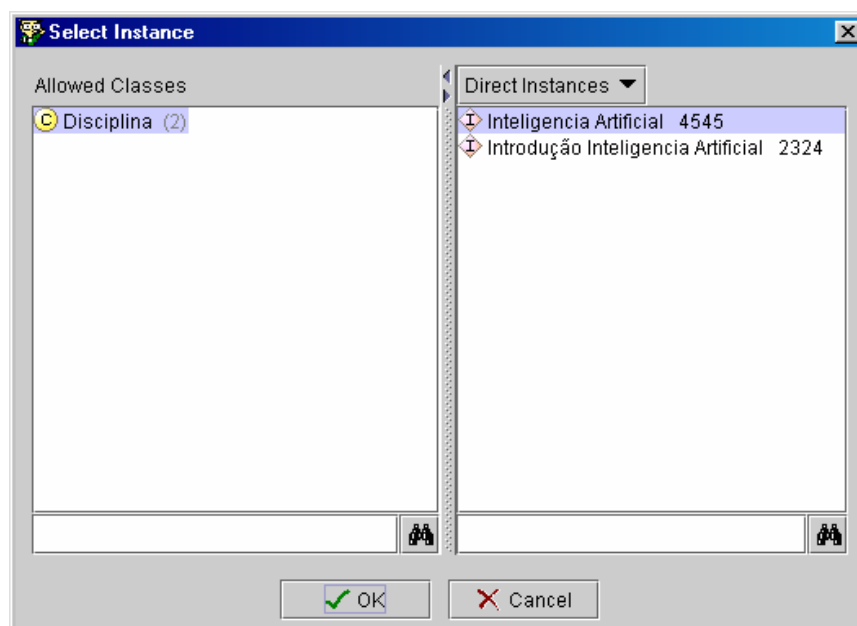
Criando uma instancia de texto



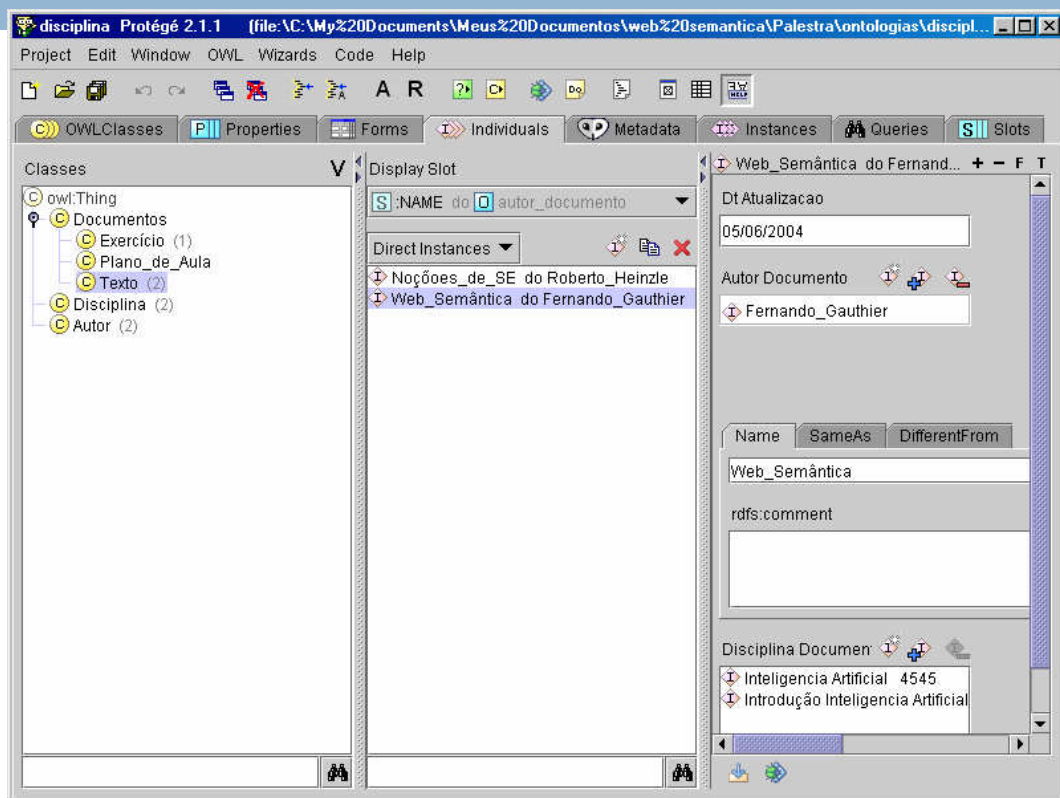
Selecionando o autor do texto



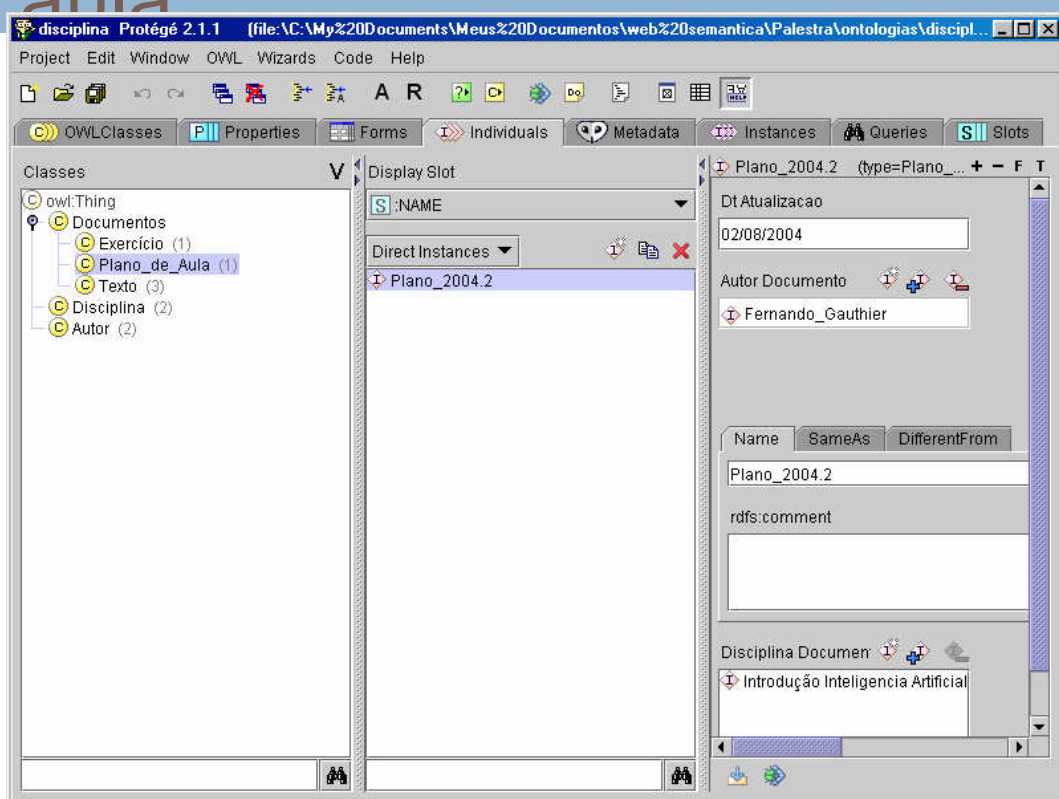
Selecionando a disciplina do texto



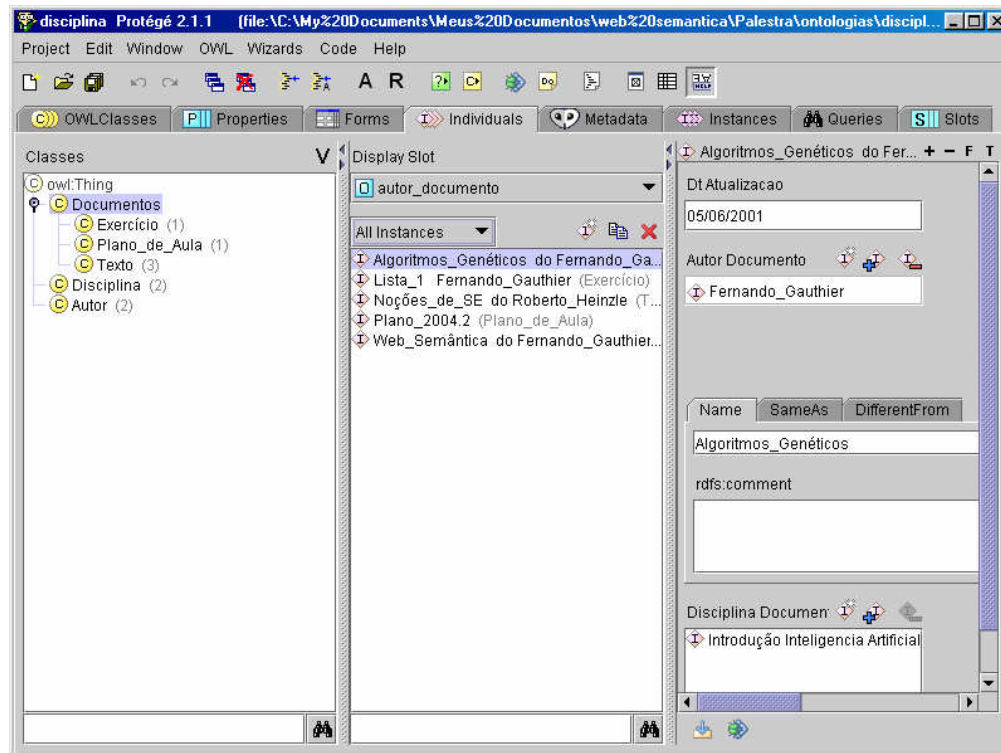
Criando outro texto



Criando uma instância de plano de aula



Exibindo todas as instâncias de documentos

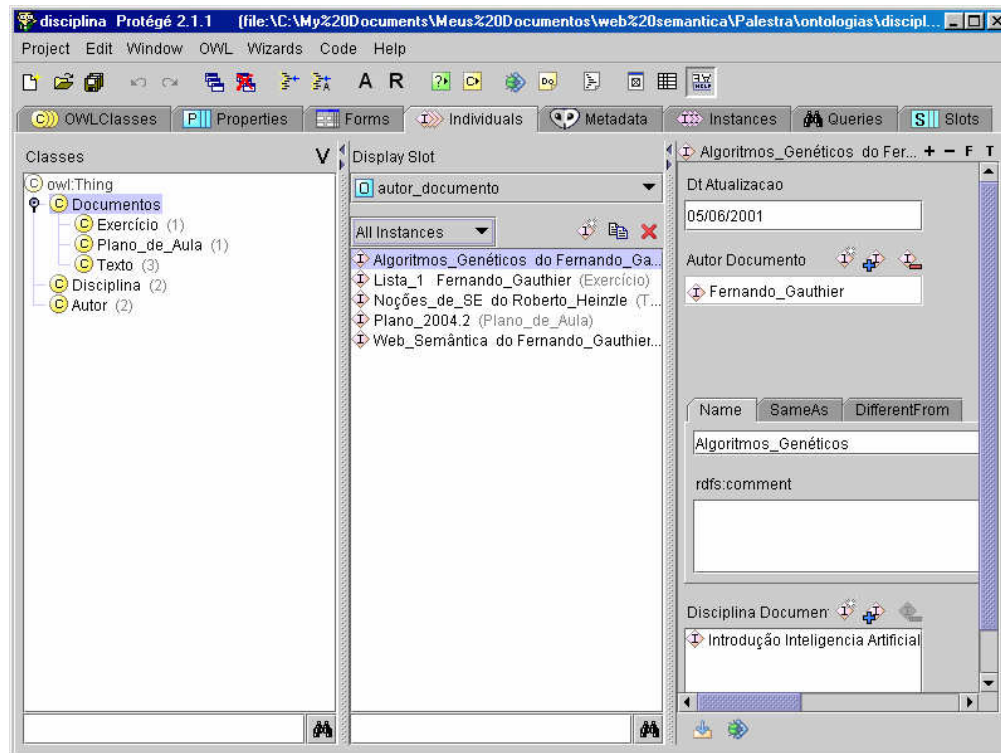


Pesquisando na ontologia com a ferramenta Queries.

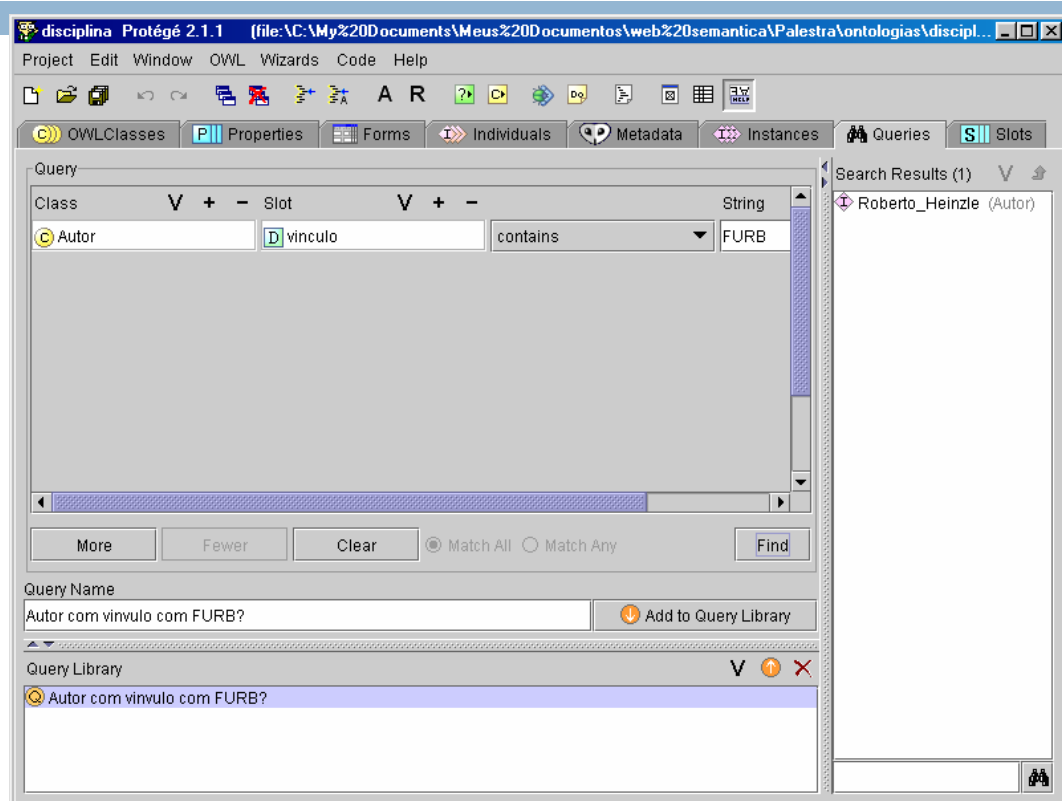
- ❑ *Quais os autores com vínculo com a Unisinos?*
- ❑ *Quais os documentos da disciplina Introdução à Inteligência Artificial?*
- ❑ *Quais as disciplinas que tem documentos com autor com vínculo com a Unisinos?*



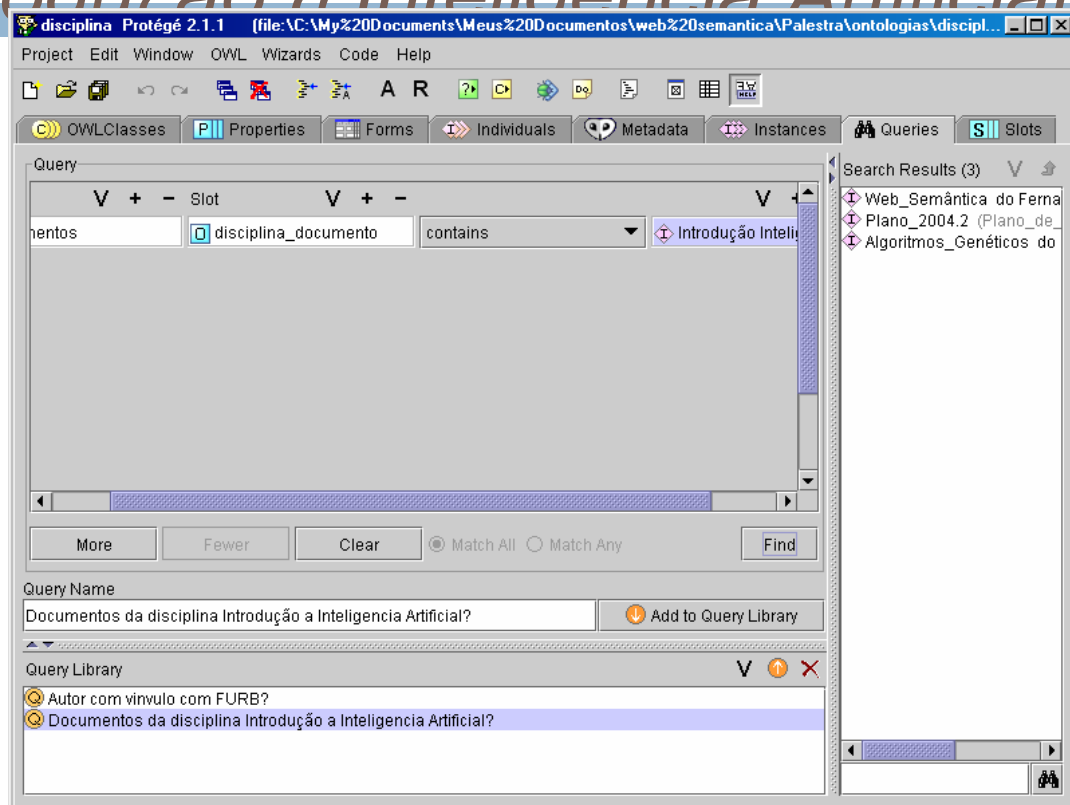
Exibindo todas as instâncias de documentos



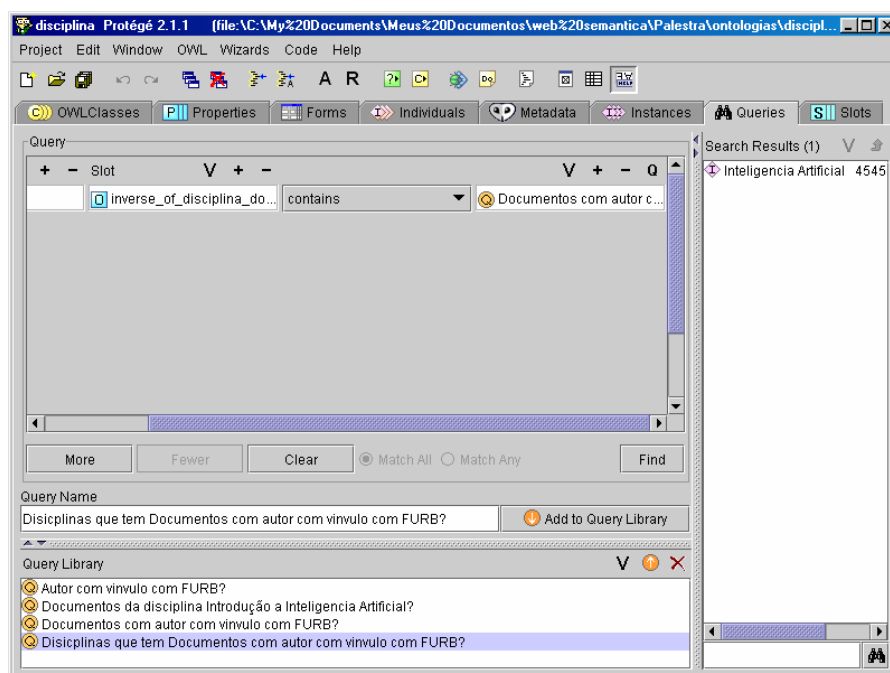
Quais os autores com vínculo com a Unisinos?



Quais os documentos da disciplina *Introdução à Inteligência Artificial?*



Quais as disciplinas que tem documentos com autor com vinculo com a Unisinos?



Outras ferramentas para trabalhar com ontologias

- OilEd
- OntoEdit
- KAON
- OntoBuilder

Bibliografia

- Apresentação de:
 - www.inf.ufsc.br/~gauthier/EGC6006/material/Aula%203/02%20Metodologia101.ppt
- Apresentação de Luiz Palazzo:
 - <http://www.inf.pucrs.br/~ai190471/Mestrado/slides/PCO3.ppt>
- Apresentação de:
 - <http://www.das.ufsc.br/~gb/pg-ia/kr-onto.ppt>
- Ontology Development 101: A Guide to Creating Your First Ontology, por Natalya F. Noy e Deborah L. McGuinness
 - http://protege.stanford.edu/publications/ontology_development/ontology101.html
- Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção, por Almeida, Mauricio B. e Bax, Marcello P. Ciência da informação, 2003, 32 (3)4056
 - http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652003000300002&lng=es&nrm=iso&tling=pt
- GRUBER, T. (1996). What is an Ontology?
 - <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>.
- Getting Started with Protege-Frames. Eliza Sachs.
 - http://protege.stanford.edu/doc/tutorial/get_started/table_of_content.html