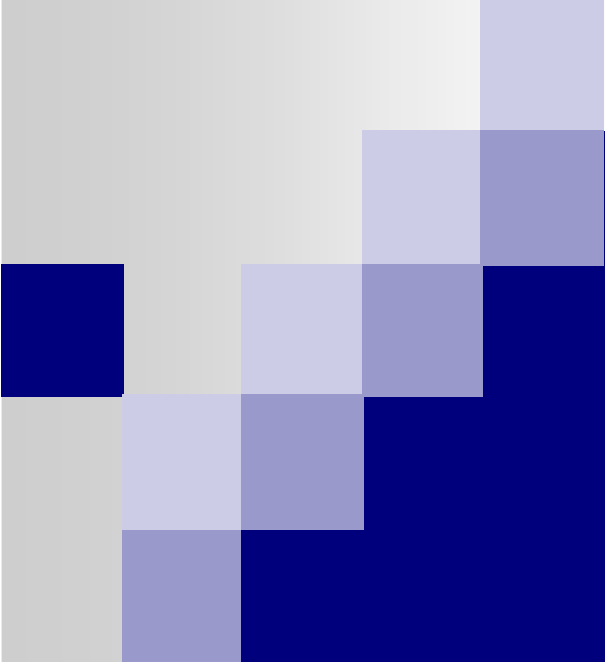
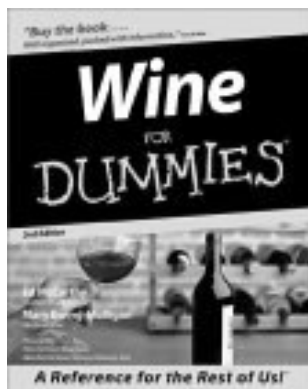




Ontológiák építése

Ontology Engineering

A Protege OWL eszköz modellezéshez javasolt
módszertan és példák áttekintése a labor előtt
(forrás: www.protege.stanford.edu)



**Ontológia
borokról és
ételekről**





Vázlat

- Mi is az ontológia?
- Miért fejlesszünk ontológiákat?
- Ontológiák építése lépésről lépésre
- Alaposabb elemzés: problémák és megoldások
- Ontológiák a szemantikus web eszközeivel
- Ontológiák fejlesztésének jelenlegi irányai



Mi az ontológia?

- Az ontológia egy tárgyterület explicit leírása:
 - ☐ Fogalmak
 - ☐ Fogalmak attribútumai és jellemzői
 - ☐ Attribútumok és jellemzők kényszerei
 - ☐ Egyedek
- Az ontológia tartalma
 - ☐ Közös szótár
 - ☐ Egy megosztható értelmezés



Ontológia példák

- Weben elérhető taxonómiák
 - Yahoo! kategóriák
- On-line boltok kategóriái
 - Amazon.com termék katalógus
- Tárgyterület függő standard terminológiák
 - Unified Medical Language System (UMLS)
 - UNSPSC – termékek és szolgáltatások terminológiái



Vázlat

- Mi is az ontológia?
- Miért fejlesszünk ontológiákat?
- Ontológiák építése lépésről lépésre
- Alaposabb elemzés: problémák és megoldások
- Ontológiák a szemantikus web eszközeivel
- Ontológiák fejlesztésének jelenlegi irányai



Miért fejlesszünk ontológiát

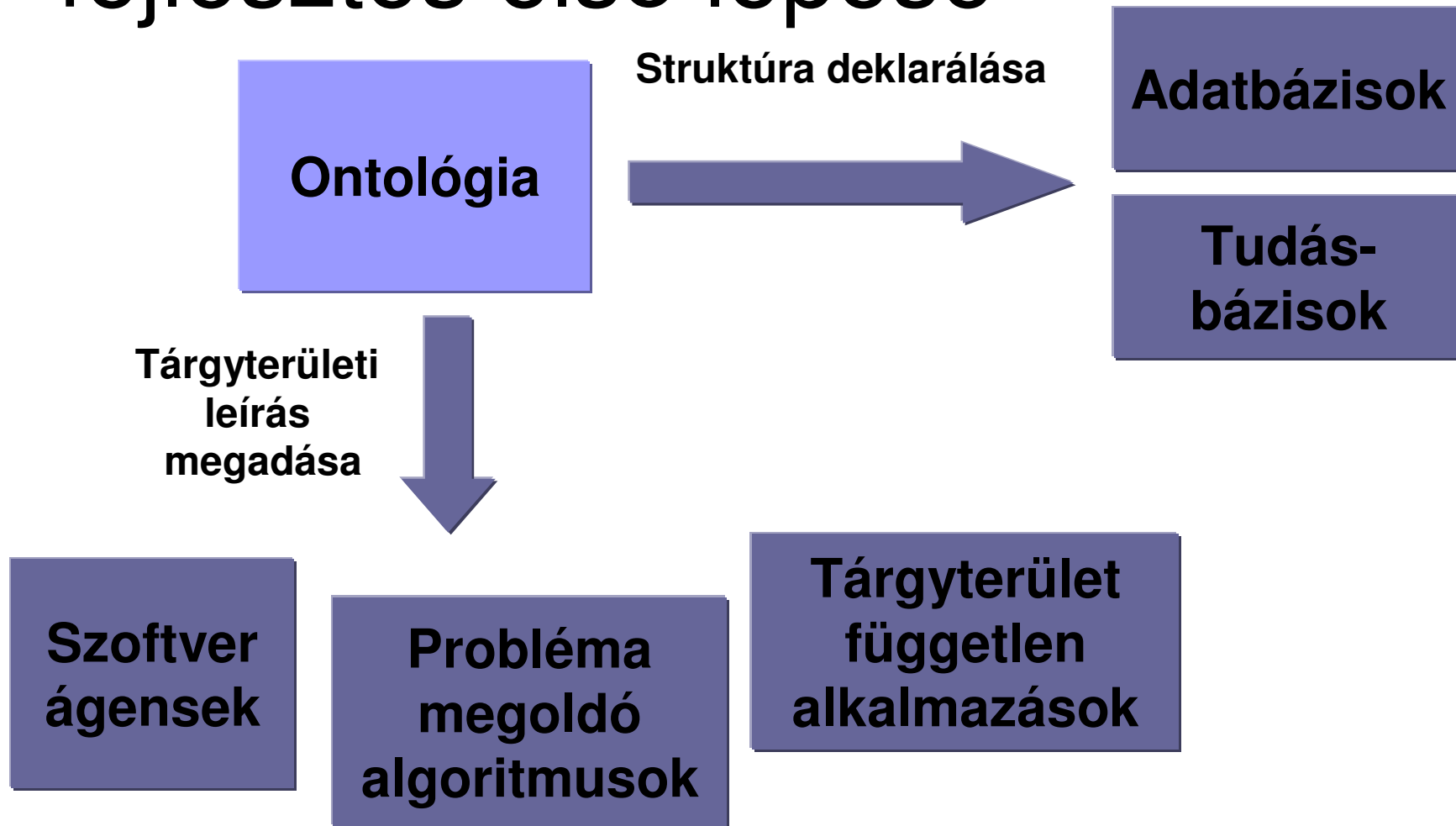
- Az információk struktúrájáról lévő értelmezés megosztása
 - ☐ emberek között
 - ☐ szoftver ágensek között
- Egy tárgyterületről leírt ismeretek újrafelhasználhatóságának biztosítása
 - ☐ elkerüljük a kerék újrafelfedezését
 - ☐ standardok bevezetése a osztott feladatmegoldás támogatásához



További érvek

- Tárgyterületi feltételezések explicit leírása
 - ☐ könnyebb változtatathatóság a tárgyterület leírásában
 - ☐ Könnyebb megérteni és módosítani az öröklött információkat
- Tárgyteületi tudás szétválasztása a működéssel kapcsolatos tudástól
 - ☐ A tárgyterületi és operatív tudás újrafelhasználhatósága külön-külön (kényszerekkel leírt konfigurációk)

Az ontológia van amikor a fejlesztés első lépése

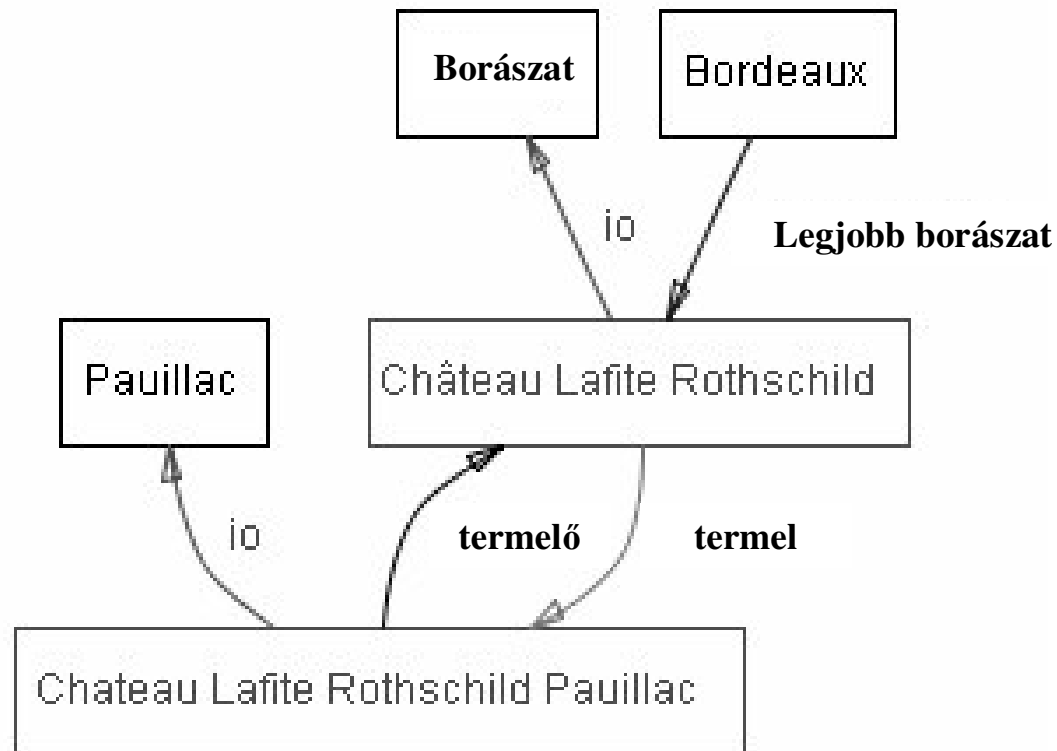




Vázlat

- Mi is az ontológia?
- Miért fejlesszünk ontológiákat?
- **Ontológiák építése lépésről lépésre**
- Alaposabb elemzés: problémák és megoldások
- Ontológiák a szemantikus web eszközeivel
- Ontológiák fejlesztésének jelenlegi irányai

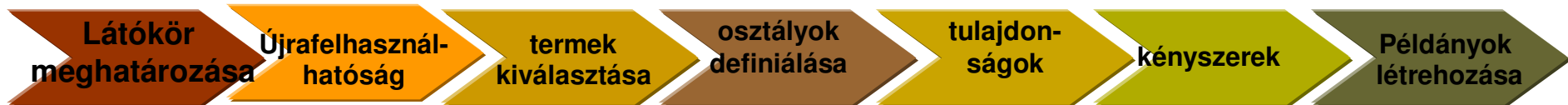
Borok és borászatok





Ontológia fejlesztési folyamat

Fontosabb lépések:



Valóságban egy iteratív folyamat:





Ontológiai építés vs. objektumorientált programstruktúrák

Ontológia

- a világ struktúrájának jellemzése
- fogalmak struktúrája
- nem foglalkozunk a fizikai reprezentációval

Egy OO osztály struktúra

- Az adat és a kód struktúrájának leírása
- Viselkedés struktúrája (metódusok)
- Az adatok fizikai reprezentációjának leírása (long int, char, etc.)



Eszközök (előzetes)

■ Protege 2000:

- ☐ Grafikus ontológia fejlesztő eszköz
- ☐ Gazdag tudás modell támogatása
- ☐ Nyitott forráskód, szabadon elérhető
(<http://protege.stanford.edu>)

■ Más eszközök:

- ☐ Ontolingua és Chimaera
- ☐ OntoEdit
- ☐ OilEd

Tárgyterület és látókör meghatározása



- Ontológia tárgyterületének megválasztása?
- Mire kívánjuk használni az ontológiát?
- Milyen típusú kérdésekre kell tudni választ adni (kompetencia kérdések)?

Afejlesztés ideje alatt ezekre a kérdésekre adott válaszok változhatnak



Kompetencia kérdések

- Milyen jellemzők alapján választunk bort?
- A bordói borok vörös vagy fehér borok?
- A Cabernet Sauvignon illik-e halételhez?
- Mi a legjobb választása sült húsokhoz?
- A bor milyen jellemzői befolyásolják, hogy illik-e egy ételtípushoz?
- Változik-e egy bor íze, testessége az évjárat függvényében?
- Melyek voltak Napa Zinfandel legjobb évjáratai?



Újrafelhasználhatóság



- Miért fontos az újrafelhasználhatóság?
 - ☐ ráfordítások csökkentése
 - ☐ más ontológiákat használó eszközökkel kooperálás
 - ☐ Olyan ontológiákat használhassunk, amelyek igazolhatóak más alkalmazások által



Mit akarunk újrafelhasználni?

■ Ontológia könyvtárak

- ☐ DAML ontológia könyvtár (www.daml.org/ontologies)
- ☐ Ontolingua könyvtár
(www.ksl.stanford.edu/software/ontolingua/)
- ☐ Protégé ontológia könyvtár (protege.stanford.edu/plugins.html)

■ Felsőbb ontológiák

- ☐ IEEE Standard Upper Ontology (suo.ieee.org)
- ☐ Cyc (www.cyc.com)



Mit akarunk újrafelhasználni?(II)

■ Általános ontológiák

- ☐ DMOZ (www.dmoz.org)
- ☐ WordNet (www.cogsci.princeton.edu/~wn/)

■ Tárgyterületi ontológiák

- ☐ UMLS Semantic Net
- ☐ GO (Gene Ontology) (www.geneontology.org)



Fontos termék felsorolása



- Mik azok a termék amiket használni akarunk?
- Milyen tulajdonságai vannak ezeknek?
- Mit akarunk ezekről kijelenteni?



Termek felsorolása – A bor ontológia

*wine, grape, winery, location,
wine color, wine body, wine flavor, sugar
content*

*white wine, red wine, Bordeaux wine
food, seafood, fish, meat, vegetables,
cheese*



Osztályhierarchia definiálása



- Egy osztály egy fogalmat jelöl
 - ☐ Borok osztálya
 - ☐ Borászatok osztálya
 - ☐ Vörös borok osztálya
- Egy osztály hasonló tulajdonságú elemek gyűjteménye
- Osztály példányok
 - ☐ Egy pohár burgundi vörösbor



Öröklődés osztályok között

- Taxonómiai hierarchiába rendeződnek az osztályok (subclass-superclass)
- IS-A hierarchia:

Egy példány tekinthető egy alosztálynak

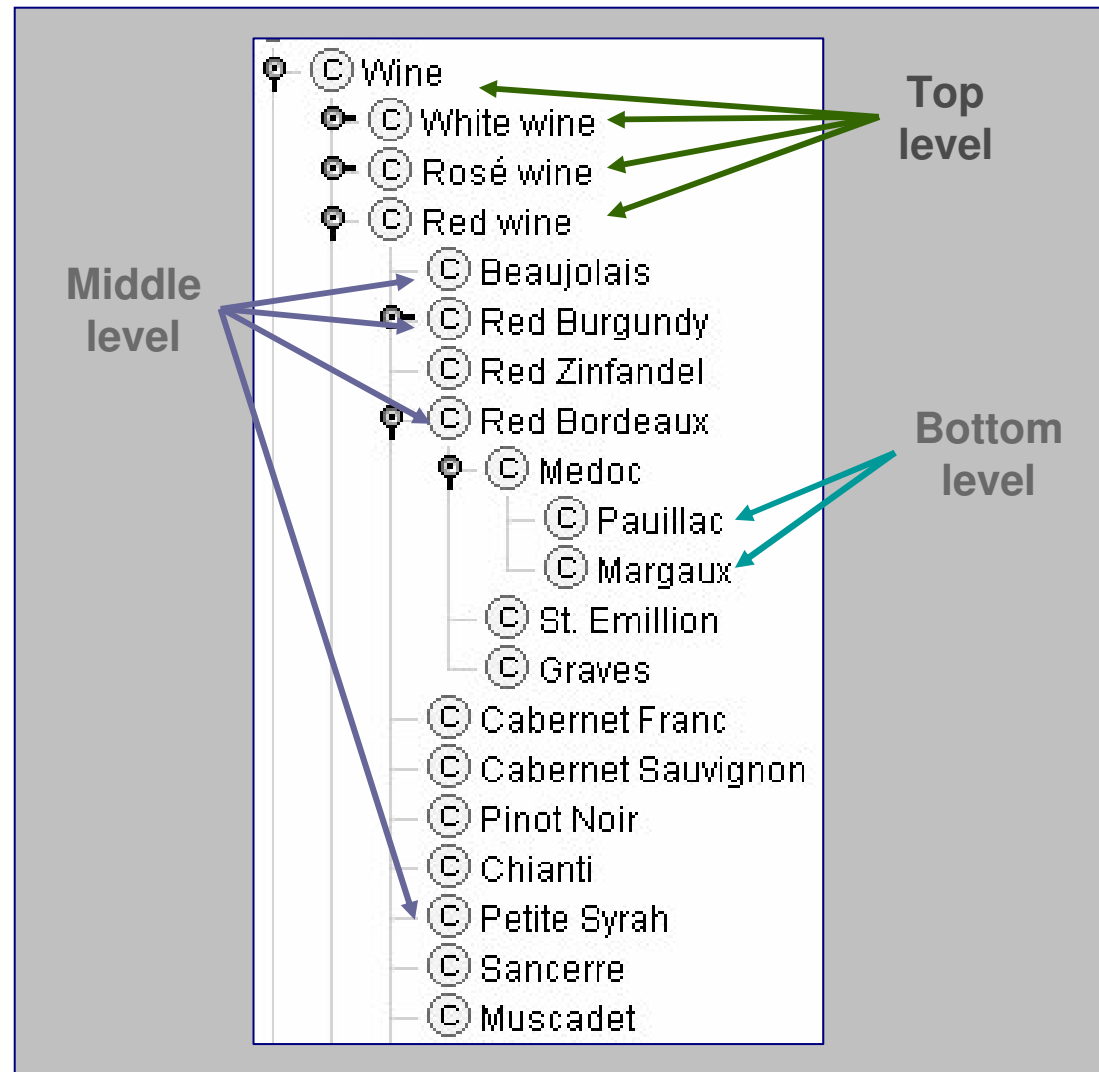
- Az osztályokat halmazokként is definiálhatjuk, egy alosztály egy részhalmaz



Öröklődés - Példa

- Az alma a gyümölcsök egy alosztálya
Every apple is a fruit
- Vörös borok a borok alosztálya
Every red wine is a wine
- Chianti bor a vörös borok egy alosztálya
Every Chianti wine is a red wine

Hierarchia szintek





Fejlesztési megközelítések

- top-down – általános elemeket definiálunk először és aztán specializáljuk ezeket
- bottom-up – a legspecifikusabb fogalmakat definiáljuk és aztán megkeressük az összetartozásokat
- kombinált – a legtipikusabb elemeket megkeressük, ezt általánosítjuk, majd ezek speciális példányait is bevesszük



Dokumentáció

- Osztályokat (és slot-okat) rendszerint dokumentáljuk
 - ☐ Osztály leírása természetes nyelven
 - ☐ Tárgyterületi feltételezések felsorolása
 - ☐ Szinonímák felsorolása
- Az osztályok és slot-ok dokumentálása hasonlóan fontos mint a programkód kommentezése!

Osztályok tuéajdonságainak definiálása –slot-ok



Az osztályok slot-jai a példányok attribútumait és a példányok relációit írják le.

Each wine will have color, sugar content, producer, etc.



Tulajdonságok (Slot)

■ Tulajdonság típusok

- ☐ “belső” tulajdonságok: bor zamata és színe
- ☐ “külső” tulajdonságok: a bor neve és ára
- ☐ részek: egy étel összetevői
- ☐ Más objektumokhoz kapcsolatok: a bor készítője (borász)

■ Egyszerű és összetett tulajdonságok

- ☐ Egyszerű tulajdonságok (attribútumok): egyszerű értékeket tartalmaznak (stringek, számok)
- ☐ Összetett tulajdonságok: tartalmaznak (vagy hivatkoznak) más objektumokat

A bor osztály attribútumai

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 ^I	Instance	single	classes={Winery}					
	name	String	single						
	sugar	Symbol	single	allowed-values={DRY,SWEET,OFF-DRY}					

Protégé-2000)



Attribútum és osztály öröklődés

- Egy alosztály megörököl minden attribútumot

Ha a bornak van színe és zamata, akkor a vörösbornak is van színe és zamata.

- Ha több szülőosztálya van egy osztálynak, akkor megörökli mindegyik szülő tulajdonságait

Egy portói bor desszert bor is és vörös bor is.



Tulajdonság kényszerek



- Tulajdonság kényszerek (facets) leírják vagy korlátozzák a felvehető értékek körét

A bor neve egy string

A bor gyártója egy példánya a Borászat osztálynak

Egy borászat pontosan egy helyhez tartozik

Kényszerek a bor osztályhoz

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 ^I	Instance	single	classes={Winery}						
name	String	single							
sugar	Symbol	single	allowed-values={DRY,SWEET,OFF-DRY}						



Tipikus kényszerek

- Slot számosság – a felvehető értékek száma
- Slot érték típus – a felvehető érték típusa
- Minimum és maximum értékek – értékkészlet
- Alapértelmezett érték – a slot által felvett érték, ha nem definiálják máshogy.



Tipikus kényszerek: Slot számosság

■ Számosság

- ☐ Az N számosság azt jelzi, hogy a slot-nak N értéke kell, hogy legyen

■ Minimum számosság

- ☐ Általában az 1 érték azt jelzi, hogy kell, hogy legyen egy értéke
- ☐ A 0 érték azt jelzi, hogy a minimum számosság tetszőleges

■ Maximum számosság



Tipikus kényszerek: Érték típus

- String: “Château Lafite”
- Szám: 15, 4.5
- Boolean: igaz vagy hamis
- Enumeráció: megengedett értékek listája (magas, közepes, alacsony)
- Komplex típusok: egy másik osztály példánya
 - Meg kell adni az osztályt, amihez a példány tartozik

A bot osztály gyártó slot-ja a borászatok közül veszi fel értékét

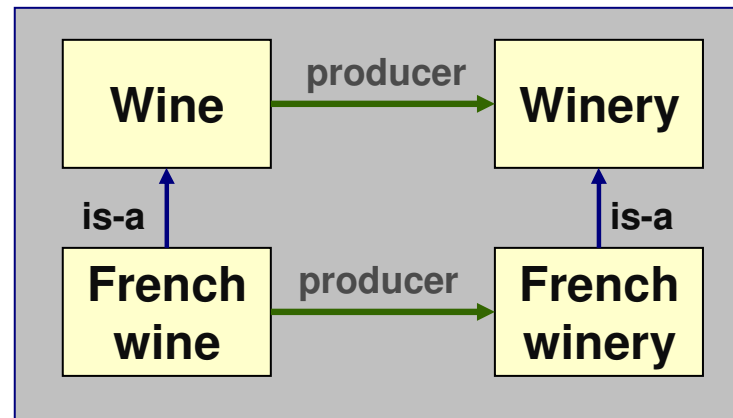


Tárgyterület és értékkészlet

- Tárgyterület (Domain) – az osztály(ok), mi tartalmazza a tulajdonságot
Pontosabban: osztálpéldányok, amik felvehetik az értéket
- Értékkészlet (Range)– az osztály, amhez a rekesz tartozik

Kényszerek és öröklődés

- Egy alosztály örökli a tulajdonságot a szülőosztálytól
- Egy alosztály felülírhatja a kényszereket, hogy szűkítse az értékkészletet





Példányok létrehozása



- Egy osztály példányának létrehozása
 - ☐ Az osztály és bármely szülőosztály a példány közvetlen típusa lesz
- Attribútum értékek meghatározása a példányokhoz
 - ☐ Az értékeknek teljesíteniük kell a rájuk vonatkozó ékényszereket

Egy példány létrehozása

Chateau Morgon Beaujolais (Beaujolais)

Name		Area
Chateau Morgon Beaujolais		Beaujolais region
Body	Color	Maker
LIGHT	RED	Chateau Morgon
Flavor	Sugar	Grape
DELICATE	DRY	Gamay grape
Tannin Level		
LOW		



Vázlat

- Mi is az ontológia?
- Miért fejlesszünk ontológiákat?
- Ontológiák építése lépésről lépésre
- Alaposabb elemzés: problémák és megoldások
- Ontológiák a szemantikus web eszközeivel
- Ontológiák fejlesztésének jelenlegi irányai



Osztályok és hierarchiák tervezése

■ Fontos:

- ☐ Nincs egy egyetlen helyes osztályozása
- ☐ De vannak iránymutatások

■ Ellenőrző kérdés:

“Egy osztály minden példánya tagja-e az összes szülőosztálynak?”

Az osztályhierarchia tranzitivitása

- Az is-a kapcsolat tranzitív:

B is-a A

C is-a B

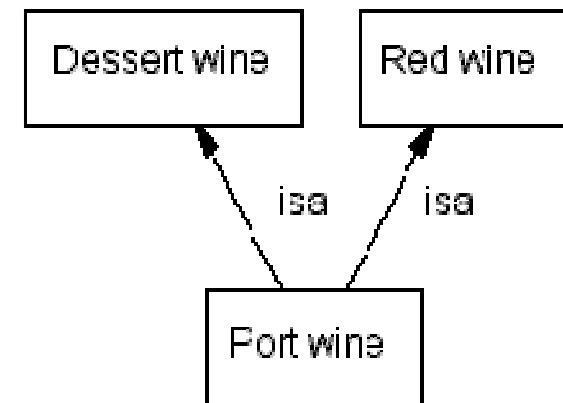
C is-a A

- Egy közvetlen szülőosztály a legközelebbi szülőosztály



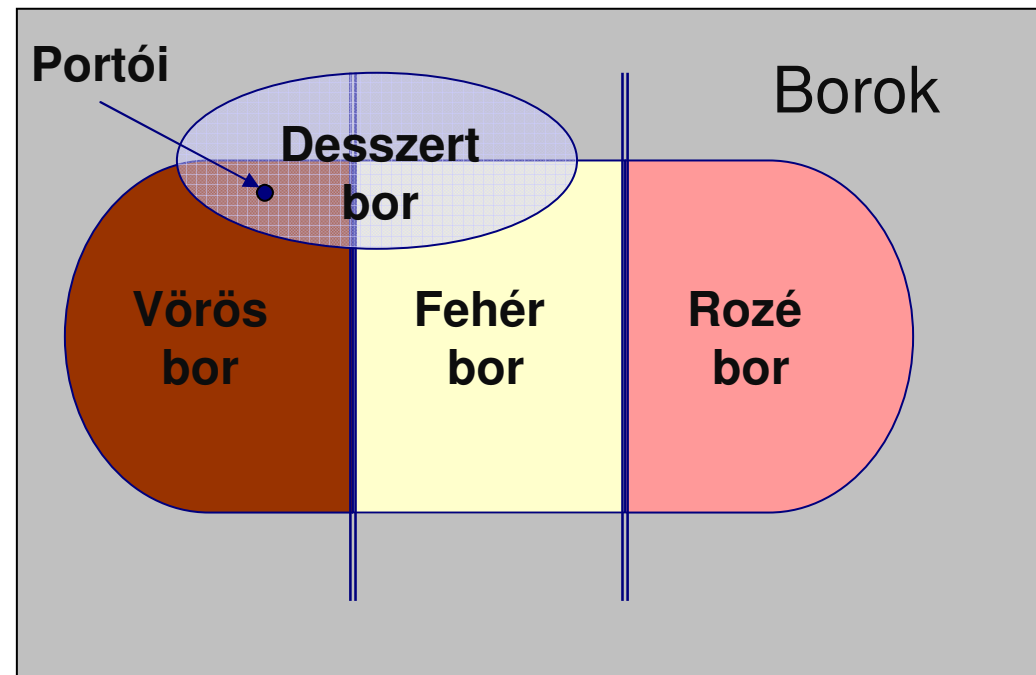
Többszörös öröklődés

- Egy osztálynak több mint egy szülőosztálya létezhet
- A konfliktusok feloldása nehéz probléma

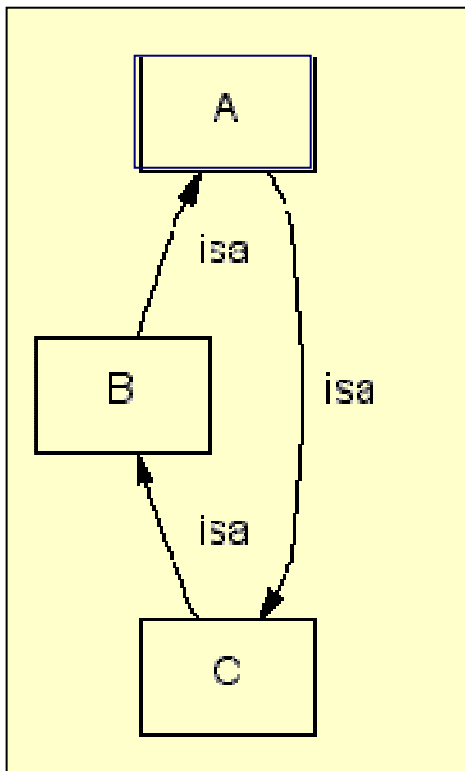


Kizáró osztályok

- Osztályok kizárják egymást, ha nincs közös példányuk
- Kizáró osztályoknak nem lehet közös alosztályuk sem



Ciklusok elkerülése az osztályhierarchiában



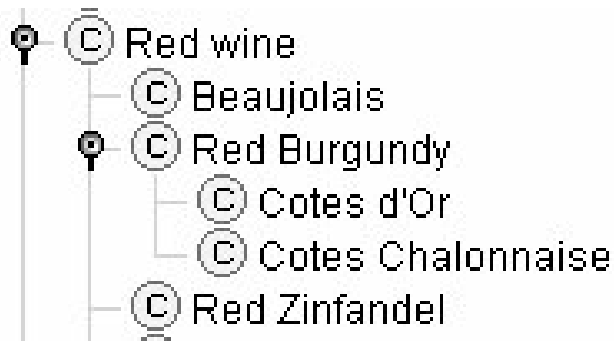
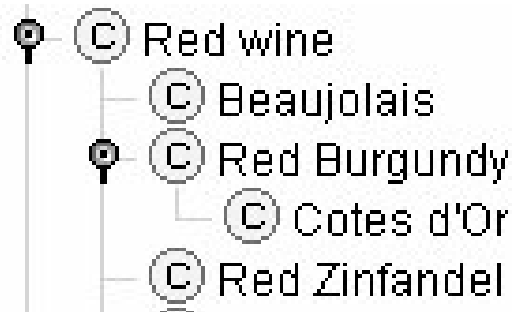
- Többszörös öröklődés veszélye: ciklusok az osztályhierarchiában
- Az ábra A, B és C osztályai elvileg ekvivalensek, mert ugyanazok a példányai

Kedvencek az osztályhierarchiában



- Minden azonos szintű fogalomnak egy hierarchia szinten kell lennie

Ideális méret megválasztása



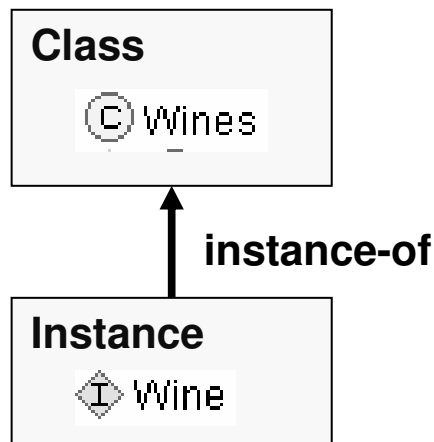
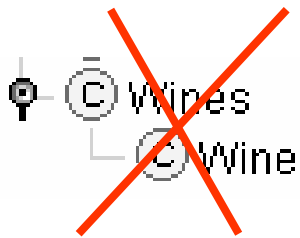
- Ha egy osztálynak egy gyereke van az valószínűleg helytelen modellezése
- Ha csak egy példány van, akkor minek neki osztály?

Ideális méret megválasztása (II)



- Ökölszabály: ha egy osztálynak egy tucatnál több alosztálya van érdemes egy közbülső szintet felvenni
- Lehet azonban egy hosszú lista természetes is.

Azonos osztálynevek problémája



- A bor ne legyen egyfajta bor is
- Esetleg lehet példány, de az sem szerencsés
- Osztálynevek lehetnek
 - ☐ egyszeresek
 - ☐ többszörösek



Osztályok és neveik

- Az osztályok fogalmakat reprezentálnak és nem a nevüket
- Az osztálynév akár változhat is egy fogalomhoz
- Synonym names for the same concept are not different classes
- Sok rendszer megenged szinonim nevek felvitelét

Egy teljes hierarchia a borokhoz



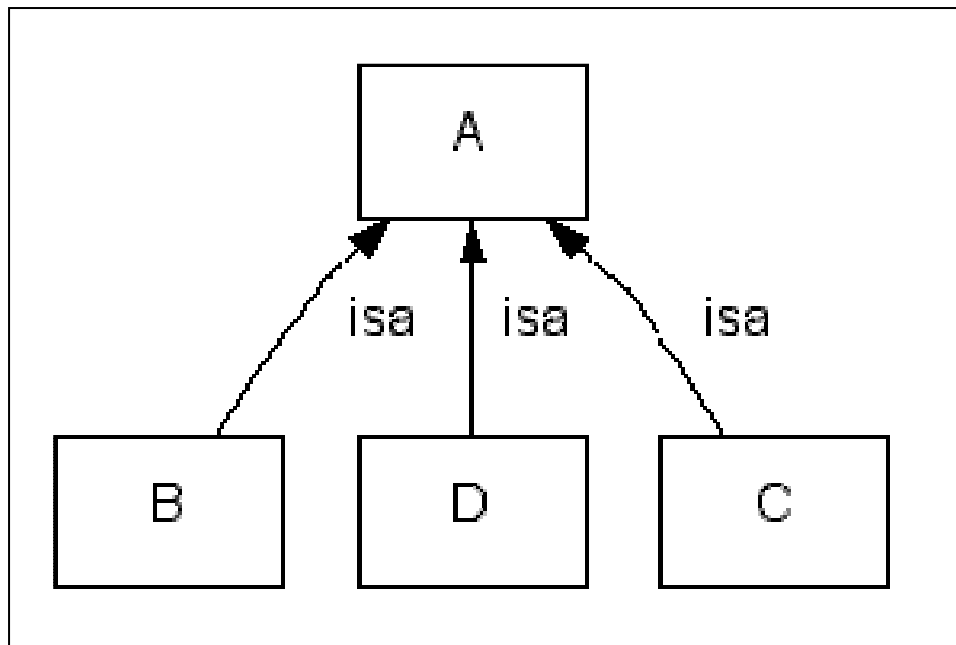
Attribútumok:

Tárgyterület és értékkészlet

- A tárgyterület és értékkészlet meghatározásához találjuk meg a legáltalánosabb szülőosztályt
- Attribútum jellegének megérzése
 - Tárgyterület: Vörös bor, fehér bor
 - Tárgyterület: bor
- Például a termék slot-ra:
 - Értékkészlet: Vörös bor, Fehér bor, ...
 - Értékkészlet: Bor



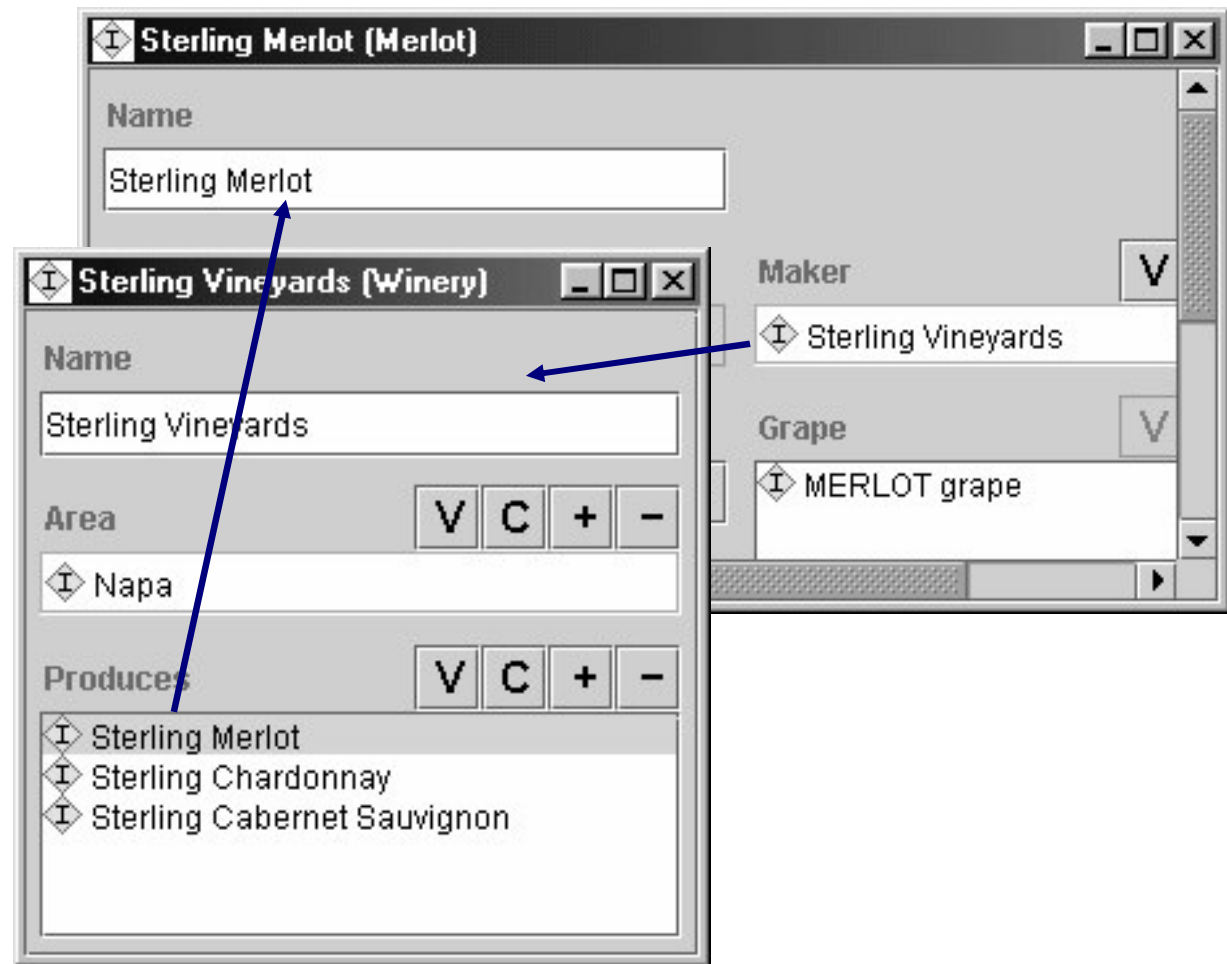
Tárgyterület és értékkészlet meghatározása



- Ha ugyanaz az osztályra és szülőjénél, akkor helyettesítsük a szülővel

Inverz attribútumok

A gyártó például
egy inver slot.





Inverz slot-ok (II)

- Az inverze slot-ok redundáns információt tartalmaznak
 - ☐ Többirányú információ elérést engednek meg
 - ☐ További ellenőrzést tesznek lehetővé
- A megvalósítás eltérő a rendszerekben
 - ☐ Mindkét értéket tároljuk?
 - ☐ Az inverz értéket mikor töltjük fel?
 - ☐ Mi történik, ha az inverz érték valamelyik kapcsolatát változtatjuk?



Alapértelmezett értékek

- A létrehozáskori érték
- Változtatható
- Nem szükséges, de általában tipikus



A vizsgált terület korlátozása

- Az ontológiának nem szabad tartalmaznia minden tárgyszerületi információt
 - Az adott alkalmazás határozza meg az általánosítás és a specializáció szintjét
 - Nincs szükség minden tulajdonság felsorolására
 - Alapvető információk
 - Alkalmazás által igényelt speciális információk



A vizsgált terület korlátozása(II)

- A példa tárgyterület valószínűleg nem tartalmaz
 - ☐ Üveg méret
 - ☐ Címke jellemzése
 - ☐ Az én kedvenc borom
- Biológiai vizsgálatokról szóló ontológia tartalmaz:
 - ☐ Biológiai organizmusok
 - ☐ Vizsgálat végzője
- A vizsgálat végzője egy biológiai organizmus?



Szemantikus web nyelvek ontológiák építéséhez

- A szemantikus web nyelvek kifejezetten ontológiák leírására lettek tervezve
 - ☐ RDF Schema
 - ☐ DAML+OIL
 - ☐ SHOE
 - ☐ XOL
 - ☐ XML Schema



SzW nyelvek

- A nyelvek különböznek

- szintakszis

- (nem elsődleges kérdés)

- terminológia

- Osztály - koncepció
 - Példány - objektum
 - Slot- tulajdonság

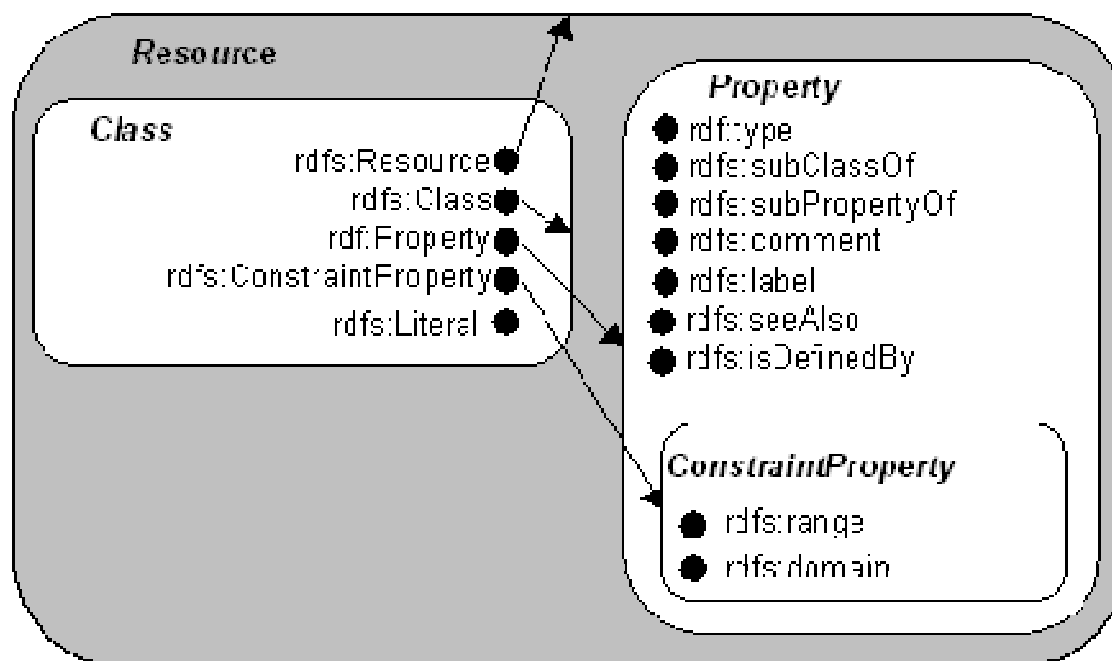
- kifejezőerő

- Van amit az egyik nyelvben ki tudunk fejezni, míg a másokban nem

- szemantika

- Ugyanaz az állítás más jelentést hordozhat

RDF és RDF Séma osztályok



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



RDF(S) terminológia és szemantika

- Osztályok és osztályhierarchiák
 - Minden osztály leszármazottja az `rdfs:Class` osztálynak
 - Osztályhierarchia: `rdfs:subClassOf`
- Példányok
 - Definíció: `rdf:type`
- Tulajdonságok
 - A tulajdonságok globálisak:
Közös névtérben a különböző osztályok azonos nevű tulajdonságai azonos jelentéssel bírnak
 - Tulajdonság-hierarchiák (`rdfs:subPropertyOf`)



Tulajdonság kényszerek RDF(S)

- Számosság kényszerek
 - ☐ Nincs explicit számosság kényszer
 - ☐ Bármely tulajdonság több értéket vehet fel
- Tulajdonság értékkészlete
 - ☐ Egy értékkészlet lehet
- Tulajdonság tárgyterülete
 - ☐ Több tárgyterülethez tartozhat
- Nincsenek kezdeti értékek



Kutatási irányok az ontológiák fejlesztésével kapcsolatosan

- Tartalom generálás
- Analízis és kiértékelés
- Karbantartás
- Ontológia nyelvek
- Eszközök fejlesztése



Tartalon: Top-Level Ontológiák

- Mit jelent a “felső-szint”?
 - ☐ Objektumok: érzékelhetők, nem érzékelhetők
 - ☐ Folyamatok, események, szereplők, szerepek
 - ☐ Ágensek, szerveződések
 - ☐ Terek, korlátok, helyek
 - ☐ Idő
- IEEE Standard Upper Ontology próbálkozás
 - ☐ Cél: Tervezni upper-level ontology-t a mérnöki terület igényei szerint
 - ☐ Folyamat: Meglévő ontológiák összeválogatásával



Tartalom: Tudásmegszerzés

- Tudás összegyűjtése a szűk keresztmetszet
- Megosztás és újrafelhasználhatóság segít a problémán
- Automatikus tudáskinyerési módszerre is szükség volna:
 - Nyelvtechnológia: ontológia kinyerése szövegekből
 - Gépi tanulás: ontológiák létrehozása strukturált szövegekből (pl. XML dokumentumok)
 - A Web struktúra kibővítése: ontológiák létrehozása Web portálok átolvasásával
 - Tudás kinyerési sémák: a szakértők definiálják egy részét az ismerteknek



Analízis

- Analízis: szemantika konzisztenciája
 - ☐ Tulajdonság kényszerek megsértése
 - ☐ Ciklusok az osztályhierarchiában
 - ☐ Nem definiált, de használt kifejezések
 - ☐ Intervallum korlátok megsértése (min > max)
- Analízis: stílus
 - ☐ Osztályok egyetlen alosztállyal
 - ☐ Osztályok és slotok definíció nélkül
 - ☐ Slot-ok kényszerek nélkül (típus, számosság)
- Automatikus analízis eszközök
 - ☐ Chimaera (Stanford KSL)
 - ☐ DAML Validator



Kiértékelés

- Egyik legbonyolultabb feladat az ontológia tervezésben
- Az ontológia tervezés szubjektív
- Mikor mondhatjuk, hogy egy ontológia helyes (objektívan)?
- A legjobb teszt az alkalmazás építése



Ontológia karbantartás

- Ontológiák összeillesztése
 - Két vagy több átfedő ontológia összerakása
- Ontology leképezés
 - Ontológiák közötti leképezés építése
- Verziókezelés
 - Kompatibilitás biztosítása a különböző verziók között
 - Kompatibilitás biztosítása a verziók és a példányok között