# ONTOLOJİ

1. **Bir ontoloji neden geliştirilir?**

Temel amaç bilginin paylaşımının kolaylaştırılmasıdır.

* Bilginin yapısına ilişkin genel anlayışın insanlar ve yazılımlar arasında paylaşılması
* Bir alanda elde edilen bilgi birikiminin yeniden kullanılmasına imkan sağlamak
* Bir alanla ilgili varsayımları açık/anlaşılır hale getirmek
* Alan bilgisi ile operasyonel bilgiyi ayırmak
* Alan bilgisini analiz etmek

1. **Bir ontolojinin içerisinde neler vardır?**

* Sınıflar (classes, concepts), kavramların tanımlarıdır.
* Özellikler (slots, properties, roles), kavramın çeşitli niteliklerini açıklar.
* Slotların kısıtlamaları (facets, restrictions on slots, role restrictions), slotlara girilebilecek değerlerin tipleri, en büyük/küçük vb. sınırlardır.
* Sınıfların benzersiz örnekleri - nesneler (individual instances of classes, objetcs), sınıfların bireysel örnekleridir, özelliklerinde değerler taşırlar.

1. **Basit bir bilgi mühendisliği metodolojisi ile bir ontoloji oluşturulurken kabul edilen temel kurallar nelerdir?**

* Bir alanın modelini oluşturmak için tek bir doğru yol yoktur. Her zaman kullanılabilecek/işe yarar alternatifler vardır. En iyi çözüm, planlanan uygulamaların ihtiyaçlarını karşılayabilecek özellikleri içerendir.
* Ontoloji geliştirme iteratif (yinelemeli) bir süreçtir.
* Ontolojideki kavramlar ve aralarındaki ilişkiler, gerçek hayattaki fiziksel/mantıksal nesnelere ve ilişkilere yakın olmalıdır.

1. **Ontoloji oluşturmanın birinci adımı olan “ontolojinin alanını ve kapsamını belirleme” aşamasında hangi basit soruların cevapları aranır?**

* Ontolojinin kapsamı/kapsayacağı alan nedir?
* Ontoloji hangi amaçla kullanılacak?
* Ontolojideki bilgiler hangi tür sorulara cevap vermelidir?
* Ontoloji kim kullanacak, kim çalışabilirliğini sürdürecek?

1. **Herhangi bir alanda oluşturulacak ontoloji için yeterlilik sorularına örnekler veriniz.**

* Bir şarap seçimi yapılırken şarabın hangi karakteristik özellikleri dikkate alınmalıdır?
* Bordo şarabı kırmızı şarap mıdır yoksa beyaz şarap mıdır?
* Cabernet Sauvignon şarabı deniz ürünleri ile iyi gider mi?
* Izgara et için en iyi şarap seçimi hangisidir?
* Şarabın hangi karakteristik özellikleri bir yemeğe uygunluğunu etkiler?
* Belirli bir şarabın tadı ve kokusu bağbozumu yılına göre değişir mi?
* Napa Zinfandel şarabı için en iyi bağ bozumu yılları hangileriydi?

1. **Ontoloji oluşturmanın ikinci adımı olan “var olan ontolojilerin kullanılıp kullanılmayacağını değerlendirme” aşamasını açıklayınız.**

Proje kapsamında belirlenen hedef alan için daha önce oluşturulmuş kaynaklar var ise bu ontolojiler işlenip proje amacına uygun olacak şekilde genişletilebilir. Oluşturulacak sistem diğer uygulamalar ile etkileşime geçecek ise bu uygulamaların ontolojilerine de ihtiyaç duyulabilir. Birçok ontoloji zaten elektronik ortamda olduğundan kullanılan ontoloji geliştirme aracı (örneğin proteqe) bu ontolojileri proje sistemine (import) ekleyebilir. Var olan ontolojileri listeleyen web sitelerinden (<http://www.ksl.stanford.edu/software/ontolingua/> --- <http://www.daml.org/ontologies/>) istenen kaynaklara ulaşılabilir.

1. **Ontoloji oluşturmanın üçüncü adımı olan “önemli kavramları listeleme” aşamasını açıklayınız.**

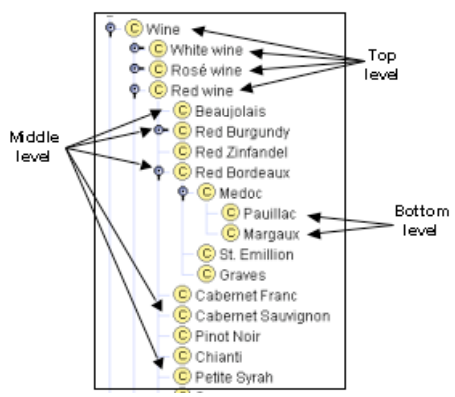
Ontolojisi oluşturulmak istenen alan ile ilgili tüm terimlerin listesi yapılır. Hakkında konuşulmak istenen terimler, bu terimlerin hangi özelliklere sahip oldukları, bu terimlerle ilgili söylenilmek istenenler bu listeye yazılır. Örnek projedeki şarap alanında, şarap, üzüm, şaraphane, yer, şarabın rengi, tat, aroma, koku, şeker seviyesi, farklı türlerdeki yemekler (örneğin balık, kırmızı et vb.), beyaz şarap/ kırmızı şarap vb. şarabın alt türleri bu listeye eklenir. Bu adımdan sonraki iki adım olan sınıfların hiyerarşilerini belirlemek ve sınıfların (kavramların) özelliklerini tanımlamak adımları (ki ontoloji oluşturmanın en önemli iki adımı) aşamalarından önce herhangi bir hiyerarşi, ilişki, özellik kısıtı olmadan bu listeyi oluşturmak önemlidir.

1. **Ontoloji oluşturmanın dördüncü adımı olan “sınıfları ve sınıflar arasındaki hiyerarşiyi tanımlama” aşamasını açıklayınız.**

Sınıflar arasındaki hiyerarşiyi oluşturmada 3 tür yaklaşım vardır.

* Yukarıdan aşağı geliştirme yaklaşımında, öncelikle en genel kavram tanımlanır. Daha sonra daha amaca yönelik alt kavramların tanımlanmasıyla devam edilir. Örnek projede öncelikle şarap tanımlanır. Daha sonra şarabın ve yiyeceklerin genel tanımlarının yapıldığı alt kavramlar (kırmızı şarap, beyaz şarap, gül şarabı, ızgara et, balık vb.) tanımlanır. Sonraki aşamalarda da bu alt kavramların da alt kavramları tanımlanarak devam edilir. Örnek projede Kırmızı şarabın alt kavramları Syrah, Red Burgundy, Cabernet Sauvignon … olarak gösterilmiştir.
* Aşağıdan yukarıya geliştirme yaklaşımında, öncelikle en spesifik kavramın tanımı yapılır. Örnek projede Pauillac şarabı için bir sınıf oluşturulmuştur. Daha sonra Margaux şarabı için de bir sınıf oluşturulmuştur. Bu iki şarap için Medoc kavramının karşılığı bir sınıf oluşturulmuştur. Örnek projede Medoc sınıfının üst sınıfı Red Bordeaux sınıfıdır.
* Kombinasyon geliştirme yaklaşımı yukarıdan aşağıya ve aşağıdan yukarıya geliştirme yaklaşımlarının bir karışımıdır. Öncelikle en göze çarpan kavramların tanımları yapılarak sınıflar oluşturulur. Daha sonra genelleştirilerek yukarı çıkılabileceği gibi özelleştirilerek aşağı da inilebilir. Her katmanda en belirgin kavramların tanımları oluşturulup sınıflar belirlendiğinden daha sonra eksik olanlar yavaş yavaş sisteme eklenirler.

Bu üç yaklaşımdan herhangi birinin diğerlerinden daha iyi olduğu söylenemez. Araştırmacı kendine göre istediği yöntemi seçer ancak genellikle kombinasyon yaklaşımı kullanılmaktadır. Sınıf hiyerarşisini doğru belirlemek çok önemlidir zira bu hiyerarşiye bakılarak önemli varsayımlarda bulunulacaktır. Örneğin *A sınıfı B sınıfının üst sınıfı (superclass) ise, B sınıfının her örneği (nesnesi) aynı zamanda A sınıfının da bir örneği* olacaktır. Başka bir deyişle *B aslında bir çeşit A*’ (kind of) dır.

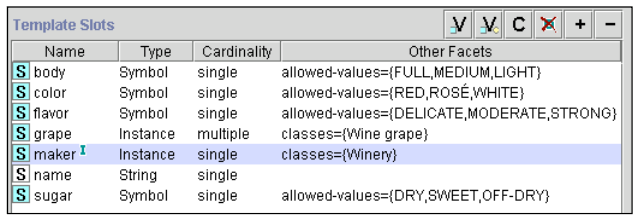


1. **Ontoloji oluşturmanın beşinci adımı olan “sınıfların özelliklerini/slotlarını tanımlama” aşamasını açıklayınız.**

Sınıflar ve sınıf hiyerarşisi belirlendikten sonra, terimler listesinde geriye kalan terimler muhtemelen özelliklerdir. Örnek projede bu terimlerden bazıları şarabın rengi, tadı, kokusu, aroması, şeker içeriği, üretildiği yerin konumu olabilir. Bu özelliklerin hangi sınıfları tanımlandıkları belirlenmeli ve o sınıfların slotları olarak atanma işlemleri yapılmalıdır. Örnek projede şarap sınıfının sahip olacağı slotlar, renk, yoğunluk, tat, şeker olabilir. Bir ontolojide slot olabilecek, genel olarak birkaç özellik vardır. Bunlar;

* Bir şarabın yoğunluğu, aroması gibi içsel özellikler
* Bir şarabın adı, üretildiği yer gibi dışsal özellikler

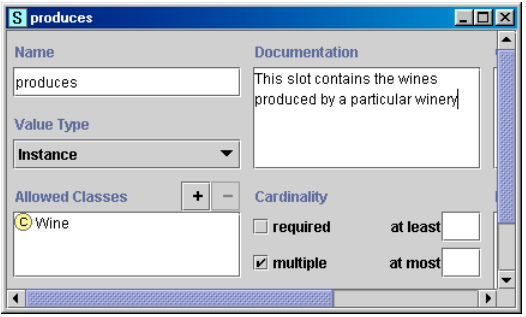
Bunlar, geldiği yer özelliği gibi, eğer nesne yapılandırılmış ise *parçalardan oluşabilir*. Sınıfın bireysel öğelerinin (nesne) *diğer nesneler ile ilişkileri* (örneğin üretici, şarap ile şaraphane arasında ilişki iken üzüm şarabın neyden üretildiğini gösterir) de olabilirler.



Bir sınıfın tüm alt sınıfları o sınıfın slotlarını devralır. Örnek projede şarap sınıfının yoğunluk, renk, tat, şeker gibi slotları doğrudan kırmızı, beyaz ve gül şaraplarının sınıflarına da aktarılacaktır. Kırmızı şarap sınıfında miras kalan bu slotlara ek olarak tanen seviyesi slotu, düşük, orta, yüksek değerlerinden birini alabilecek şekilde eklenir. Kırmızı şarabın alt sınıfları olan Bordeaux ve Beaujolais sınıfları doğrudan bu 5 slota sahip olacaklardır. Yeni bir slot eklenirken bu slota sahip olacak en genel sınıfa bu slot eklenmelidir.

1. **Ontoloji oluşturmanın altıncı adımı olan “slotların kısıtlarını tanımlama” aşamasını açıklayınız.**

Slotlar, veri tipleri (value type), izin verilen değerler (allowed values), kaç adet değer alınabileceği (cardinality) vb. kısıtlara (facets) sahiptir. Örnek projede wine sınıfının slotlarından biri olan name slotu değer olarak string bir ifade alabilirken, winery (şaraphane) sınıfının slotlarından biri olan produces (üretir) slotu, wine sınıfından üretilen nesnelerden birden çok değer alabilir. Winery sınıfında produces slotunun kısıtları tanımlanırken Type Instance, Cardinality multiple, Other Facets classes={Wine} girildiği tahmin edilebilir.



Slotların Cardinality kısıtına Minimum (en az N adet değer) ve Maksimum (en çok M adet değer) değerlerin girilmesine izin veren ontoloji geliştirme sistemleri de vardır. Maksimum cardinalitiy sıfır (0) olarak atanırsa bu sınıfın alt sınıflarının bu slota değer atayamayacağı anlamına gelir.

Slotların value type kısıtına String, Number (float tipinde), Boolean (true/false), Enumerated (protege de Symbol – aralarından değer seçilebilen liste), Instance (nesne) sınırlamaları girilebilir.

1. **Slotların Domain ve Range kavramlarını açıklayınız.**

Bir slotların değeri izin verilen sınıflardan üretilmiş bir nesne ile belirleniyor ise o sınıfa o slotun range’ i denir. Örnekteli produces slotunun range’ i olarak Wine sınıfı tanımlanmıştır. Slotun tanımlandığı sınıfa ise o slotun domain’ i denir. Bir sınıf bir slotu tanımlayan bir özelliğe sahipse de o sınıf o slotun domain’ i olur. Örnekte Winery (şaraphane) sınıfı produces slotunun domain’ idir.

Bir slot için Domain ve Range tanımlanırken en genel sınıf (örneğin ontolojideki en genel sınıf olan THING) seçilmemelidir. Range, slotun istediği değeri verecek nesnelerden daha fazlasını kapsamamalıdır.

Domain ve Range belirlenirken uygulanabilecek kurallar;

* *Eğer bir domaini veya range’ i tanımlayan bir sınıf listesi, bir sınıfı ve o sınıfa ait bir sınıfı içeriyorsa, alt sınıfı listeden çıkarın.*
* *Eğer bir domaini veya range’ i tanımlayan bir sınıf listesi, A sınıfın tüm alt sınıflarını içeriyor ancak A sınıfının kendisini içermiyorsa, A sınıfının alt sınıflarını listeden çıkarıp A sınıfının kendisini listeye ekleyin.*
* *Eğer bir domaini veya range’ i tanımlayan bir sınıf listesi, A sınıfın bazı alt sınıflarını içeriyor ancak A sınıfının kendisini içermiyorsa, alt sınıfların yerine A sınıfının daha iyi iş çıkarıp çıkarmayacağını araştırın.*

1. **Ontoloji oluşturmanın yedinci adımı olan “instansları/Nesneleri oluşturma” aşamasını açıklayınız.**

Bu son adımda hiyerarşide sınıflardan benzersiz örnekler (nesneler) oluşturulur. Benzersiz örnekler oluşturmak için;

* 1. Sınıfı seçme
  2. Seçilen sınıftan bir nesne oluşturma
  3. Nesnenin slotlarını değerler ile doldurma

adımları takip edilir.

Örnekte, Beaujolais sınıfından Chateau-Morgon-Beaujolais nesnesi oluşturulur. Bu nesnenin slotlarının değerleri aşağıdaki gibi girilir;

Body: Light

Color: Red

Flavor: Delicate

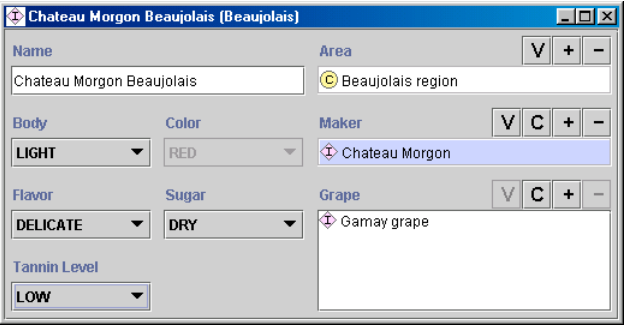
Tannin level: Low

Grape: Gamay (instance of the Wine grape class)

Maker: Chateau-Morgon (instance of the Winery class)

Region: Beaujolais (instance of the Wine-Region class)

Sugar: Dry



## MODEL HATALARININ KONTROLÜ

1. **Sınıfları ve sınıflar arasındaki hiyerarşiyi tanımlama aşamasında, “is – a” ilişkisini açıklayınız.**

Eğer A sınıfı B sınıfının alt sınıfı ise A sınıfından türetilen her nesne aynı zamanda B sınıfının da bir nesnesidir. “is-a” deyiminin diğer bir şekilde söylenişi “kind-of” deyimidir. Chardonnay şarabı bir Beyaz Şarap türüdür veya Chardonnay bir çeşit Beyaz Şaraptır denebilir (Chardonnay is a subclass of White wine - Chardonnay a kind of White wine).

Sınıf hiyerarşisi oluşturulurken yapılan hatalardan biri isimlerde tekil/çoğul tanımlamalardan kaynaklanır. En üst sınıf Wines yani şaraplar ismi ile tanımlanırken Wines’ ın alt sınıfı olarak Wine yani şarap tanımlanamaz. Yani bir şarap bir çeşit şaraplar’ dır cümlesi doğru olmaz (wine is not a kind of wines). Sınıf isimleri verilirken sürekli tekil veya sürekli çoğul isimler verilmelidir.

1. **Sınıfları ve sınıflar arasındaki hiyerarşiyi tanımlama aşamasında, “hiyerarşik ilişkilerin geçişkenliği” açıklayınız.**

Eğer B sınıfı A sınıfının alt sınıfı ise ve eğer C sınıfı da B sınıfının alt sınıfı ise bu durumda C sınıfı aynı zamanda A sınıfının da alt sınıfıdır.

A ᴐ B ise ve B ᴐ C ise 🡪 A ᴐ C

B sınıfı A sınıfının doğrudan alt sınıfıdır (Direct subclass) fakat C sınıfı A sınıfının alt sınıfı olmasına rağmen doğrudan alt sınıfı değildir.

1. **Sınıfları ve sınıflar arasındaki hiyerarşiyi tanımlama aşamasında, “sınıf hiyerarşisindeki evrimsel değişimi” açıklayınız.**

Üzerinde çalışma yapılan alan genişledikçe sınıf hiyerarşisinde tutarlılığı sağlamak zorlaşabilir. Örnek projede Zinfandel şarabı uzun yıllar kırmızı şarabın (red wine) alt sınıfı iken, şarap üreticilerinin yaptıkları bazı işlemler sonucu renkte değişmeler sağlanmıştır. Sonuç olarak Zinfandel şarabının rengi artık kırmızı veya beyaz (gül rengi) olabilmektedir. Öyleyse alan artık Zinfandel şarabının yeni bir sınıfına ihtiyaç duymaktadır. Bu durumda kırmızı şarabın (red wine) altında kırmızı Zinfandel (red Zinfandel) bulunurken, gül rengi şarabın (rose wine) altına beyaz Zinfandel (white Zinfandel) eklenmelidir.

1. **Sınıfları ve sınıflar arasındaki hiyerarşiyi tanımlama aşamasında, “sınıf isimleri” verilirken eş anlamlı kelimeler olduğunda dikkat edilmesi gereken noktalar nelerdir?**

Aynı kavramın eş anlamlıları farklı sınıfları temsil etmez. Hatıra ve anı kelimeleri aynı kavramı ifade ettiklerinden farklı sınıflara ihtiyaç duymazlar. Çoğu ontoloji geliştirme ara yüzleri bu tür eşanlamlı kelimelerin ilişkilendirilmesine izin verir. Kullanılan araçta bu özellik yoksa bile sınıf dökümanlarına eşanlamlı kelimeler eklenebilir.

1. **Sınıfları ve sınıflar arasındaki hiyerarşiyi tanımlama aşamasında, “kaçınılması gereken sınıf döngülerini” açıklayınız.**

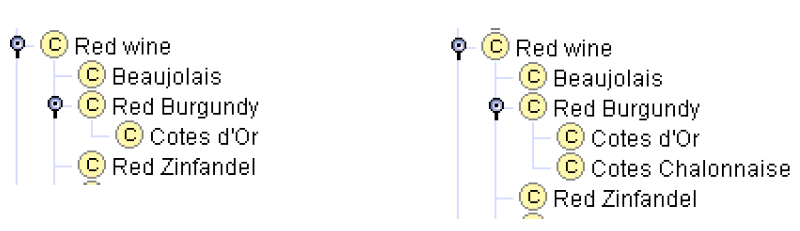
Sınıf hiyerarşilerindeki döngülerden kaçınılmalıdır. Bir A sınıfı bir B sınıfının alt sınıfı iken aynı zamanda B sınıfı da A sınıfının alt sınıfı ise bu bir döngü oluşturur. Bunun sonucunda A sınıfından üretilen her bir nesne aynı zamanda B sınıfının da nesnesi iken B sınıfından üretilen bir nesne aynı zamanda A sınıfının da bir nesnesi olacaktır ki bu da A ve B sınıfını eşit sınıflar yapar.

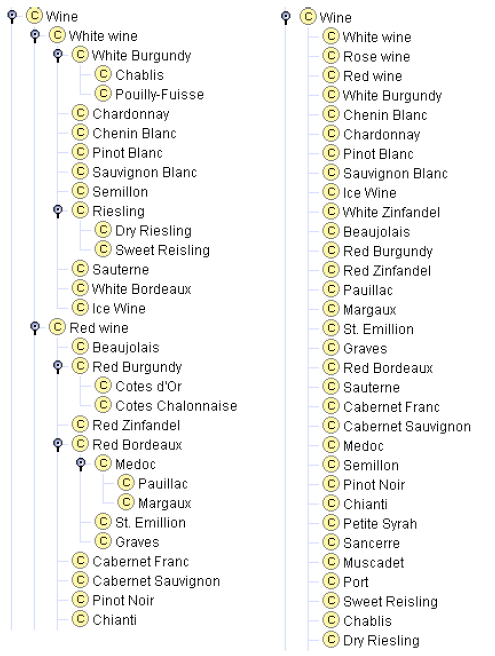
1. **Sınıfları ve sınıflar arasındaki hiyerarşiyi tanımlama aşamasında, sınıf hiyerarşisinde aynı düzeydeki kardeş (sibling) üyeler kavramını açıklayınız.**

Sibling’ ler aynı sınıfın doğrudan alt sınıflarıdır. Root hariç tüm siblingler aynı genellik düzeyindedirler. Chardonnay ve White wine aynı sınıfın alt sınıfı olamazlar çünkü White wine Chardonnay’ den daha genel bir kavramdır. Kitaplardaki içindekiler sayfası benzeri hiyerarşik bir yapıyı barındırır.

1. **Aynı düzeydeki kardeş üyelerin sayısı ne zaman çok fazla ne zaman çok az olarak nitelenebilir?**

* *Bir sınıfın sadece tek bir doğrudan alt sınıfı var ise ya modelde hata vardır ya da ontoloji henüz tamamlanmamıştır.*
* *Bir sınıfın bir düzineden fazla alt sınıfı var ise ara sınıflar oluşturmanın gerekliliği araştırılmalıdır.*





1. **Sınıfları ve sınıflar arasındaki hiyerarşiyi tanımlama aşamasında, “çoklu kalıtım” özelliğini açıklayınız.**

Birçok ontoloji geliştirme platformu çoklu kalıtıma izin vermektedir. Örnek projede Port isimli yeni bir sınıf Dessert wine ve Red wine sınıflarının ikisinin de alt sınıfı olarak üretilmektedir. Bu nedenle Port sınıfından üretilecek tüm nesneler aynı zamanda Dessert wine ve Red wine sınıflarının da nesneleri olacaktır. Port sınıfı Desset wine ve Red wine sınıflarının tüm slotlarını ve bu slotların facets’ lerini kalıtım yoluyla almaktadır. Örneğin Dessert wine’ ın Sugar slotundan SWEET değerini alırken, Red wine’ ın tannin level slotunu ve color slotunun renk değerini alacaktır.

1. **Sınıfları ve sınıflar arasındaki hiyerarşiyi tanımlama aşamasında, ne zaman yeni bir sınıf eklenmesine ihtiyaç duyulur/duyulmaz?**

* *Alt sınıf üst sınıfın sahip olmadığı ek bir özelliğe sahiptir*
* *Alt sınıfın kısıtları üst sınıftan farklıdır*
* *Alt sınıf üst sınıftan farklı ilişkilere katılır*

Bazen, yeni bir özelliğe sahip olmasa da yeni bir sınıf oluşturmak faydalı olabilir. Örneğin doktorların kullandığı bir terminolojide hiçbir özelliği olmayan bir sınıf bazı durumlarda;

* Aramayı ve gezinmeyi kolaylaştırabilir
* Terimlerin genelleştirilmesini kolaylaştırabilir

Her yeni kısıtlama için yeni bir sınıf oluşturmak şart değildir. Örnek projede Red wine, White wine, Rose wine olmak üzere 3 sınıf tanımlanmıştır çünkü şarap dünyasında bu durum doğaldır. Daha spesifik sınıflar tanımlanmadı. Ontolojide sadece red Port’ lar gösterildi çünkü white Port’ lar son derece nadir görülmektedir.

1. **Eklenecek yeni kavramın sınıf mı yoksa özellik değeri mi olduğuna nasıl karar verilir?**

Bir alan modellenirken farklılığı gösteren bir kriterin bir sınıf mı yoksa bir özellik mi olması gerektiğinin belirlenmesi gerekir. Örneğin beyaz şarap sınıfı mı oluşturulmalı yoksa genel bir şarap sınıfı oluşturulup renk slotuna beyaz, kırmızı vb. farklı değerler mi girilmelidir? Slot değerleri farklı kavramlar, diğer sınıflar için farklı değer kısıtlamaları alıyorlarsa o zaman ayrım için yeni bir sınıf oluşturulmalıdır. Aksi takdirde fark bir slot değerinde gösterilir.

Beyaz şarap kavramı modellenen alan için ne kadar önemlidir? Şarabın beyaz olup olmadığının, diğer nesneler ile ilişkisinde, bir çıkarım oluştururken özel bir önemi varsa beyaz şarap sınıfı oluşturulur. Eğer detaylı bir ontoloji oluşturulacaksa bu fark çok önemli olacaktır. Bir restorandaki, şarap ve yiyecek kombinasyonları belirlenirken şarabın rengi (hangi şarap hangi yiyecekle uyumludur) çok önemli olacaktır. Benzer şekilde şarabın tadım sırası belirlenirken de şarabın rengi önemlidir. Bu gibi durumlarda beyaz şarap ayrı bir sınıf olmalıdır.

Şarap etiketi üreten bir fabrika modellenirken, etiketi üretirken renk değeri üzerinden özel işlemler gerekmediğinden yani bütün renkler için aynı işlemler yapıldığından, rengin beyaz olup olmadığı o kadar da önemli olmadığından burada renk sadece bir özellik olarak tanımlanabilir.

*Farklı slot değerlerine sahip kavramların (örneğin şarap), farklı durumlarda sınıflarındaki slotlarının değişik kısıtlamaları (facets) oluyor ise yeni sınıf oluşturulmalıdır aksi taktirde yeni özellik tanımlanmalıdır.*

Örneğin Merlot sınıfının alt sınıfları olan Red Merlot ve White Merlot aynı üzümden üretilmelerine rağmen gerçekten çok farklı şaraplardır.

*O farklılık kriteri ayrım için çok önemliyse, farklı nesne türlerinde o ayrım kriteri farklı değerler alıyor ise, o ayrım kriteri için sınıf oluşturulmalıdır.*

Bireysel bir örnek olan nesnenin, ait olduğu sınıf sık sık değişmemelidir. Sınıflar arasındaki farkı göstermek için içsel özelliklerden ziyade dışsal özellikler kullanıldığında nesnenin ait olduğu sınıf değişebilmektedir. Örneğin soğutulmuş şarap bir sınıf olmamalıdır. Nesnenin “soğutulmuş” özelliği olmalıdır. Çünkü nesne olarak örnek bir şarabın ısısı hızla değişebileceğinden bu nesnenin sınıfının değiştirilmesi de gerekebilir. Genellikle sınıfların slot değerleri sayılar, renkler, üretim bölgeleri vb. olurlar. Ancak örnek projede şarabın rengi şarap için çok önemli olduğundan önemli bir istisnadır ve özellik olarak kullanılmak yerine sınıfı belirlemektedir.

Örneğin ayrıntılı bir insan anatomosi ontolojisi geliştirilmek isteniyorsa, 1. Sol kaburga, 2. Sol kaburga vb. her kaburga için bir sınıf oluşturulabilir çünkü her kaburgaya özgü belirli bir işlev, konum, koruduğu organ vb. vardır. Bu ayrıntılı ontoloji yerine daha genel bir ontoloji oluşturulmak isteniyorsa, örneğin sadece hangi kaburganın kırık olduğu bilgisi gerekli ise, sadece “yanal konum” ve “sıra” slotlarına sahip daha genel bir “kaburga” sınıfı yeterli olabilir.

1. **Eklenecek yeni kavramın sınıf mı yoksa nesne mi olduğuna nasıl karar verilir?**

Bir ontolojideki belirli bir kavramın bir sınıf mı yoksa bir nesne mi olacağına karar vermek o ontoloji kullanacak olan potansiyel uygulamalara bağlıdır. Sınıfların nerede biteceğine ve nesnelerin nerede başlayacağına, temsilde en küçük ayrıntı düzeyinin ne olduğuna karar vermekle başlanır. Bilgi tabanında (veri tabanında) temsil edilecek (gösterilecek/saklanacak) en alt seviye (spesifik) ögeler nelerdir sorusuna cevap vermek için ontolojideki yeterlilik soruları çok faydalı olacaktır.

*Bireysel örnekler (nesneler) bir bilgi tabanında temsil edilen en alt seviye kavramlardır.*

Örneğin şarap ve yemek eşleşmeleri ile ilgileniliyorsa, şarap şişelerinin fiziksel özellikleri en alt seviyede gösterilmeyecektir. Şarap sınıfları tek tek şarap şişelerini değil, belirli şarap imalathaneleri tarafından üretilen şarap koleksiyonlarını göstermektedir. Bu nedenle bu projede “Sterling Vineyards Merlot” en alt seviye terime örnek olup bilgi tabanında nesne olarak gösterilecektir.

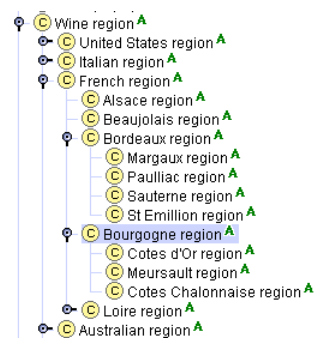
Ürünün her bir özel rekoltesi (yılı vb.) farklı özelliklerin bilgi tabanında tutulması isteniyorsa, tüm bağ bozumları için Sterling Vineyards Merlot sınıfından nesneler oluşturulur, bu nesneler bilgi tabanında gösterilir.

Öte yandan, iyi şarap-yemek eşleşmelerine ek olarak, bir restorandaki şarapların envanterinin tutulması isteniyorsa, her şarabın ayrı şişeleri bilgi tabanında birer nesne halinde gösterilebilir.

Bazı durumlarda bireysel örnek olan nesneler sınıflara dönüştürülebilirler.

*Kavramlar doğal bir hiyerarşi oluşturuyorlar ise bunlar sınıflar ile gösterilmelidir.*

Örneğin “şarap bölgeleri” (wine regions) kavramını ele alınırsa, ilk olarak Fransa, Birleşik Devletler, Almanya vb. ana şarap bölgeleri tanımlanabilir. Daha sonra bu ana şarap bölgelerinden “Fransa” sınıfından “Bourgogne region” nesnesi bireysel örnek olarak oluşturulabilir. Ancak “Cotes d’Or” bölgesi de bir Bourgogne bölgesi. Bu nedenle Bourgogne bölgesi bir sınıf olmalı (Alt sınıfı veya bir nesnesi var). Ancak Bourgogne bölgesini bir sınıf haline getirmek ve Cotes d’Or bölgesini onun bir nesnesi yapmak keyfi bir işlemdir. Hangi bölgelerin sınıf hangi bölgelerin nesne olarak gösterilmesi gerektiğini belirlemek oldukça zor görünüyor. Bu nedenle hepsi sınıf olarak tanımlanabilir. Protégé-2000, kullanıcıların bazı sınıfları soyut (abstract) sınıf olarak tanımlamasına imkan verir. Soyut sınıflar doğrudan örnek alamayan (nesne oluşturulamayan) sınıflardır. Örnek projede de tüm bölge sınıfları soyut sınıflardır.



Şarap bölgeleri (wine regions) “bölgeleri” kelimesi çıkarılırsa aynı sınıf hiyerarşisi yanlış olurdu çünkü “Alsace” sınıfının Fransa sınıfının bir alt sınıfı olduğu söylenemez. Alsace Bölgesi bir çeşit fransa bölgesi olsa da Alsace bir çeşit Fransa değildir. Terimler arasında bölüm 4.2 de anlatıldığı gibi doğal bir hiyerarşi var ise bu terimler sınıflar (kendilerine ait örnekler-nesneler olmasa da) olarak tanımlanabilirler.

1. **Ontolojinin kapsamının sınırlarına nasıl karar verilir?**

Bir ontolojinin tanımlanmasının, sınıf hiyerarşisi tanımlama aşamasının ne zaman biteceğinin belirlenmesinde bazı kurallar yardımcı olur.

*Ontoloji, o alana ait tüm olası bilgileri içermemelidir. Ontolojiyi kullanacak uygulamaların ihtiyacı olduğundan daha fazla ayrıntıya (özelleştirme-spesifikasyon) veya daha fazla genelleştirmeye (her iki yönden en fazla 1 seviye gidilebilir) gidilmemelidir. Örnek projede karideslerin nasıl pişirileceğine ya da şişelerin etiketlenmesinde hangi tür kağıdın kullanıldığı bilgilerine ihtiyaç yoktur.*

Örnek projede, bir yemeğin veya şarabın sahip olabileceği tüm özellikler ontolojiye dahil edilmiyor. Ontolojide sınıfların en belirgin özellikleri temsil edildi. Hatta şarap kitapları üzümün büyüklüğünden bahsetse bile projeye bu bilgi dahil edilmedi. Benzer şekilde, nesneler arasında tasavvur edilebilecek tüm ilişkiler oluşturulan modele eklenmedi. Örneğin projede “favori şarap”, “favori yemek” gibi bir ilişkiler bulunmamaktadır.

Biyoloji deneylerini tanımlayan bir ontoloji muhtemelen biyolojik organizma kavramını da içerecektir. Bu ontoloji deneyi yapan kişinin adı, üyeliği, resmi gibi bilgileri de içerecektir. Deneyi yapan kişi bir biyolojik organizma olmasına rağmen muhtemelen kendi üzerinde deney yapmayacağından kendi bilgilerini ontolojiye girmemelidir. Çünkü ontolojinin amacı deneyci üzerinde deney yapmak değildir. Bu tasarım kararı dokümantasyona yazılmalıdır çünkü ontolojiyi bulup kullanmak isteyecek kişi deneyciyi de ontolojinin içerisinde varsayarak ontoloji kullanabileceğini düşünebilir.

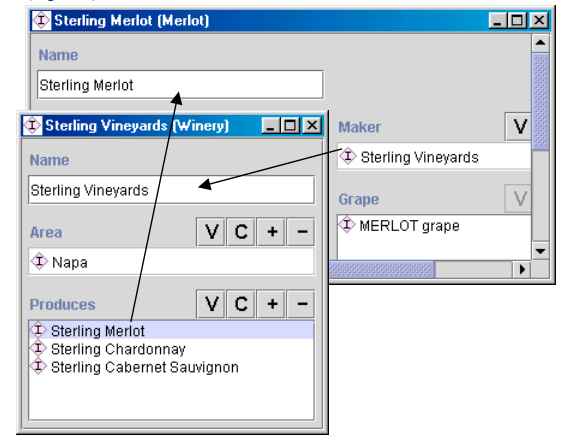
1. **Ayrık (disjoint) alt sınıf kavramını açıklayınız.**

Sınıfların ortak nesneleri yok ise bu sınıflar ayrık (disjoint) sınıflardır. Bir çok yazılım ayrık sınıfları açıkça belirtme yeteneğine sahiptir. Örnek projede tatlı şarabı (dessert wine) ile beyaz şarap (white wine) ayrık sınıflar değillerdir. Birçok şarap her iki sınıfın da örneğidir. “The Rothermel Trochenbierenauslese Riesling” şarabı (nesne) “Sweet Riesling” sınıfının da bir örneğidir. Örnek projedeki Kırmızı şarap ve Beyaz şarap sınıfları ayrık sınıflardır. Hiçbir şarap aynı anda hem kırmızı j-hem de beyaz şarap olamaz. Sınıfların ayrık olduklarının belirtilmesi sistemin ontoloji daha iyi doğrulamasını sağlar. Eğer ontolojide kırmızı şarap ve beyaz şarap ayrık sınıflar olarak tanımlanmış iseler, daha sonra “Riesling” (Beyaz şarabın alt sınıflarından biri) ve “Port” (Kırmızı şarabın alt sınıflarından biri) sınıflarından ortak bir alt sınıf oluşturulmaya çalışıldığında sistem modelleme hatası vererek uyarıda bulunacaktır.

## ÖZELLİK BELİRLEMEDE AYRINTILAR

1. **Özellikler tanımlanırken slotlar arasında kullanılabilen, iki yönlü ilişki (Inverse slots) kavramını açıklayınız.**

Bir slotun alacağı değer başka bir slotun değerine bağlı olabilir. Örneğin bir şarabın üreticisi bir şaraphane ise otomatik olarak bu şaraphanenin ürettiklerinden biri o şaraptır. Bu iki ilişki (üretici/maker ve üretir/produces) iki yönlü ilişki (inverse relations) olarak tanımlanır. Bilginin her iki yönde de saklanması gereksizdir. Bir şarabın bir şaraphane tarafından üretildiği bilgisi saklanıyor ise, ontolojiyi kullanacak uygulama programı, iki yönlü ilişkiyi kullanarak, bir şaraphanenin ürettiği şarapları bulabilir. Bununla birlikte, her iki bilginin de açıkça mevcut olması, bilgi tutarlılığının kontrolü için faydalıdır.



1. **Özellikler tanımlanırken kullanılabilen, varsayılan değerler kavramını açıklayınız.**

Birçok ontoloji geliştirme frame’ i slotlar için varsayılan değerler belirlenmesine izin verir. Eğer bir sınıfın belirli bir slotunun değeri, o sınıftan üretilen nesnelerin çoğunda aynı olacak ise, o değer slotun varsayılan değeri olarak tanımlanabilir. Bu sayede sınıftan nesne oluşturulduğunda kullanılan ara yüz o nesnenin o slotuna otomatik olarak o değeri girer. Nesnenin varsayılan olarak girilen bu değeri istenilirse nesne tarafında kısıtların izin vereceği şekilde değiştirilebilir. Sonuç olarak varsayılan değer, veri girişinde kolaylık sağlamak için vardır, yeni bir kısıt tanımlamak ya da modeli değiştirmek gibi bir amacı yoktur.

Slot değeri ile bu varsayılan özellik değeri karıştırılmamalıdır. Tatlı şarap (Dessert wine) sınıfının ve tüm alt sınıflarının (tabi ki tüm alt nesnelerinin de) “sugar” slotlarının değeri “SWEET” dir ve herhangi bir alt sınıfta veya nesnelerinde bu değer değiştirilemez.

Üzerinde konuşulan şarapların çoğunun “body” özelliğini değeri “full” ise, bu özelliğe varsayılan olarak “full” değeri girilebilir. Herhangi bir nesne oluşturulurken varsayılan gelen bu değer değiştirilebilir.

1. **Bir ontoloji oluşturulurken, sınıflara/slotlara isim verilirken, kullanılan ontoloji geliştirme sisteminin (örneğin proteqe) hangi özelliklerine dikkat edilmelidir?**

* *Sistem aynı ada sahip bir sınıfa, bir slota, bir nesneye izin veriyor mu (yoksa tek bir ad alanı-namespace mı kullanıyor)? Örneğin “saraphane” isimli bir sınıf varken “saraphane” isimli slota izin veriyor mu?*
* *Sistem büyük küçük harfe duyarlı mı?*
* *Sistem isimlerde hangi sınırlayıcılara izin veriyor (boşluk, virgül, yıldız vb.)*

Protégé-2000 tek bir ad alanı kullanır, büyük küçük harfe duyarlıdır. Bu nedenle hem “saraphane” sınıfı hem de “saraphane” slotu oluşturulamaz iken, “saraphane” sınıfı varken “Saraphane” slotu tanımlanabilir.

CLASSIC ise sınıflar, slotlar ve nesneler için farklı ad alanları kullanır, büyük küçük harfe duyarlı değildir. Bu nedenle hem “saraphane” sınıfı hem de “saraphane” slotu oluşturulabilir.

1. **İsim verme kurallarına örnekler veriniz.**

* ***Sınıf isimlerini büyük harflerle yazmak, slot isimlerini küçük harflerle yazmak gibi kurallar uygulanırsa ontolojinin okunurluğu kolaylaşabilir.***
* ***İsim için birden fazla kelime kullanmak gerektiğinde;***
  + ***Boşluk kullanmak (Yemek adi) // Boşluk kullanmak çok tercih edilmez çünkü bu ontolojiyi geliştiren platform boşluk kullanmayı destekliyor olsa da bu ontoloji kullanacak uygulama programları boşluk kullanmayı desteklemiyor olabilir.***
  + ***Her kelimenin ilk harfini büyük yapmak (YemekAdi)***
  + ***Alt çizgi kullanmak (Yemek\_Adi)***
* ***İsimlendirmede çoğul veya tekil kullanımdan sadece biri tercih edilmelidir. Sınıflar genellikle çok nesneye referans olacaklarından daha çok çoğul isimlendirme önerilir. Bazı sistemler daha projenin başında isimlendirme stilini (tekil/çoğul) sorarlar.***
* ***Bazı sistemler isimlendirmede önek/sonek önerirler. … has-winery veya maker-of … gibi isimlendirme uzun gibi görünse de bir bakışta sınıf ve slotları ayırt etmeye yardımcı olacaktır.***
* ***Kavramların isimlerine class, property, slot gibi isim uzantıları eklenmemelidir.***
* ***İsim kısaltmalarından kaçınılmalıdır.***
* ***Kavram isimleri verilirken benzer yapılar kullanılmalıdır. Red Wine, White Wine şeklinde olabilirken, Red Whine, White şeklinde kullanımdan kaçınılmalıdır.***