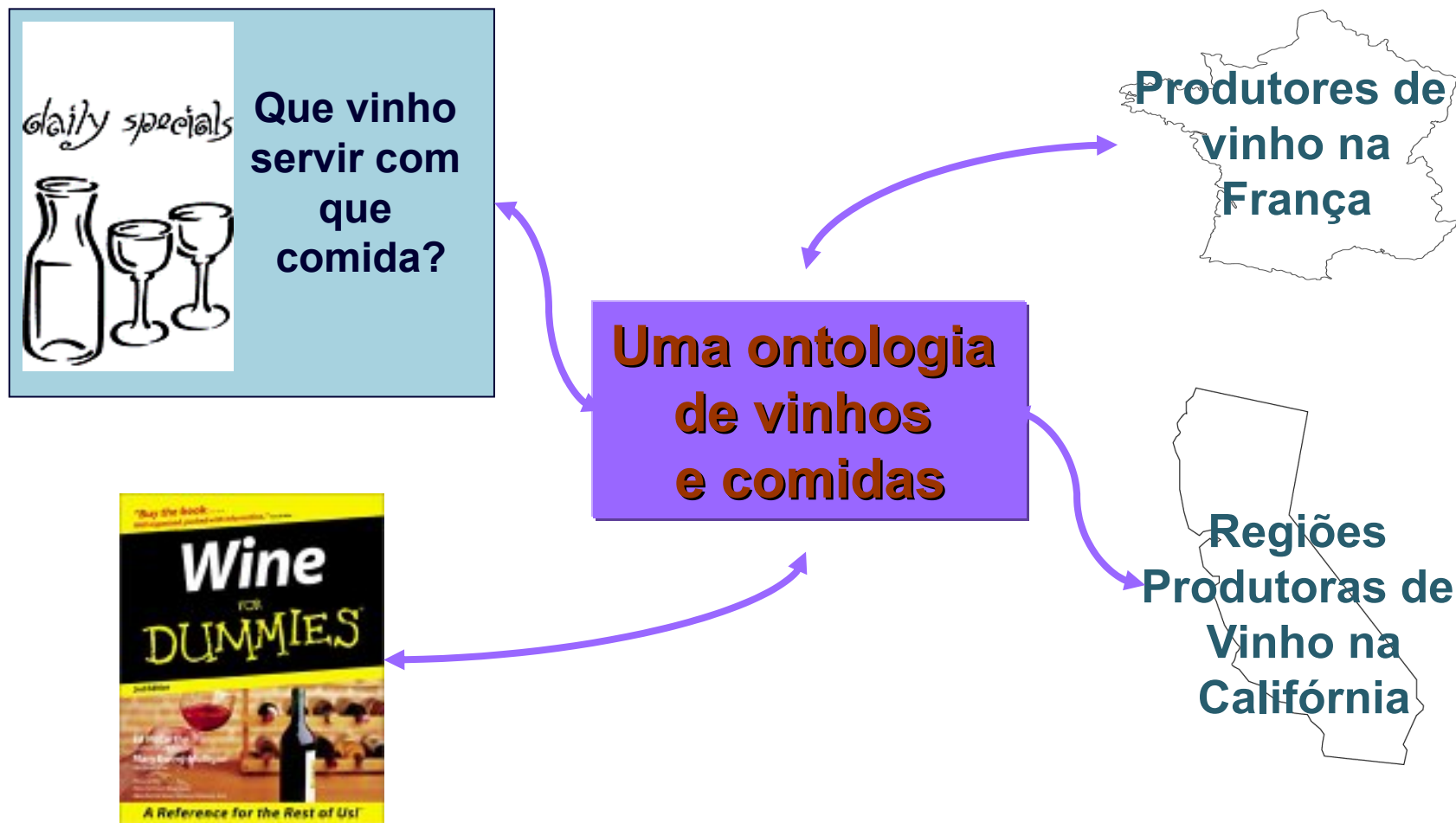


Introdução à construção de ontologias

Fernando Silva Parreiras
www.fernando.parreiras.nom.br



Sumário

- ❑ O que é uma ontologia?
- ❑ Por que desenvolver uma ontologia?
- ❑ Criando uma ontologia passo a passo
- ❑ Problemas e soluções comuns
- ❑ Ontologias na web semântica
- ❑ Pesquisas na criação de ontologias

O que é uma ontologia

- Uma ontologia é uma descrição explícita de um domínio:
 - Conceitos;
 - Propriedades e atributos do conceito;
 - Restrições à propriedade e atributos;
 - Instâncias.
- Uma ontologia define
 - Um vocabulário comum;
 - Um entendimento compartilhado.

Exemplos de ontologias

- Taxonomias na Internet
 - Categorias no Yahoo!
- Catálogos de compras on-line
 - Amazon.com
- Padrões terminológicos de domínios específicos
 - Unified Medical Language System (UMLS)
 - UNSPSC - terminology for products and services

O que é engenharia de ontologias?

Definição de termos e um domínio e os relacionamentos entre eles

- Definição de conceitos em um domínio (classes)
- Organizar os conceitos em hierarquias (subclasses-superclasses)
- Definindo quais atributos e propriedades (slots) as classes devem ter, e as restrições de valores
- Definição de instâncias e preenchimento dos slots

Sumário

- ❑ O que é uma ontologia?
- ❑ Por que desenvolver uma ontologia?
- ❑ Criando uma ontologia passo a passo
- ❑ Problemas e soluções comuns
- ❑ Ontologias na web semântica
- ❑ Pesquisas na criação de ontologias

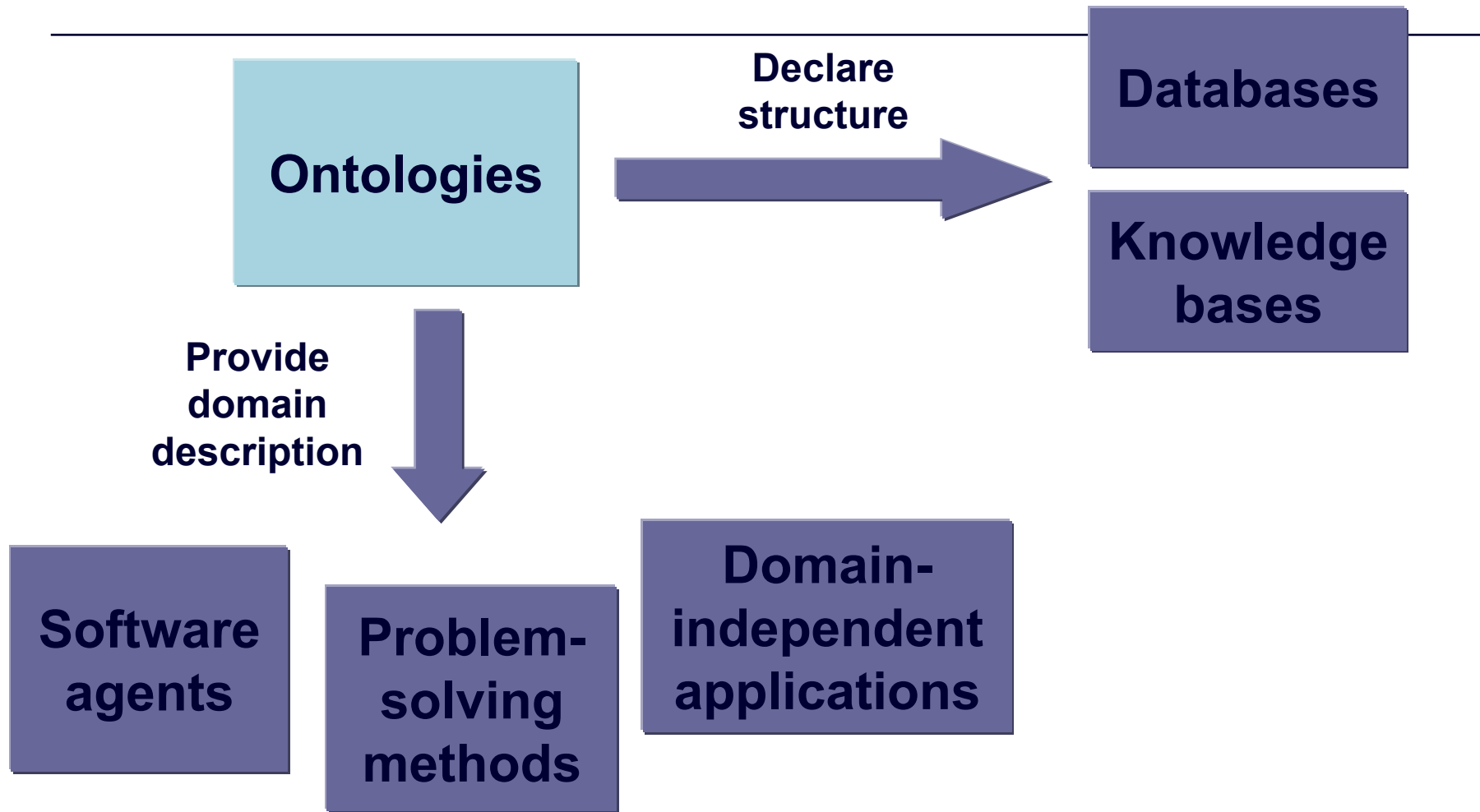
Porque desenvolver uma ontologia?

- Para compartilhar o entendimento de uma estrutura de informação
 - Entre pessoas
 - Entre agentes de software
- Para permitir o reuso de um domínio de conhecimento
 - Para evitar “re-inventar a roda”
 - Para criar padrões e permitir a interoperabilidade

Mais razões

- Para separar conhecimento de domínio de conhecimento operacional
 - Reuso de conhecimento de domínio e operacional separadamente.

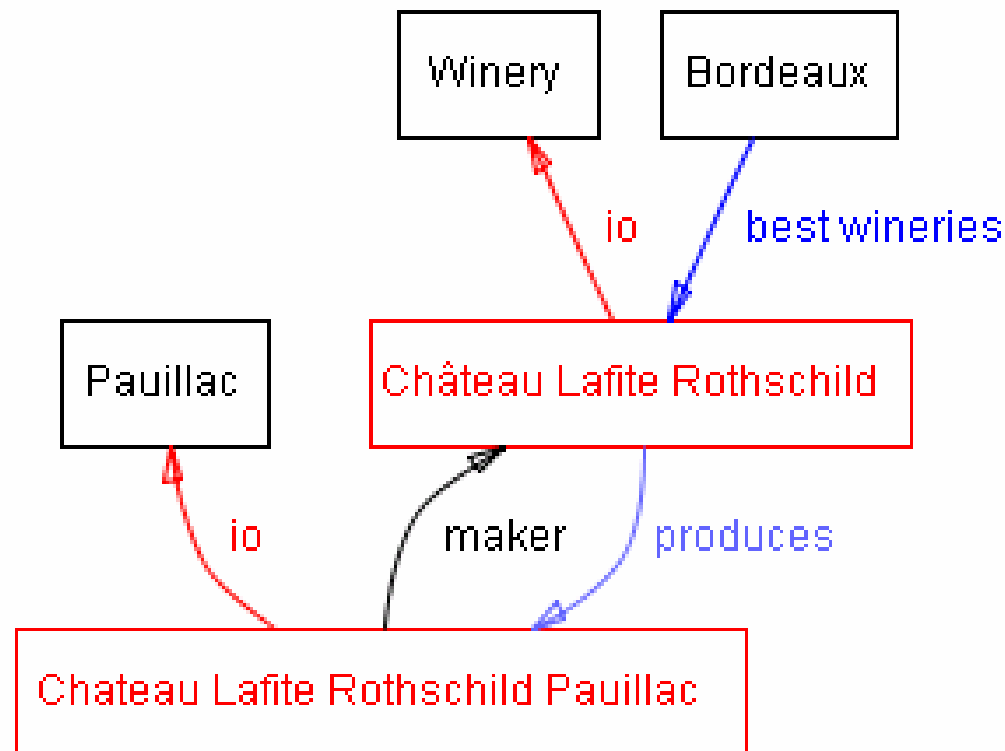
Uma ontologia é só o começo



Sumário

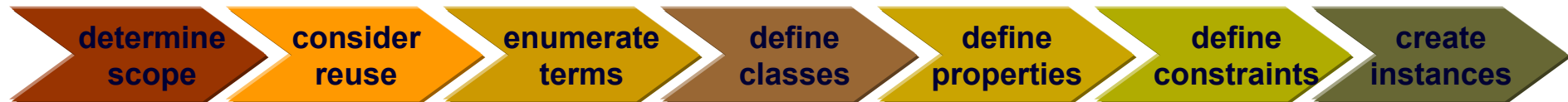
- ❑ O que é uma ontologia?
- ❑ Por que desenvolver uma ontologia?
- ❑ Criando uma ontologia passo a passo
- ❑ Problemas e soluções comuns
- ❑ Ontologias na web semântica
- ❑ Pesquisas na criação de ontologias

Wines and Wineries

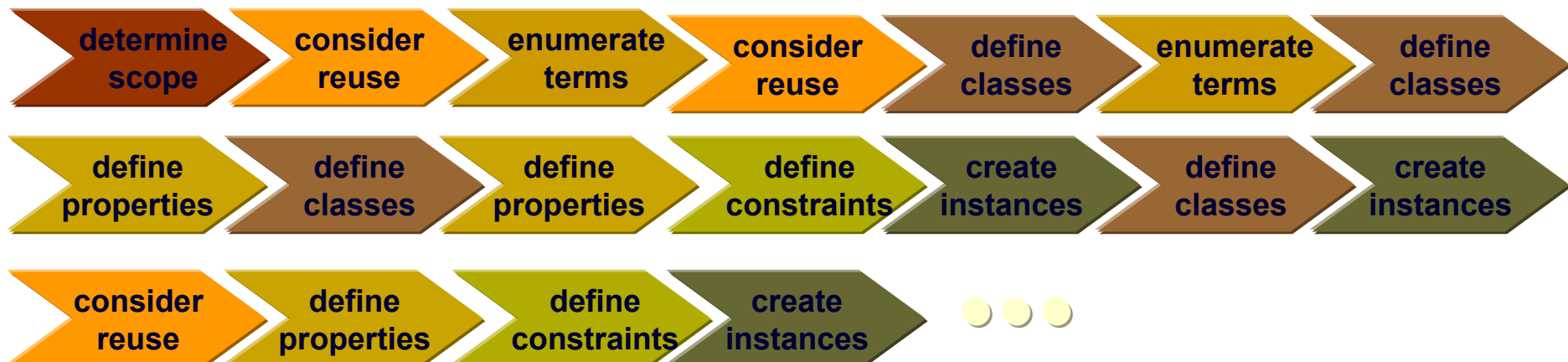


Processo de desenvolvimento

Neste mini-curso:



Na realidade:



Engenharia de ontologias versus modelagem OO

Uma ontologia

- ❑ Reflete a estrutura do mundo
- ❑ É geralmente sobre estruturas de conceitos
- ❑ Não se preocupa com a representação física

Uma estrutura OO

- ❑ Reflete a estrutura de dados e código
- ❑ Trata comportamento (métodos)
- ❑ Descreve a representação física dos dados

Ferramentas

- Protégé-2000:
 - Editor gráfico de ontologias
 - Suporta um modelo rico de conhecimento
 - É código livre
(<http://protege.stanford.edu>)
- Outras ferramentas
 - Ontolingua and Chimaera
 - OntoEdit
 - OilEd

Determinando domínio e escopo



- Que domínio a ontologia vai cobrir?
- Para que nós vamos utilizar a ontologia?
- Que tipo de questões as informações na ontologia vão responder?

As respostas à estas questões podem mudar ao longo do tempo

Questões de competência

- ❑ Quais características do vinho eu devo considerar na escolha
- ❑ Bordeaux é um vinho tinto ou branco
- ❑ Cabernet Sauvignon combina com pescado?
- ❑ Qual é a melhor escolha de vinho para carnes grelhadas?
- ❑ Quais características de um vinho afetam sua combinação com um prato?
- ❑ O sabor ou cor de um vinho específico muda de acordo com a safra?

Considerando a reutilização



- Por que reutilizar ontologias?
 - Para poupar esforço
 - Para interagir com as ferramentas que usam outras ontologias
 - Para utilizar ontologias validadas através do uso em aplicações

Onde buscar ontologias existentes?

- Bibliotecas de ontologias
 - DAML ontology library (www.daml.org/ontologies)
 - Ontolingua ontology library (www.ksl.stanford.edu/software/ontolingua/)
 - Protégé ontology library (protege.stanford.edu/plugins.html)
- Upper ontologies
 - IEEE Standard Upper Ontology (suo.ieee.org)
 - Cyc (www.cyc.com)

Onde buscar ontologias existentes? (II)

- Ontologias gerais
 - DMOZ (www.dmoz.org)
 - Yahoo! (www.yahoo.com.br)
 - WordNet (www.cogsci.princeton.edu/~wn/)
- Específicas de um domínio
 - UMLS Semantic Net
 - GO (Gene Ontology) (www.geneontology.org)

Enumerando Termos Importantes

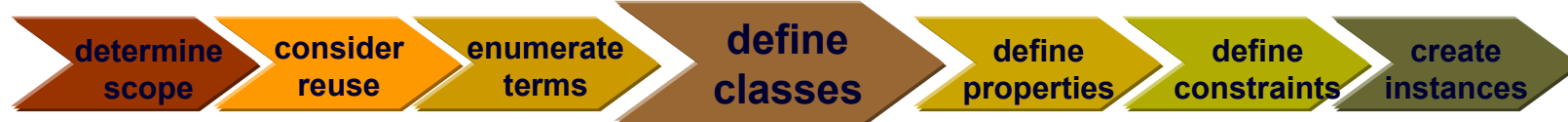


- ❑ Quais são os termos que nós estamos tratando?
- ❑ Quais são suas propriedades?
- ❑ O que nós queremos dizer sobre estes termos

The Wine Ontology

*wine, grape, winery, location,
wine color, wine body, wine flavor, sugar content
white wine, red wine, Bordeaux wine
food, seafood, fish, meat, vegetables, cheese*

Definindo as Classes e a Hierarquia das Classes



- Uma classe é um conceito em um domínio
 - Uma classe de wines
 - Uma classe de wineries
 - Uma classe de red wines
- Uma classe é uma coleção de elementos com propriedades similares
- Instâncias de classes
 - Uma garrafa de vinho qualquer

Herança de classes

- Classes usualmente constituem uma hierarquia taxonômica (uma hierarquia subclasse-superclasse)
- Uma hierarquia de classe é usualmente uma hierarquia "IS-A":

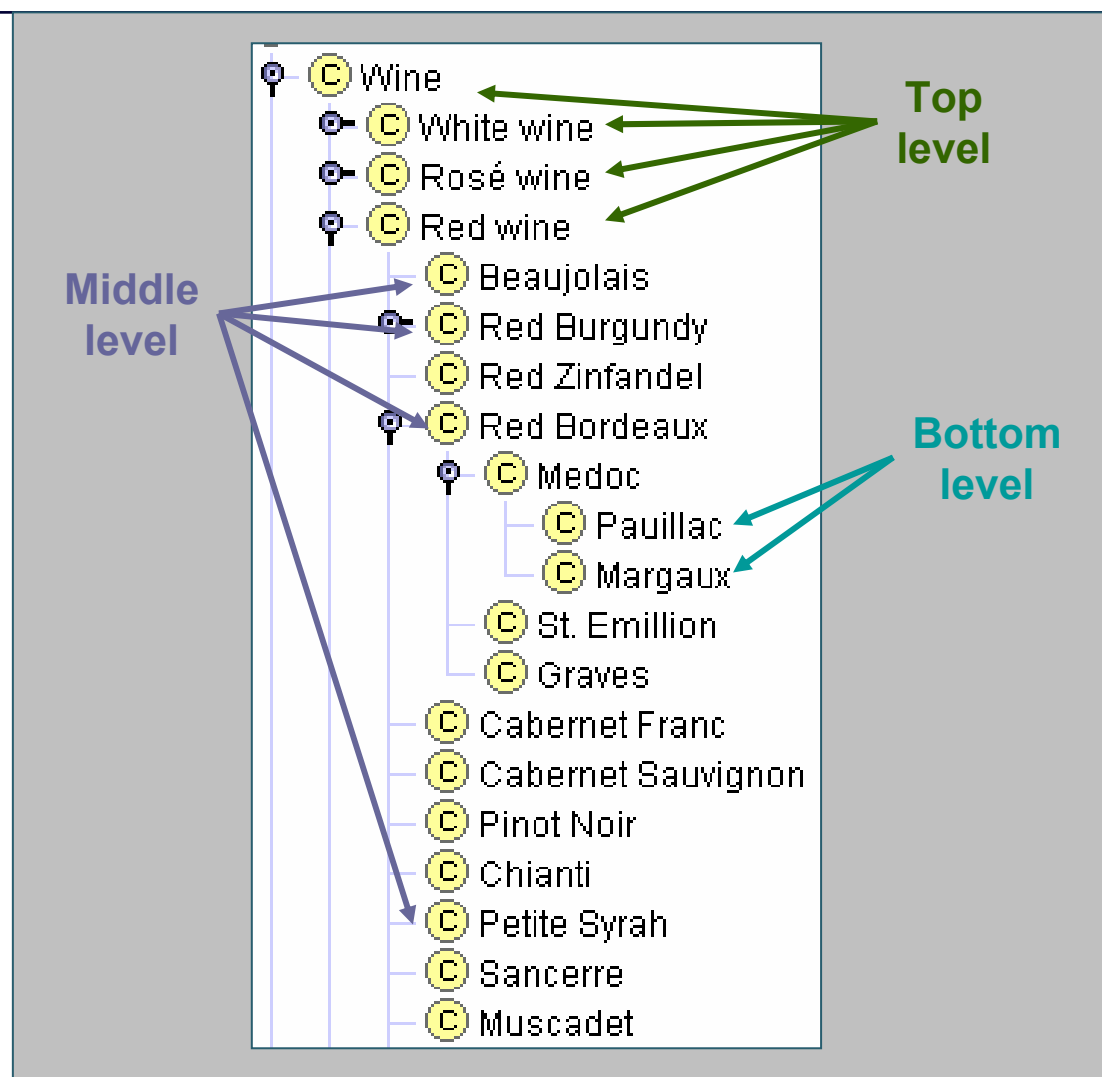
Uma instância de uma subclasse é uma instância de uma superclasse

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

Exemplos

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

Níveis de hierarquia



Modos de desenvolvimento

- ❑ top-down - Define os conceitos genéricos e depois os detalha
- ❑ bottom-up - define os conceitos específicos e então os organiza em classes mais genéricas
- ❑ combination - define os conceitos mais salientes primeiro e depois os generaliza e especializa

Documentação

- Classes (e slots) usualmente tem documentação
 - Descrevendo a classe em linguagem natural
 - Listando premissas do domínio relevantes para a definição da classe
 - Listando sinônimos
- Documentar classes e slots é tão importante quanto documentar código fonte!

Definindo Propriedades das classes - Slots

















- Slots em uma definição de classe, descreve atributos de instancias da classe e as relações com outras instâncias

Cada vinho tem cor, produtor, acidez, etc.

Propriedades (Slots)

- Tipos de propriedades:
 - propriedades: name e price do vinho
 - partes: ingredientes em um prato
 - Relações com outros objetos: produtora de vinho (winery)
- Propriedades simples e complexas
 - Propriedades simples (atributos): contêm valores atômicos (número, caracteres)
 - Propriedades complexas: contém outros objetos

Slots da classe vinho

Template Slots				     
Name	Type	Cardinality	Other Facets	
 body	Symbol	single	allowed-values={FULL,MEDIUM,LIGHT}	
 color	Symbol	single	allowed-values={RED,ROSÉ,WHITE}	
 flavor	Symbol	single	allowed-values={DELICATE,MODERATE,STRONG}	
 grape	Instance	multiple	classes={Wine grape}	
 maker 	Instance	single	classes={Winery}	
 name	String	single		
 sugar	Symbol	single	allowed-values={DRY,SWEET,OFF-DRY}	

(in Protégé-2000)

Slot e herança de classes

- Uma subclasse herda todos os slots da superclasse

Se um vinho tem um nome e sabor, um vinho tinto também tem nome e sabor

- Se uma classe tem múltiplas superclasses, ela herda slots de todas

Restrições das propriedades

















- Restrições das propriedades (facetas) descrevem ou limitam o conjunto de valores possíveis para um slot

The name of a wine is a string

The wine producer is an instance of Winery

A winery has exactly one location

Facetas dos Slots da Classe vinhos

Template Slots				     
Name	Type	Cardinality	Other Facets	
 body	Symbol	single	allowed-values={FULL,MEDIUM,LIGHT}	
 color	Symbol	single	allowed-values={RED,ROSÉ,WHITE}	
 flavor	Symbol	single	allowed-values={DELICATE,MODERATE,STRONG}	
 grape	Instance	multiple	classes={Wine grape}	
 maker 	Instance	single	classes={Winery}	
 name	String	single		
 sugar	Symbol	single	allowed-values={DRY,SWEET,OFF-DRY}	

Facetas comuns

- ☐ Cardinalidade
- ☐ Tipo de dado
- ☐ Valor mínimo e máximo
- ☐ Valor padrão

Facetas comuns: Cardinalidade

- Cardinalidade
 - N se o slot deve ter N valores
- Cardinalidade mínima
 - 1 se o slot deve ter pelo menos um valor (requerido)
 - 0 se o slot é opcional
- Cardinalidade máxima
 - 1 se no máximo (single-valued slot)
 - >1 se o slot é multivalorado (multiple-valued slot)

Facetas comuns: Tipos de dados

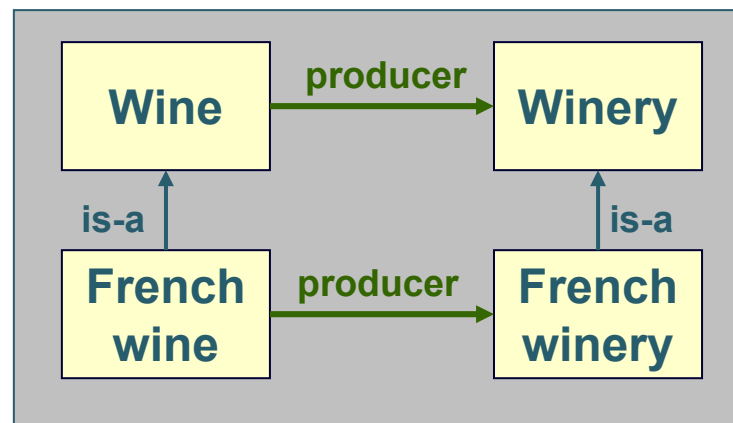
- ❑ String: uma cadeia de caracteres
 - ❑ Number: um inteiro ou ponto flutuante
 - ❑ Boolean: true/false
 - ❑ Enumerated type: lista de valores
 - ❑ Complex type: uma instância de outra classe
- The Wine class is the value type for the slot “produces” at the Winery class*

Domínio e intervalo de Slot

- Domain - a classe (ou classes) que tem o slot
 - Mais precisamente: instância de classe (ou classes) que podem ter o slot
- Range - a classe (ou classes) a qual os valores do slot pertencem

Facetas e Herança de Classe

- uma subclasse herda todos os slots de uma superclasse
- Uma subclasse pode sobrescrever as facetas para restringir a lista de valores permitidos
 - Diminuindo a cardinalidade



Criando instâncias



- Ao criar a instância de uma classe
 - A classe se torna um tipo direto da instância
 - Qualquer superclasse do tipo direto é um tipo da instância
- Atribuindo valores
 - Os valores devem ser conforme com as facetas
 - Ferramentas de aquisição de conhecimento verificam os valores

Exemplo

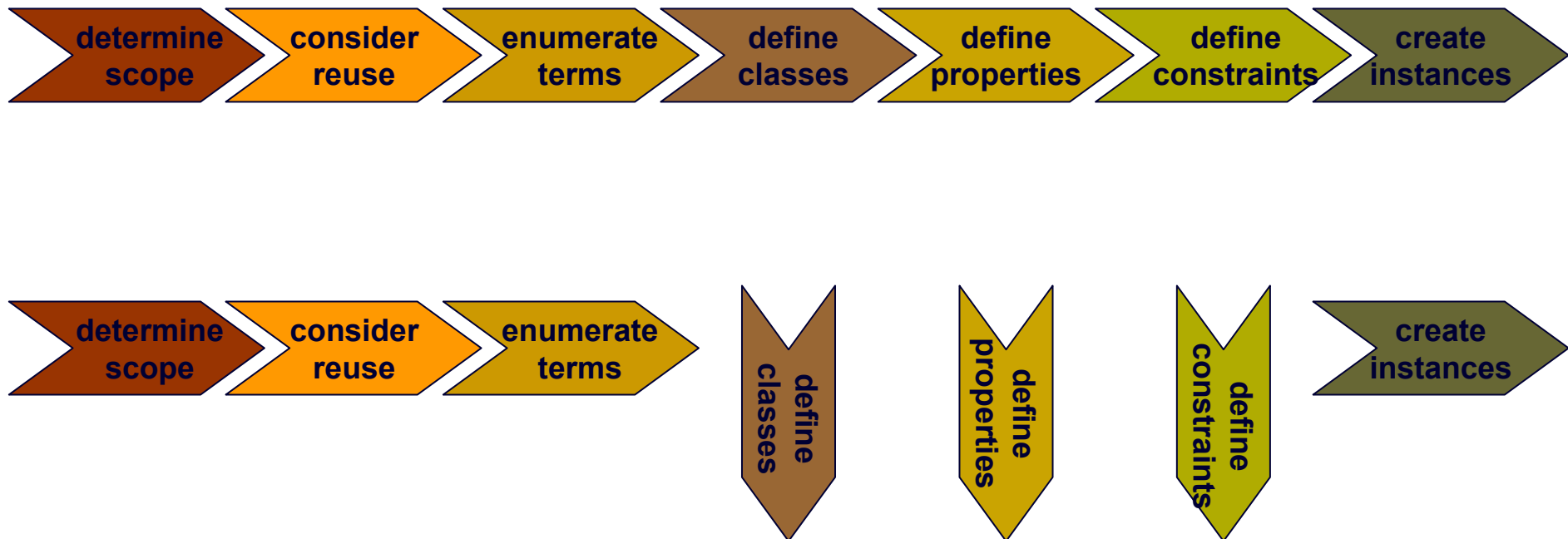
The screenshot shows a software window titled "Chateau Morgon Beaujolais (Beaujolais)". The window contains several input fields and dropdown menus for wine data entry:

- Name:** A text box containing "Chateau Morgon Beaujolais".
- Area:** A text box containing "Beaujolais region" with a yellow circle icon to its left. To the right are buttons "V", "+", and "-" stacked vertically.
- Body:** A dropdown menu showing "LIGHT".
- Color:** A dropdown menu showing "RED".
- Maker:** A text box containing "Chateau Morgon" with a diamond icon to its left. To the right are buttons "V", "C", "+", and "-" stacked vertically.
- Flavor:** A dropdown menu showing "DELICATE".
- Sugar:** A dropdown menu showing "DRY".
- Grape:** A text box containing "Gamay grape" with a diamond icon to its left. To the right are buttons "V", "C", "+", and "-" stacked vertically.
- Tannin Level:** A dropdown menu showing "LOW".

Sumário

- ❑ O que é uma ontologia?
- ❑ Por que desenvolver uma ontologia?
- ❑ Criando uma ontologia passo a passo
- ❑ Problemas e soluções comuns
- ❑ Ontologias na web semântica
- ❑ Pesquisas na criação de ontologias

Aprofundando



Definindo classes e sua hierarquia

- Lembre-se:
 - Não existe uma hierarquia correta
 - Mas sim melhores práticas
- Pergunte-se:
 - “Cada instâncias de uma subclasse uma instância de uma superclasse?”

Transitividade da hierarquia de classes

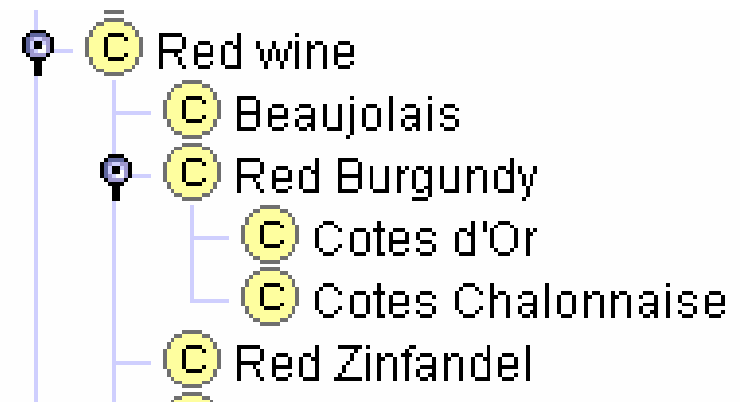
- O relacionamento “is-a” transitivo:

B é uma subclasse de A

C é uma subclasse de B

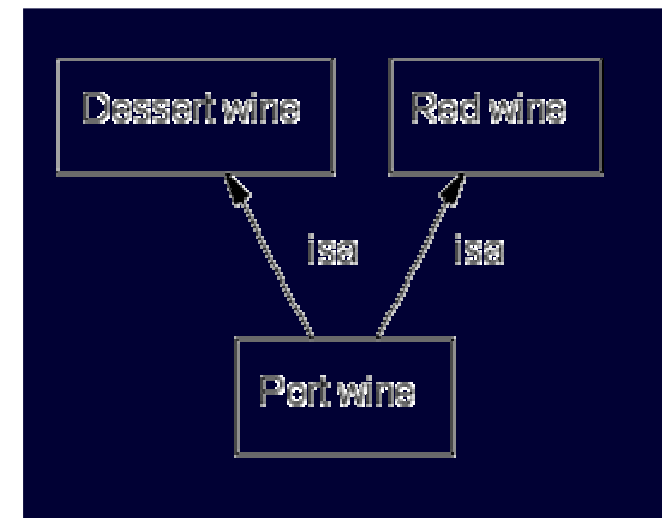
~~C é uma subclasse de A~~

- Uma superclasse direta de uma classe é sua superclasse mais perto



Herança múltipla

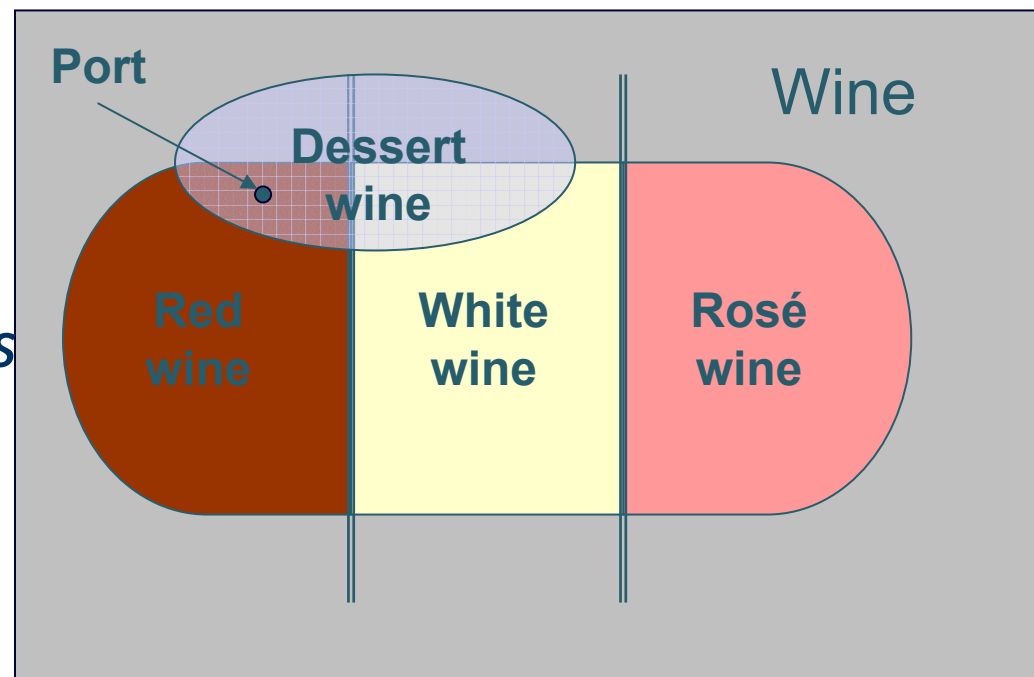
- ❑ Uma classe pode ter mais de uma superclasse
- ❑ Uma subclasse herda slots e facetas de todos os pais
- ❑ Sistemas diferentes resolvem conflitos de maneira diferente



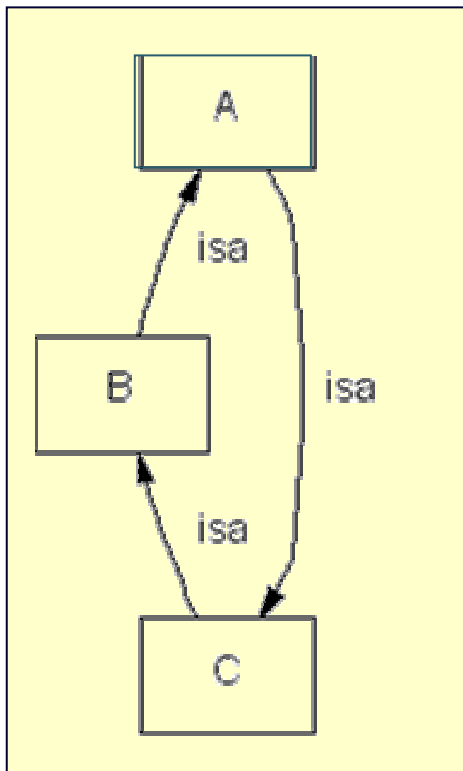
Classes Disjuntas

- Classes são disjuntas se elas não podem ter instâncias em comum
- classes disjuntas não podem ter nenhuma subclasse em comum

*Red wine, White wine,
Rosé wine são disjuntas*
*Dessert wine and Red
wine não são disjuntas*

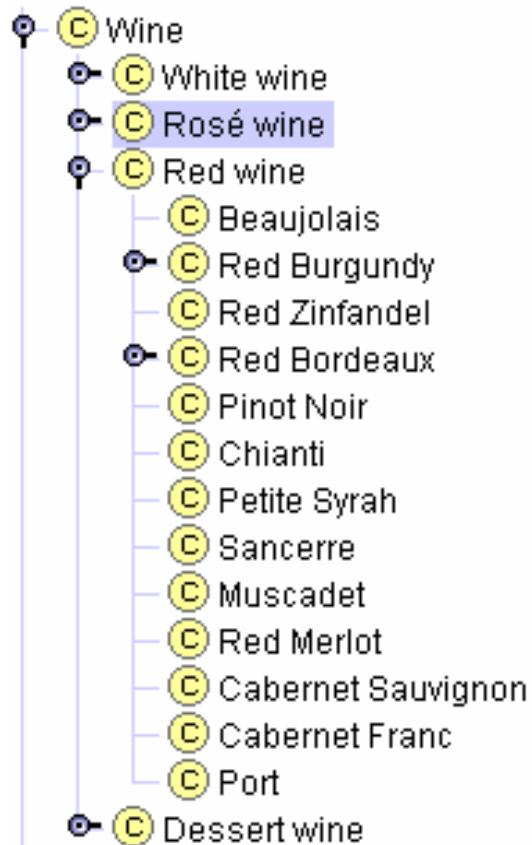


Evitando Ciclos de Classes



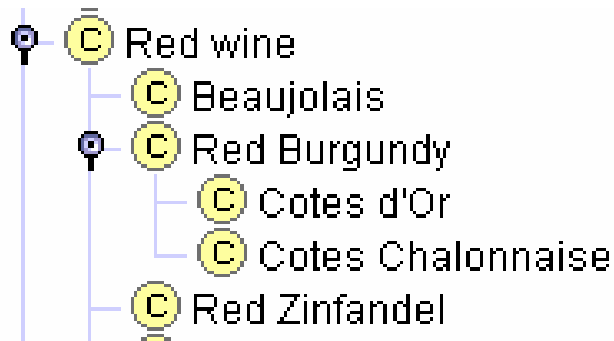
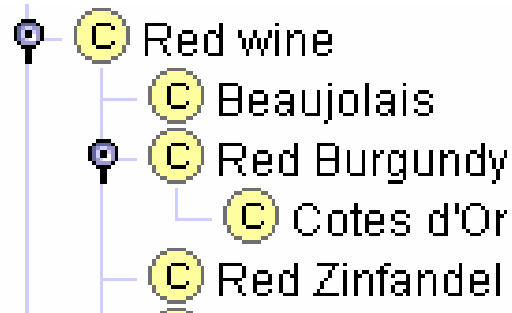
- ❑ Perigo de herança múltipla: hierarquia cíclica
- ❑ Classes A, B, e C tem conjuntos equivalentes de instância
 - A, B, e C são equivalentes

Irmãos na hierarquia



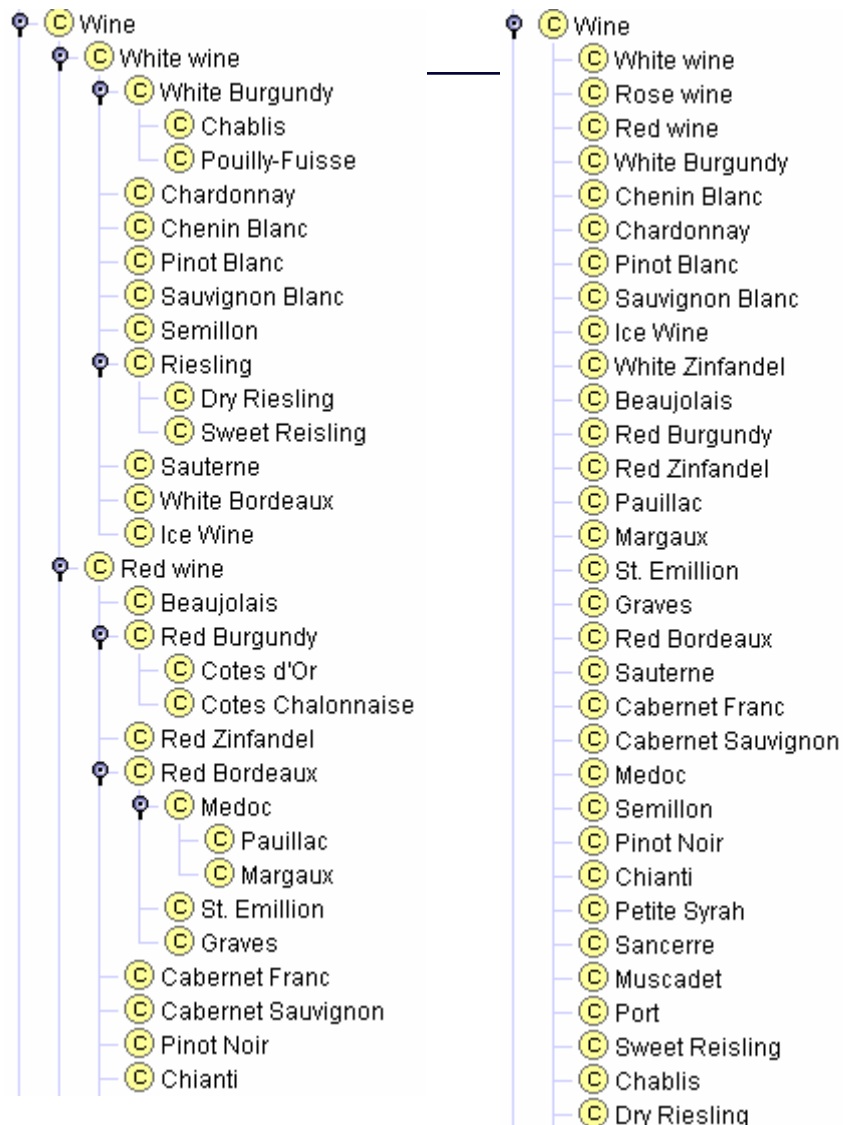
- ❑ Todos irmãos devem estar no mesmo nível de generalidade
- ❑ Compare com as seções e subseções de um livro

O tamanho perfeito



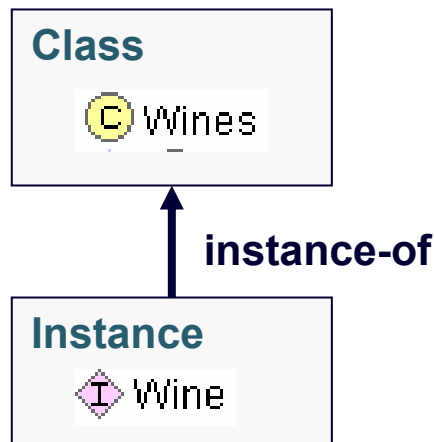
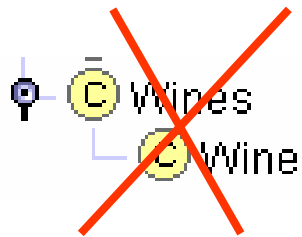
- Se uma classe tem apenas um filho, há um problema de modelagem
- Se o único Red Burgundy que tem-se é Côtes d'Or, porque criar uma hierarquia?
- Compare com uma lista marcada

O tamanho perfeito(II)



- Se uma classe tem dezenas de filhos, subcategorias podem ser necessárias
- Entretanto, se não existe uma classificação natural, isso pode ser natural

Nomes no plural e no singular

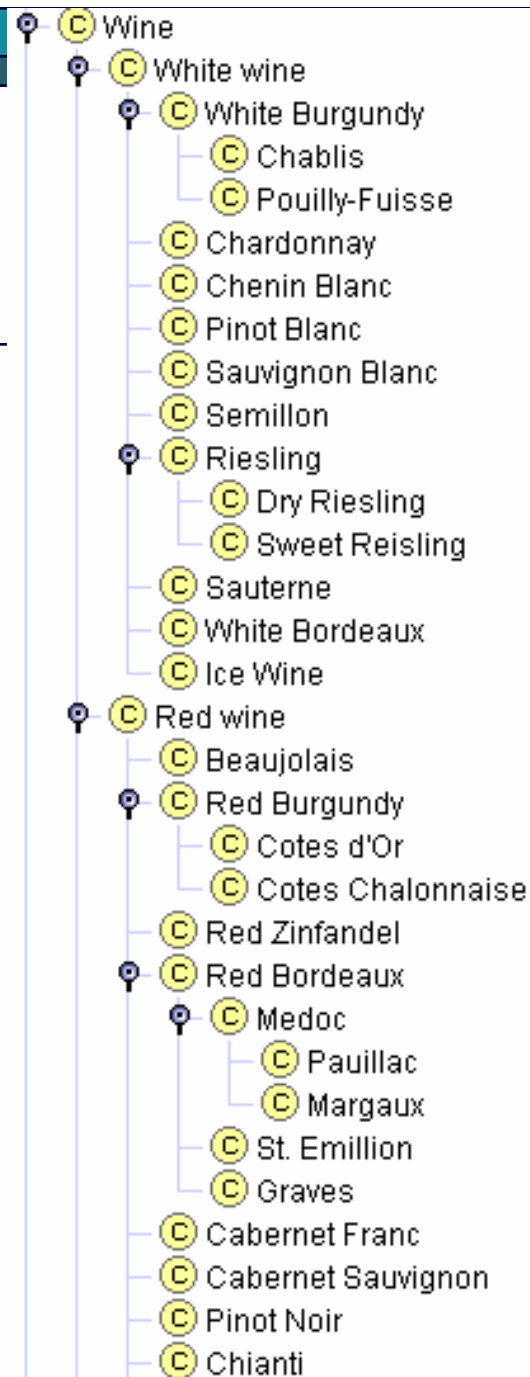


- Um “wine” não é um tipo de “wines”
- Um wine é uma instância da classe Wines
- Os nomes devem ser
 - Todos no singular ou
 - Todos no plural

Nomes de classes

- Classes representam conceitos no domínio, não os nomes
- O nome da classe pode mudar, mas ele vai continuar a se referir ao mesmo conceito
- Nomes sinônimos para o mesmo conceito não são classes diferentes
 - Muitos sistemas permitem listar sinônimos como parte da definição das classes

A hierarquia completa de vinhos



Slots: domínio e intervalo

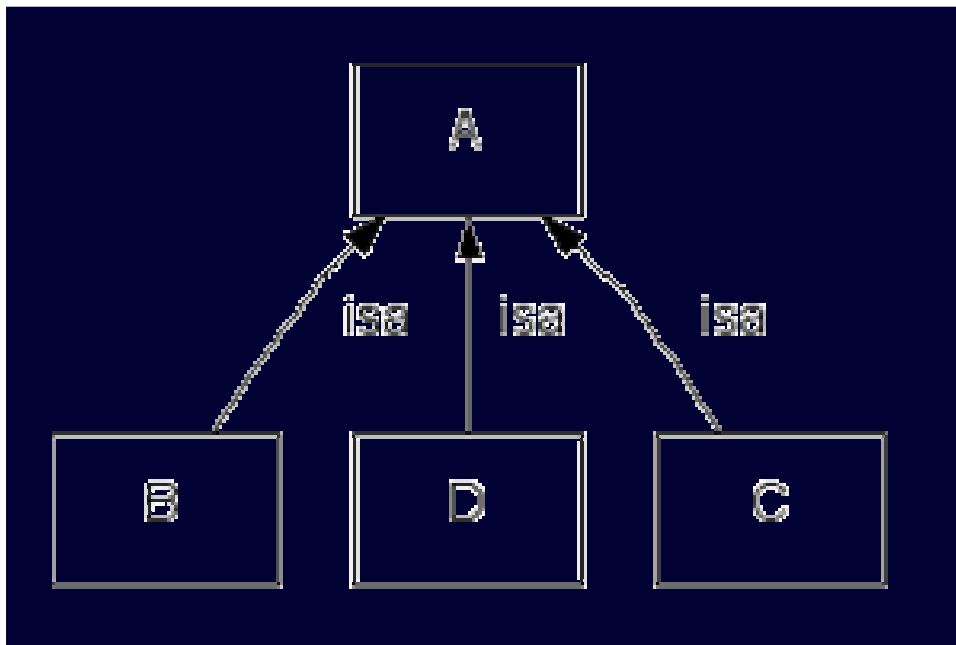
- Considerando o slot flavor
 - Domain: Red wine, White wine, Rosé wine
 - Domain: Wine
- Considerando o slot produces para uma Winery:
 - Range: Red wine, White wine, Rosé wine
 - Range: Wine

Domínio e intervalo



- Considerando o slot **flavor**
 - Domain: Red wine, White wine, Rosé wine
 - Domain: Wine
- Considerando o slot **produces** para uma **Winery**:
 - Range: Red wine, White wine, Rosé wine
 - Range: Wine

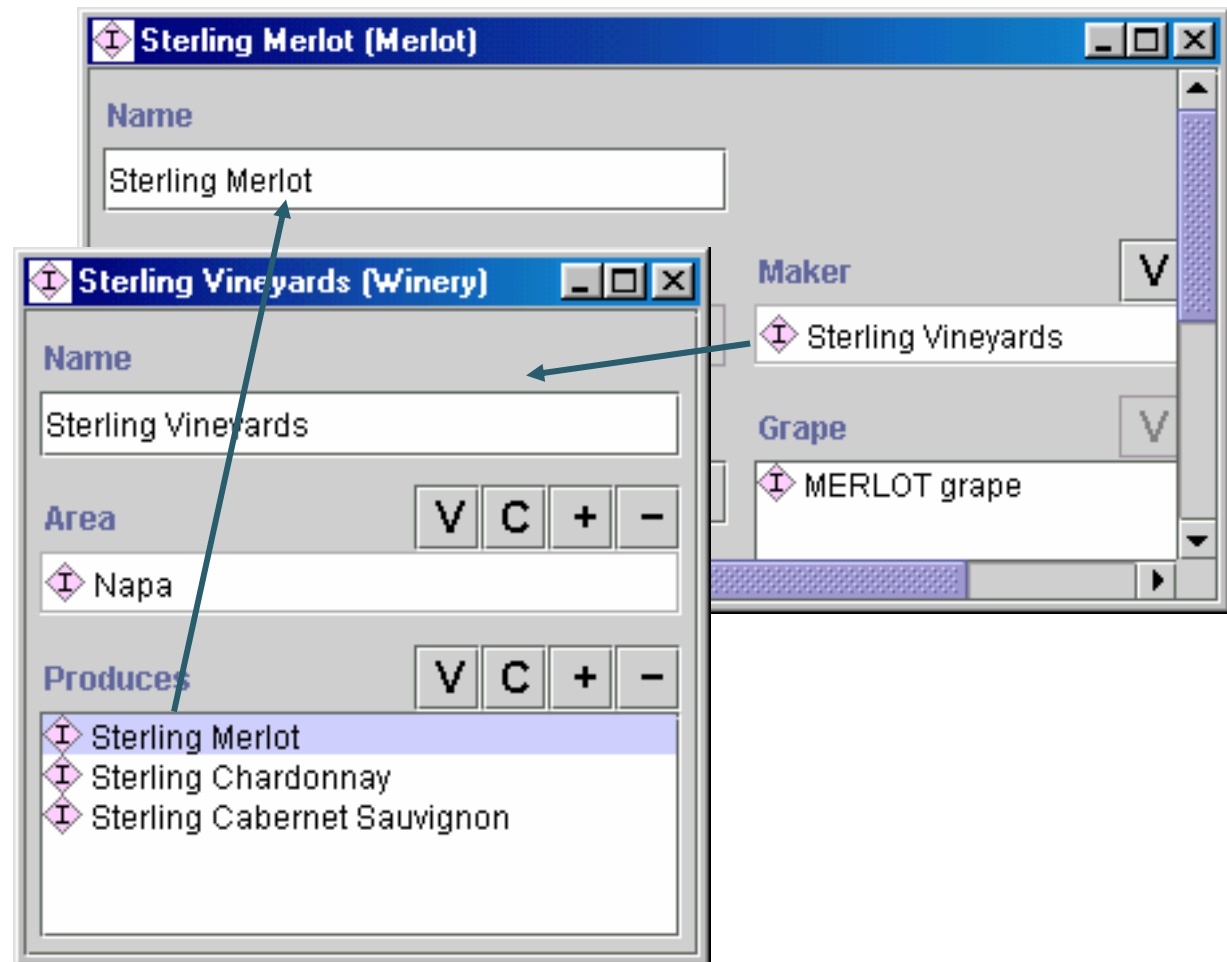
Definindo domínio e intervalo



- Uma classe e uma subclasse - substitua por uma superclasse
- Muitas subclasses em uma classe - considere substituir por superclasses

Slots inversos

Maker e
Producer
are são slots inversos



Slots inversos (II)

- Slots inversos contem informação redundante, mas
 - Permite aquisição de conhecimento nas duas direções
 - Habilita verificação
 - Permite apresentação da informação em duas direções
- A implementação atual se difere de sistema para sistema
 - Os dois valores são armazenados?
 - Quando preencher?
 - O que acontece se mudar um link?

Valores Padrão

- ❑ Valor Padrão- um valor assumido pelo slot na criação da instância
- ❑ Pode ser mudado
- ❑ É um valor comum, não necessariamente requerido
- ❑ O valor padrão para “wine body” pode ser “FULL”

Limitando o escopo

- Uma ontologia não deve conter todas as informações possíveis sobre o domínio
 - Não é necessário especializar ou generalizar mais que a aplicação requer
 - Não é necessário incluir todas propriedades possíveis de uma classe
 - Apenas as mais salientes
 - Apenas as que a aplicação requer

Limitando o escopo (II)

- A ontologia de vinhos provavelmente não incluirá:
 - Bottle size
 - Label color
 - My favorite food and wine
- Uma ontologia de experimentos médicos conterà
 - Organismo Biológico
 - Experimentador
- A classe experimentador é uma subclasse de organismo biológico?

Sumário

- ❑ O que é uma ontologia?
- ❑ Por que desenvolver uma ontologia?
- ❑ Criando uma ontologia passo a passo
- ❑ Problemas e soluções comuns
- ❑ Ontologias na web semântica
- ❑ Pesquisas na criação de ontologias

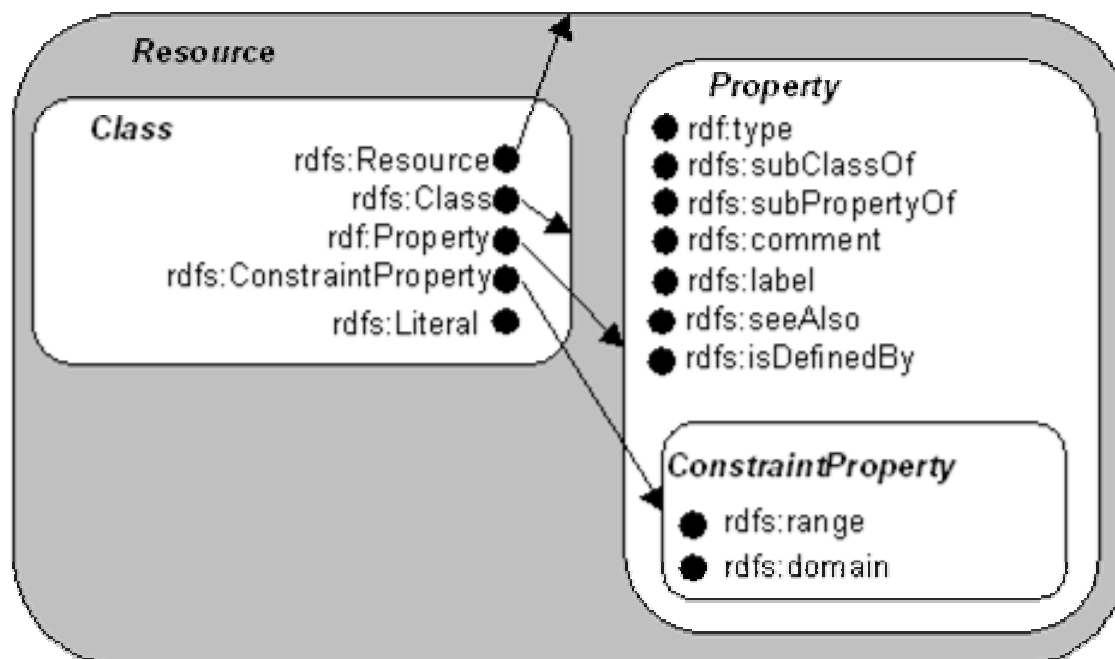
Ontologias e as linguagens da WS

- As linguagens de web semântica podem ser utilizadas para definir ontologias
 - RDF Schema
 - DAML+OIL
 - SHOE
 - XOL
 - XML Schema

Linguagens SW

- As linguagens podem ser diferentes em
 - sintaxe
 - terminologia
 - expressividade
 - semântica

Classes RDF e RDF Schema



RDF Schema Specification 1.0 (<http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>)

RDF(S) Terminologia e Semântica

- Hierarquia de classes
 - Todas classes são instâncias de `rdfs:Class`
 - Uma hierarquia é definida por `rdfs:subClassOf`
- Instâncias de uma classe
 - Definidas por `rdf:type`
- Propriedades
 - Propriedades são global:
Uma propriedade `name` em um lugar é o mesmo que a propriedade `name` em outro lugar (assumindo o mesmo namespace)
 - Propriedades tem hierarquia também (`rdfs:subPropertyOf`)

Restrições em RDF(S)

- ❑ Cardinalidade
 - Não há restrições explícitas
 - Qualquer propriedade pode ter múltiplos valores
- ❑ Intervalo de propriedades
 - Uma propriedade pode ter apenas um intervalo
- ❑ Domínio
 - Uma propriedade pode ter mais de um domínio (Pode ser ligada à mais de uma classe)
- ❑ Não há valores padrão

DAML+OIL: Classes e hierarquia

- Classes
 - Cada classe é uma instância de `daml:Class`
- Hierarquia
 - Definida por `rdfs:subClassOf`
- Mais formas de organizar as classes
 - disjuntura (`daml:disjointWith`)
 - Equivalência (`daml:sameClassAs`)
- A hierarquia de classes pode ser calculada a partir das propriedades das classes

Mais definições de classes em DAML+OIL

- União de classes

A class Person is a union of classes Male and Female

- Restrição em propriedades

A class Red Thing is a collection of things with color: Red

- Intercessão classes

A class Red Wine is an intersection of Wine and Red Thing

- Complemento de uma classe

Carnivores are all the animals that are not herbivores

- Listagem de elementos

A class Wine Color contains the following instances: red, white, rosé

Restrições em DAML+OIL

- Cardinalidade
 - Mínimo, máximo, cardinalidade exata
- Range
 - Um intervalo podem incluir múltiplas classes: o valor de uma propriedade precisa ser uma instância de cada classe
 - Pode especificar a união explícita de classe se necessitar semântica diferente
- Domínio: o mesmo
- Sem valores padrão

Sumário

- ❑ O que é uma ontologia?
- ❑ Por que desenvolver uma ontologia?
- ❑ Criando uma ontologia passo a passo
- ❑ Problemas e soluções comuns
- ❑ Ontologias na web semântica
- ❑ Pesquisas na criação de ontologias

Pesquisas na engenharia de ontologias

- ❑ Geração de conteúdo
- ❑ Análise e avaliação
- ❑ Manutenção
- ❑ Linguagens
- ❑ Ferramentas

Conteúdo

- O que significa “top-level”?
 - Objetos: tangíveis, intangíveis
 - Processos, eventos, atores, papéis
 - Agentes, organizações
 - Espaço, limites, localização
 - Tempo
- IEEE Standard Upper Ontology effort
 - Meta: Desenha uma ontologia de alto nível
 - Processo: utilizar ontologias existentes

Aquisição de conhecimento

- ❑ Aquisição de conhecimento é o gargalo
- ❑ Compartilhar e reusar alivia o problema
- ❑ Técnicas de aquisição automática são necessárias
 - Linguística computacional
 - Machine-learning: gerar ontologias a partir de documentos estruturados
 - Explorar a web para gerar estruturas

Análise

- Consistência semântica
 - Violação de restrições
 - Hierarquia cíclica
 - Termos que são usados mas não definidos
 - (min > max)
- estilo
 - Classes com uma única subclasse
 - Classes e slots sem definições
 - Slots sem restrições
- Ferramentas para análise automática

Avaliação

- ❑ Um dos maiores problemas no desenho de ontologias
- ❑ O desenho é subjetivo
- ❑ Como saber se uma ontologia está correta
- ❑ O melhor teste é a aplicação da ontologia

Manutenção de ontologias

- Mesclagem
 - Duas ontologias criam uma nova
- Mapeamento
 - Criar um mapeamento entre ontologias
- Versionamento e evolução
 - Compatibilidades entre versões da mesma ontologia
 - Compatibilidades entre versões da ontologia e das instâncias

Ferramentas

- ❑ Suporte a várias linguagens
- ❑ Expressividade
- ❑ Usabilidade

Por onde seguir?

□ Tutoriais

- Natalya F. Noy and Deborah L. McGuinness (2001) "Ontology Development 101: A Guide to Creating Your First Ontology"
http://protege.stanford.edu/publications/ontology_development/ontology101.html
- Farquhar, A. (1997). Ontolingua tutorial.
<http://ksl-web.stanford.edu/people/axf/tutorial.pdf>
 - We borrowed some ideas from this tutorial

□ Metodologia

- Gómez-Pérez, A. (1998). Knowledge sharing and reuse. Handbook of Applied Expert Systems. Liebowitz, editor, CRC Press.
- Uschold, M. and Gruninger, M. (1996). Ontologies: Principles, Methods and Applications. Knowledge Engineering Review 11(2)

Perguntas?

Obrigado!

www.fernando.parreiras.nom.br