VERİ AKIŞ DİYAGRAMI – KAVRAMSAL SINIF DİYAGRAMI

Sistem Analizi ve Tasarımı Dersi

İçindekiler

Veri Akış Diyagramları 3
Veri Akış Diyagramları Çizim Kuralları 4
Taslak (Context) Diagram 5
Birinci Seviye Diyagram 6
Veri akış diyagramı çizimi 6
Kavramsal Sınıf Diyagramı 7
Sınıf Diyagramları Çizimi 8
Use Case Diagram 10
Kullanım senaryosu diyagramı çizimi 13

Veri Akış Diyagramları

Veri akış diyagramları

1. dışsal birim (varlık)
2. proses (işlem)
3. veri deposu
4. veri akışı

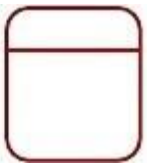
olmak üzere dört adet eleman kullanılarak çizilmektedir.

Dışsal Birim 

Veri kaynağı yada verinin varış noktaları olan sistem dışı olguları belirlemek için dikdörtgen kullanılır. Aynı varlık bir diyagramda tekrarlı olarak çizildiğinde, tekrarlı çizilen kaynağın kenarına çizgi koyarak, aynı varlığı temsil ettiği belirtilir.

Dış olgulara örnekler;

- Bir kişi, Müşteri veya Öğrenci.
- Bir şirket veya örgüt, Banka veya Tedarikçi.
- Örgüt içi bir birim, Satış departmanı.

Proses (İşlem) 

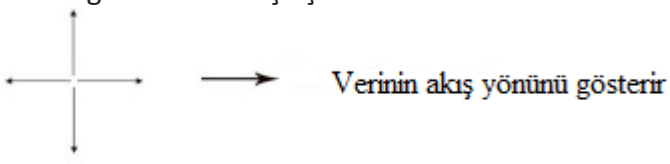
- Prosesler, yapılan bir fonksiyonu ya da işlemi tanımlar. Proseslere bir isim ve numara verilir. Proses ismi olarak emir cümleleri kullanmak uygun olacaktır (Net geliri hesapla gibi).

Veri Deposu 

- Analiz esnasında, verilerin depolanmasına ihtiyaç duyulan yerler olur. Bu yerler veri deposu olarak isimlendirilir. Veri deposu veritabanı tabloları, xml gibi veri saklama üniteleri olabilir. Her bir veri deposu için ayrıca bir de isim verilir.

Veri akışı

- Bir noktadan diğerine veri akışı için oklar kullanılır.

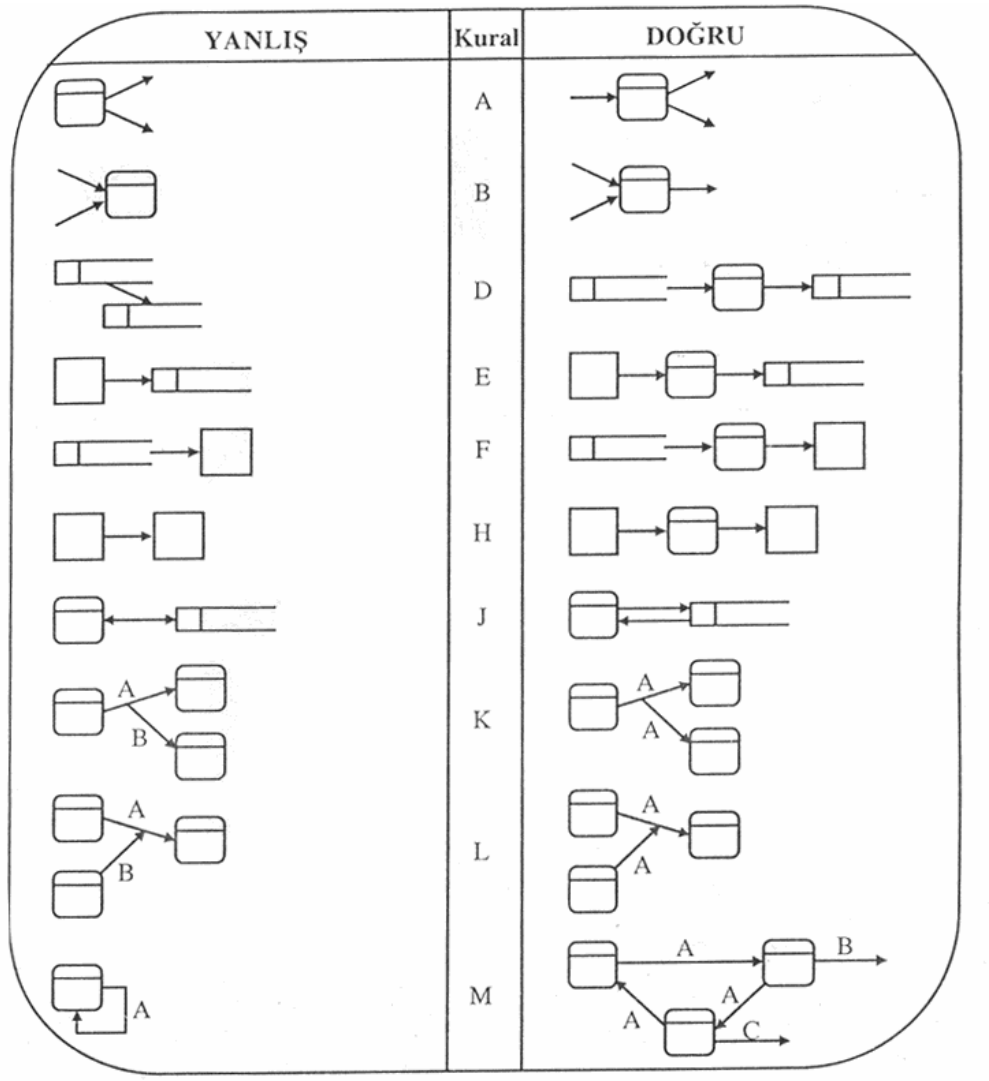


  - Sistemde hareket eden veriyi tanımlar. Veri bilgisi oka yazılarak belirtilir.
  - Ok ucu veri akış yönünü gösterir. Bir veri akışının veri deposuna gitmesinin anlamı, güncellemedir. Bir veri deposundan veri akışının çıkmasının anlamı, getirme ya da kullanmadır.

Veri Akış Diyagramları Çizim Kuralları

Veri akış diyagramları aşağıda belirtilen kurallara uygun olarak çizilmesi gerekmektedir.

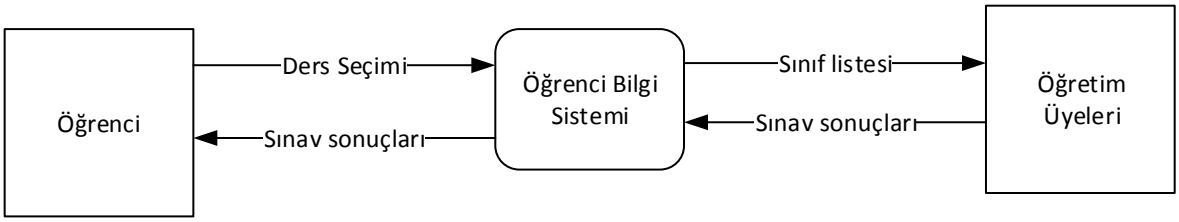
1. Hiçbir proses sadece çıktılara sahip olamaz. (Şekil 1. A)
2. Hiçbir proses sadece girdilere sahip olamaz. (Şekil 1. B)
3. Veri, bir veri deposundan diğerine doğrudan taşınamaz. Veri bir prosesle taşınmalıdır. (Şekil 1. D)
4. Veri, doğrudan bir dışsal kaynaktan bir veri deposuna taşınamaz. Dışsal birimden veriyi alan ve veri deposuna yerleştiren bir prosesle taşınmalıdır. (Şekil 1. E)
5. Veri, bir veri deposundan doğrudan bir dışsal birime taşınamaz. Veri bir prosesle taşınmalıdır. (Şekil 1. F)
6. Veri doğrudan bir varlıktan diğerine taşınamaz. (Şekil 1. H)
7. Bir veri akışı, semboller arasında tek bir akış yönüne sahip olmalıdır. Bir proses ve veri deposu arasında, veri deposundan okuma ve proseste güncellemenin gösterilmesi için her iki yönlü akış olabilir, ancak bunların iki ayrı ok şeklinde gösterilmesi gerekir. (Şekil 1.J)
8. Çatallı bir veri akışının anlamı, aynı verinin ortak bir lokasyondan iki ya da daha fazla farklı prosese, veri deposuna yada dışsal birime gitmesi demektir. (Şekil 1. K)
9. Veri akışlarının birleşmesinin anlamı, aynı verinin herhangi iki ya da daha fazla farklı prosesten, veri deposundan ya da dışsal birimden, ortak lokasyona gelmesidir. (Şekil 1. L)
10. Bir veri akışı, doğrudan aynı prosese geri dönemez. Veri akışını alıp, başka veri akışlarını üreten ve başladığı prosese orijinal veri akışını getiren en az bir prosesin olması gerekir. (Şekil 1. M)



Şekil 1. Veri akış diyagramı kuralları

Taslak (Context) Diagram

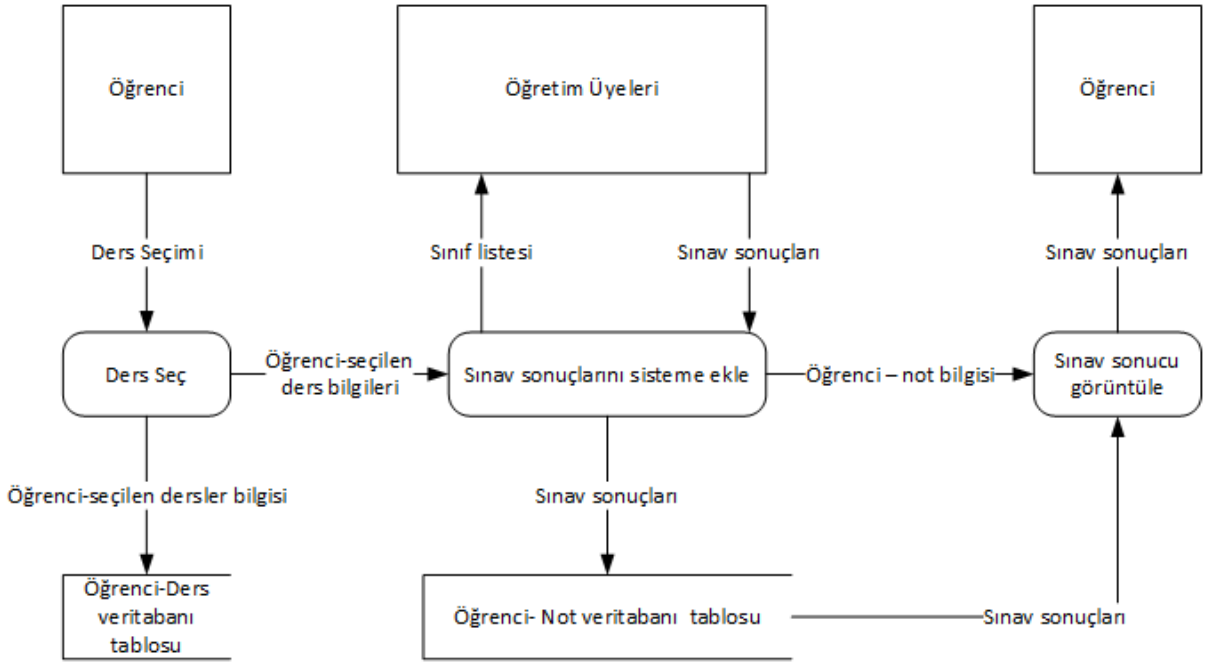
Sistemin dışsal birimlerinin sistem ile ilişkisini basitçe gösterildiği diyagramdır. Tek bir işlem olur ve sistemin ismi ile belirtilir.



Şekil 2. Örnek taslak diyagram

Birinci Seviye Diyagram

Taslak diyagramda çizilen grafiğin, bir seviye detaylandırma yapılan versiyonudur. Diyagram detaylandırma işleminde, veri akış ve diğer elemanlar, bir üst diyagram ile uyumlu olmalıdır. Örneğin, taslak diyagramda "öğrenci" varlığından sisteme doğru "ders seçimi" veri akışı var ise, alt seviyesinde de bu veri akışı olmalıdır.



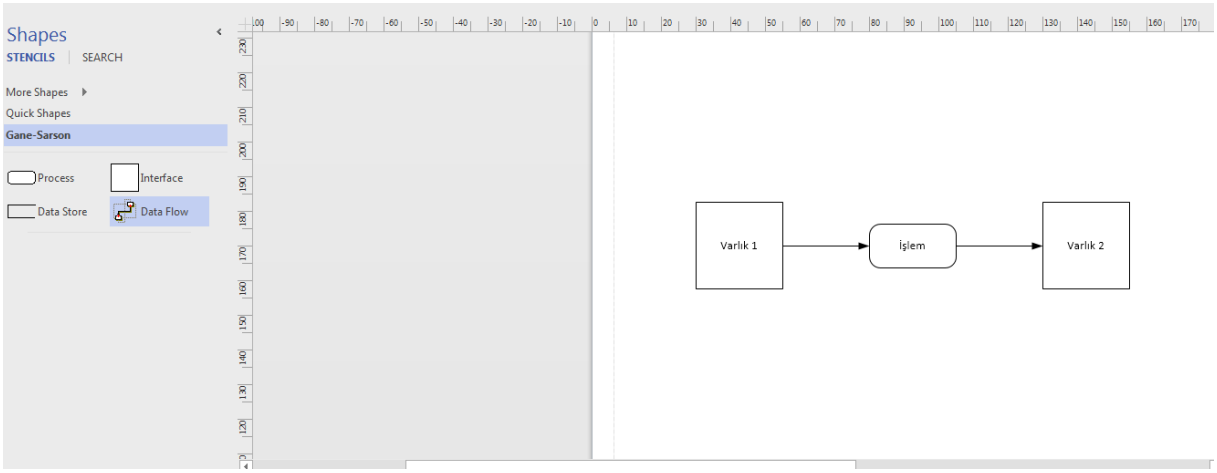
Şekil 3. Örnek birinci seviye diyagram

Veri akış diyagramı çizimi

Diyagram çizimi için Microsoft Visio programını kullanabiliriz. Visio programı çizilen diyagramın kurallara uygunluğunu denetlememektedir. MS Visio 2016'da veri akış diagramı kategorilerden "Software and Database" seçip, "Data flow model diagram" seçeneği seçilir. "Metric Units" seçilip, "create" butonuna basılır.

MS Visio 2003'de "Software" kategorisi içerisinde "Data Flow Model Diagram(Metric)" seçilir.

Veri akış diagramının dört elemanı sol taraftaki şekiller kısmında bulunmaktadır. "Varlık" elemanı "interface" olarak isimlendirilmiştir. Sürükle bırak ile çizim yapılmaktadır. Elemanların üstüne çift tıklayarak isimlendirme yapılmaktadır. Çizim sırasında ok yönüne dikkat edilmelidir.



Şekil 4. Veri akış diyagramı çizimi (MS Visio 2016)

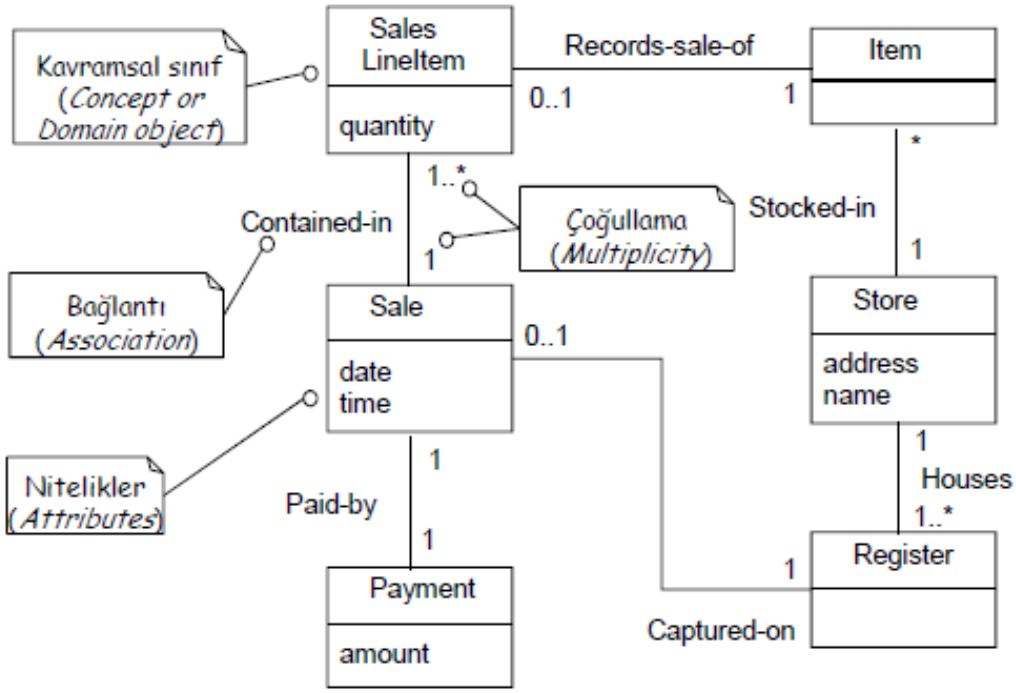
Çizim sırasında okların karışmamasına dikkat ediniz. Bunun için, aynı varlığı diyagramın diğer kısmında tekrardan yazarak, ok karmaşasının önüne geçilebilir. Bu durumda, diğer varlığın kenarına çizgi işareti konularak, aynı varlığı temsil ettiği belirtilir. MS Visio için bu özellik bulunmamaktadır, aynı isimlendirme yapmak yeterlidir.

Okları elemanlar ile eşleştirirken, veri akışı okunu, kaynak elemanın orta kısmına getirdiğinizde, elemanın üzeri yeşil olmaktadır, eşleştirmeyi bu şekilde yapınız. Böylece, veri akış oku o elemana bağlanmış olmaktadır. Elemanın yerini değiştirdiğimizde, bağlanan ok da onunla birlikte hareket eder. Kaynak elemana bağladıktan sonra, okun uç kısmından tutup uzatarak, hedef varlığın orta kısmına getirdiğimizde- elemanın üstü yeşil olacaktır – oku bırakarak, hedef varlığa bağlanır. Bu sayede elemanlar yer değiştirdiği zaman, bağlı olan veri akış okları da onlar ile birlikte hareket eder.

Ek olarak, ok karmaşasını önlemek için, fazla sayıda girdi-çıktıya sahip elemanları daha uzun çizerek, okları birbirinden uzaklaştırabiliriz. Elemanlara bağlı olan oklara tıklayıp, fare imlecini okun orta kısmında nokta ile belirtilen eklem yerine getirdiğimizde, fare imlecinde sağa sola hareket ettirebileceğimizi gösteren işaret çıkacaktır. Bu kısımdan sürükleyerek, elemanlara bağlantısını bozmadan, oku kaydırabiliriz.

Kavramsal Sınıf Diyagramı

Sistemde kullanılacak sınıflar, sınıf üyeleri ve sınıflar arası ilişkilerin tanımı ile oluşturulan diyagramdır. Kavramsal sınıf diyagramlarında, sistemin iç tasarımın ayrıntılarını göstermeden, gereksinimleri karşılayacak sınıfları, sınıfların niteliklerini ve sınıflar arası bağlantıları gösterir.

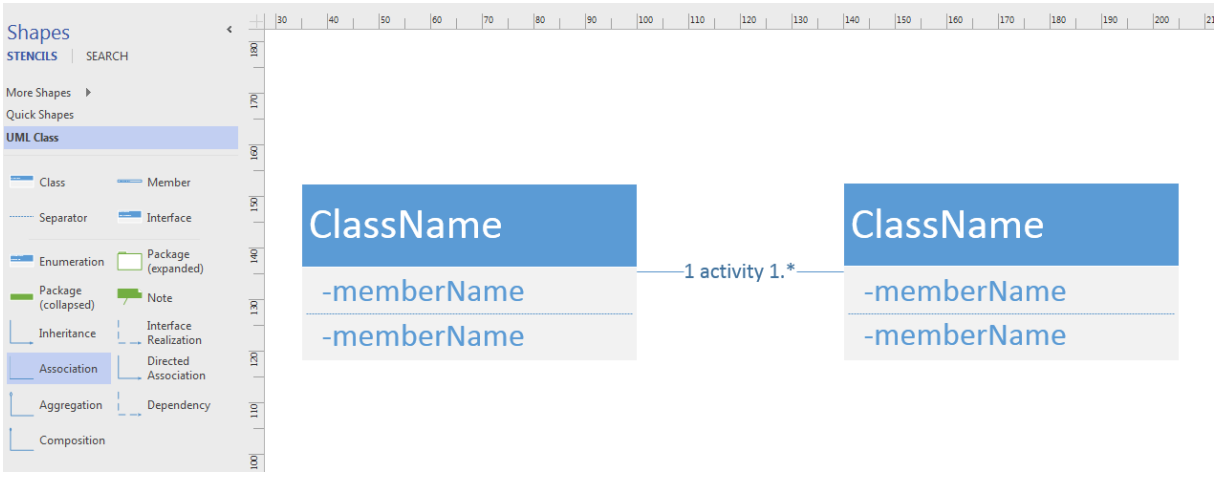


Şekil 5. Kavramsal sınıf diyagramında sınıf ve ilişki tanımı

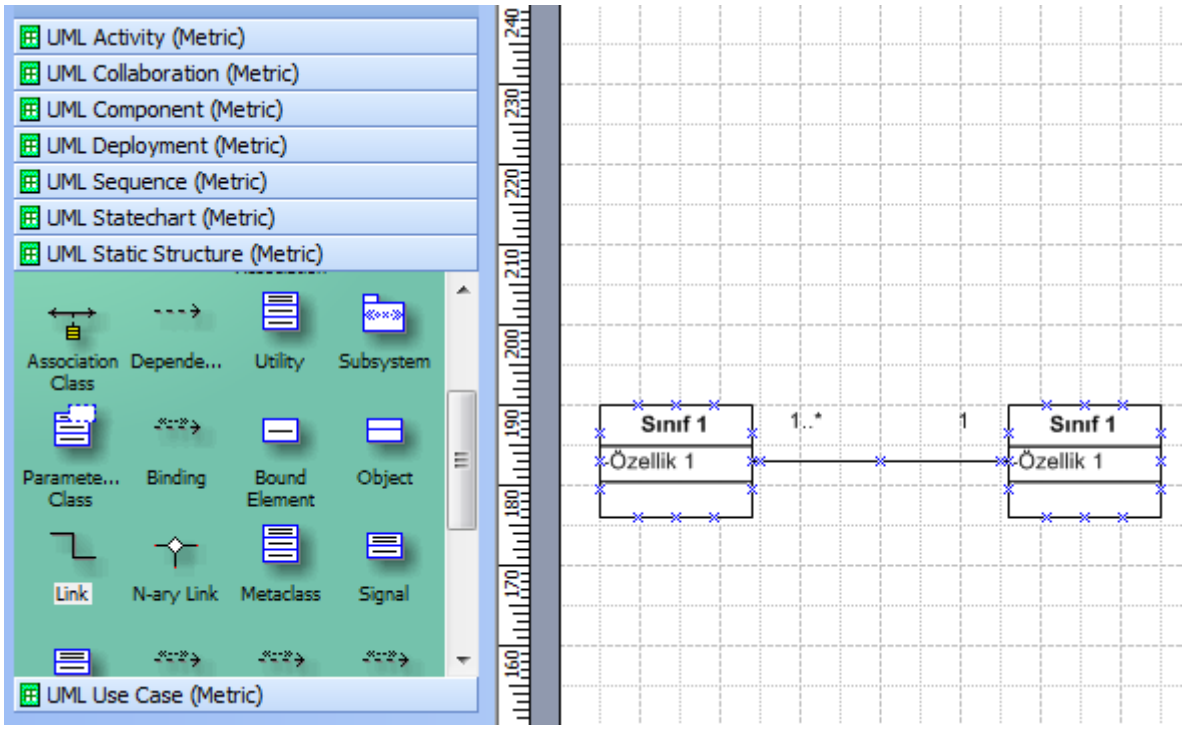
Sınıf Diyagramları Çizimi

Sınıf diyagramlarını MS Visio ile çizebiliriz. Kategoriler içerisinden “Software and Database” kategorisi seçilir ve “UML Class” diyagramı seçilir. “Metric Units” seçilip, “create” butonuna basılır. Soldaki şekilleri sürükle bırak ile diyagramı çizebiliriz. Elemanlara çift tıklayarak isimlendirme yapabiliriz. Sınıflar arası ilişkilendirme "Association" oku ile yapılmaktadır. MS Visio için, oka çift tıklayarak aradaki ilişki türünü okun üzerine yazabiliriz.

MS Visio 2003 te "Software" içerisinde "UML Model Diagram (Metric)" seçilir. Açılan pencerede sol tarafta UML diyagramlarının ismi bulunmaktadır. "UML Static Structure" sekmesinin altındaki elemanlar ile sınıf diyagramını çizebiliriz. "Class" ile sınıf ve nitelik tanımı, "Link" elemanı ile sınıflar arası bağlantı yapılır. Sürükleyip, çizime bıraktığımız "class" elemanına çift tıkladığımızda açılan pencereden sınıf isimlendirmeleri, açılan pencerede "attributes" seçeneğinden "new" butonuna basarak sınıf özelliği tanımlaması yapılır. "Link" elemanı ile sınıfları bağladıktan sonra, "link" elemanının üstüne çift tıkladığımızda açılan pencerede "End1", "End2" yazan kısımları değiştirip, ilişki türü ("1","1..*" gibi) belirtmesi yapabiliriz.

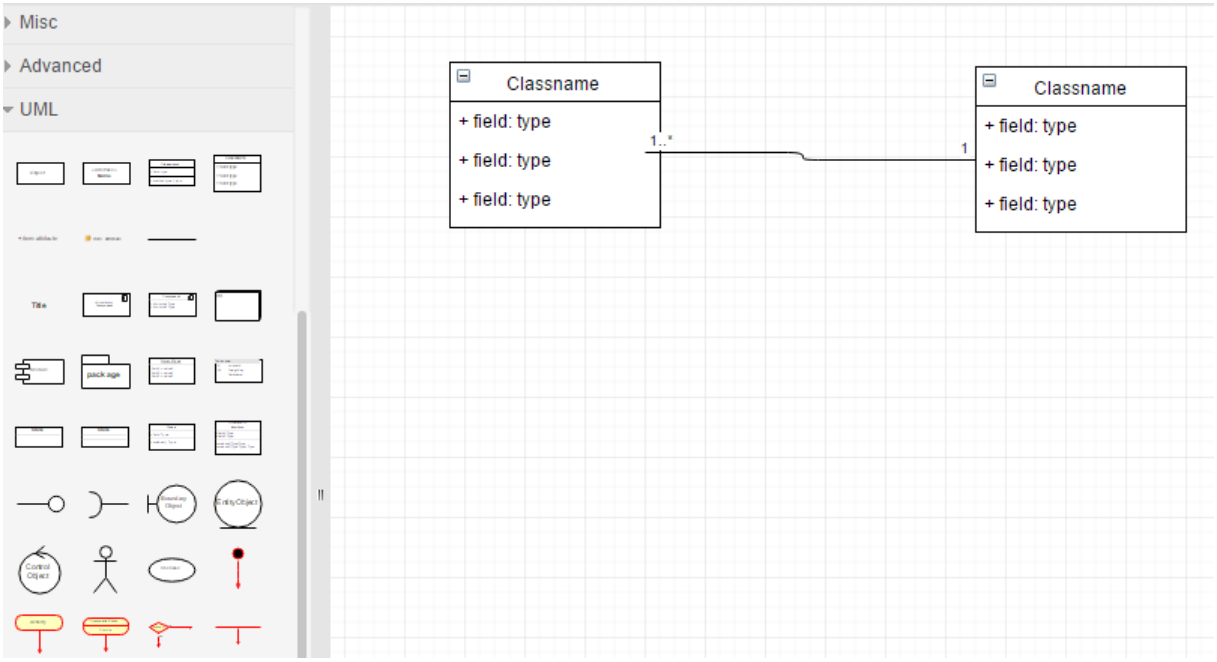


Şekil 6.1. MS Visio 2016 sınıf diyagramı çizimi



Şekil 6.2. MS Visio 2003 ile Sınıf diyagramı çizimi

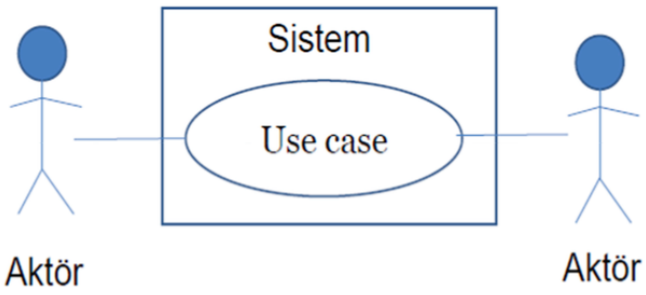
"www.draw.io" adresindeki çizim araçları ile de UML diyagram çizebiliriz. UML sekmesi altında "Class" veya Class 2" elemanı ile sınıf ve nitelik tanımlaması yapılmaktadır. "Association 1" elemanı ile sınıflar birbirine bağlanıp, ilişki türleri belirtilebilmektedir.



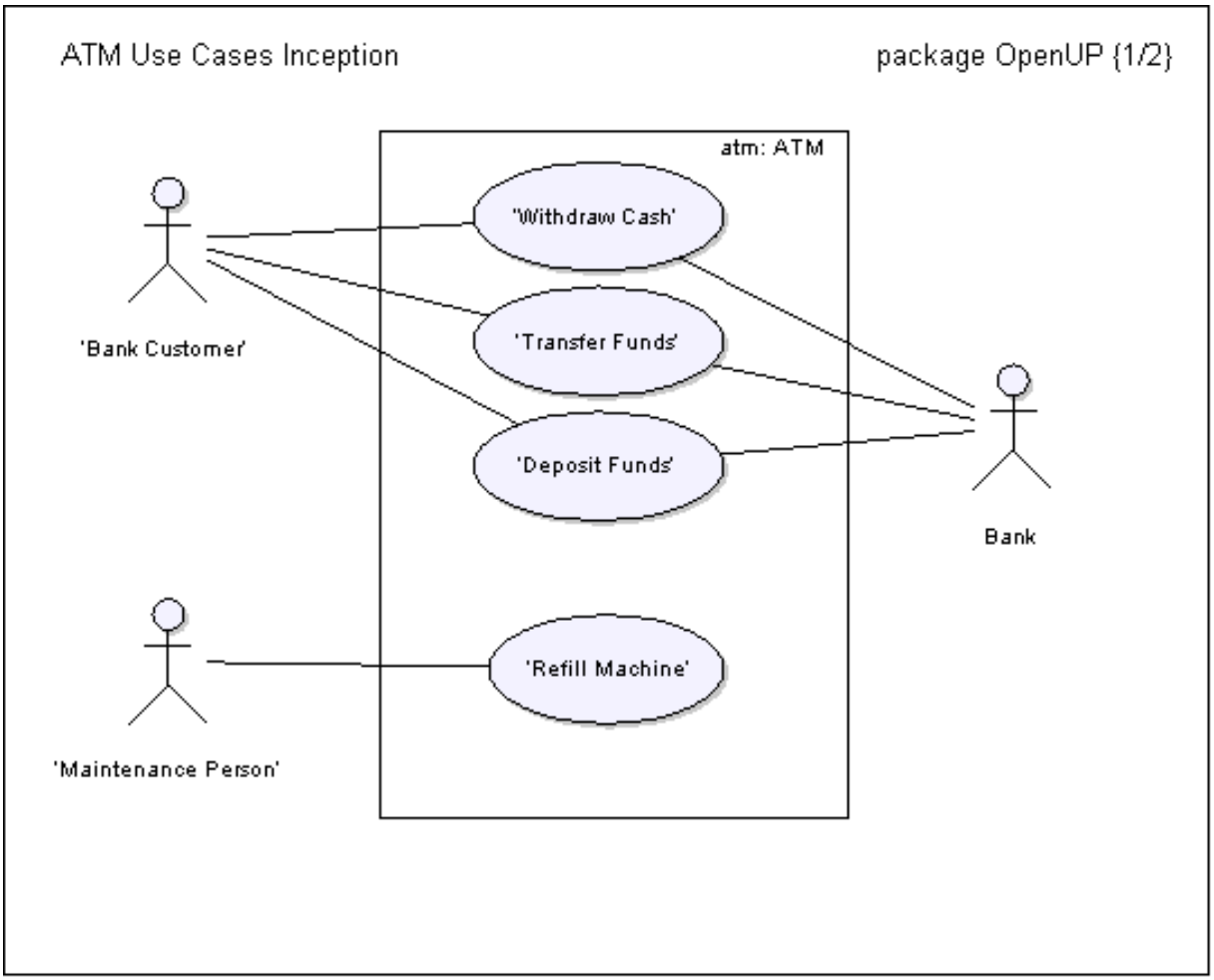
Şekil 6.3. www.draw.io ile UML Class Diyagram Çizimi

Use Case Diagram

Kullanım senaryosu diyagramı (use case diagram), sistem içerisindeki aktörleri ve aktörlerin etkileşimde olduğu olayları gösteren diyagramdır. Kullanım senaryosu diyagramı bir davranış diyagramıdır ve sistemdeki aktörlerden beklenen davranışı modeller. En basit hali ile şekilde gösterildiği üzere aktörler ve davranışlar çizilir.



Şekil 7. 1 Kullanım senaryosu temel elemanları

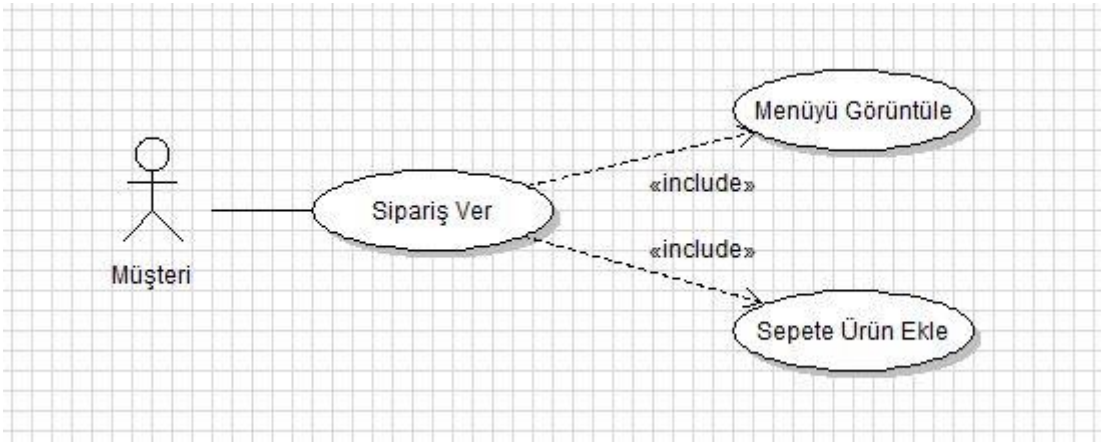


Şekil 7. 2 UML Kullanım diyagramı örneği

Case (senaryo) 'lar arası içerme (include, uses) ve genişletme (extend) ilişkisi bulunabilir.

İçerme (include)

Bazı durumlarda bir use case'in tamamlanabilmesi için, başka use case'lerin işin içerisine katılması zorunludur. Yani söz konusu use case, bir takım başka use case'ler olmaksızın tamamlanamaz. Örnek vermek gerekirse, müşteri, yemeksepeti.com sitesi üzerinden sipariş verebilmek için önce sipariş vermek istediği restoranı seçip, menüyü görüntülemeli ve daha sonra sipariş etmek istediği ürünü / ürünleri alışveriş sepetine eklemelidir. Yani, "Sipariş Ver" isimli use case'in tamamlanabilmesi, "Menüyü Görüntüle" ve "Sepete Ürün Ekle" isimli use case'lerin çalıştırılmasına bağlıdır. Include olarak adlandırdığımız bu ilişki türü, use case diyagramlarında aşağıdaki resimde aktarıldığı şekilde ifade edilmektedir. Burada okların yönü, hangi use case'in tamamlanabilmek için diğer hangi use case'lere bağımlı olduğunu gösterir.



Genişletme (extend)

Use case'ler arasındaki diğer bir ilişki türü extend olarak adlandırdığımız ilişki türüdür. Bu ilişki türü, iki use case arasında opsiyonel olarak var olabilecek bir durumu ifade etmek için kullanılır. Biz bu iki use case'i "base use case" ve "extending use case" olarak adlandıracağız.

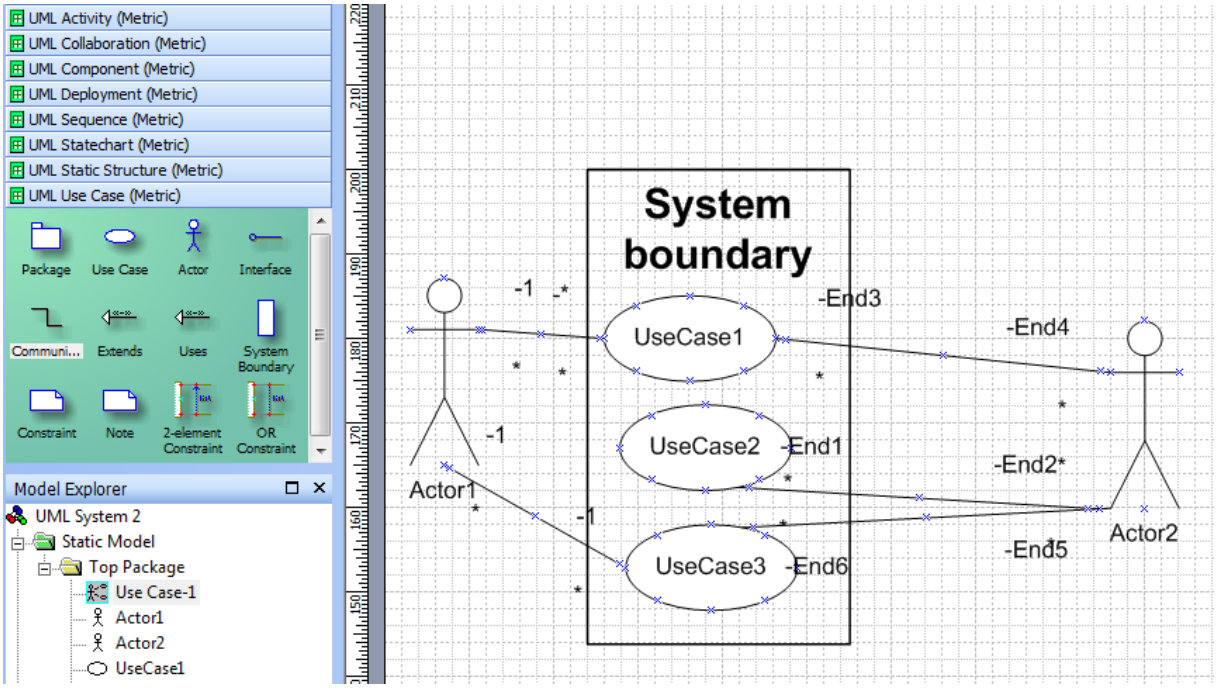
Bu ilişki, bir use case'in kapsamının başka bir use case tarafından genişletilebilmesinin olası olduğu durumlar için geçerlidir. Elimizde bir use case ve bu use case'in kapsamını oluşturan bir dizi adım olduğunu varsayarsak; sürecin bazı noktalarında, başka bir use case'in kapsamını oluşturan adımların kullanılması suretiyle, use case'in adımlarına yeni adımlar eklenmesi ve kapsamının genişletilmesi mümkün olabilir.

Bu durumu bir örnekle anlatalım. Farzedin ki, ATM kullanarak kredi borcunuzu ödüyorsunuz. Bu use case'in ismi "Kredi Ödemesi Yap" olsun. Ödemenizi yaptıktan sonra ATM size makbuz isteyip istemediğinizi soracaktır. İşlemi tamamlamak için makbuz almak gibi bir zorunluluğunuz yoktur ama ATM size böyle bir seçenek sunar. Bu örnekte, "Kredi Ödemesi Yap" base use case'imiz olurken; "Makbuz Yazdır" extending use case'imiz olacaktır ve iki use case arasındaki ilişki bir extend ilişkisidir.

Diyagram üzerindeki okun yönü bize use case'ler arasındaki bağımlılığın yönünü gösterir. Extending use case'in, base use case'e bağlı olduğu unutulmamalıdır. Yani bir extending use case, base use case olmadan tek başına bir anlam ifade etmez. Kredi borcu yatırmadan, borcunuzu yatırdığınıza ilişkin makbuz alamazsınız. Dolayısı ile extend okunun yönü türetilen senaryo'dan ana senaryoya doğrudur.

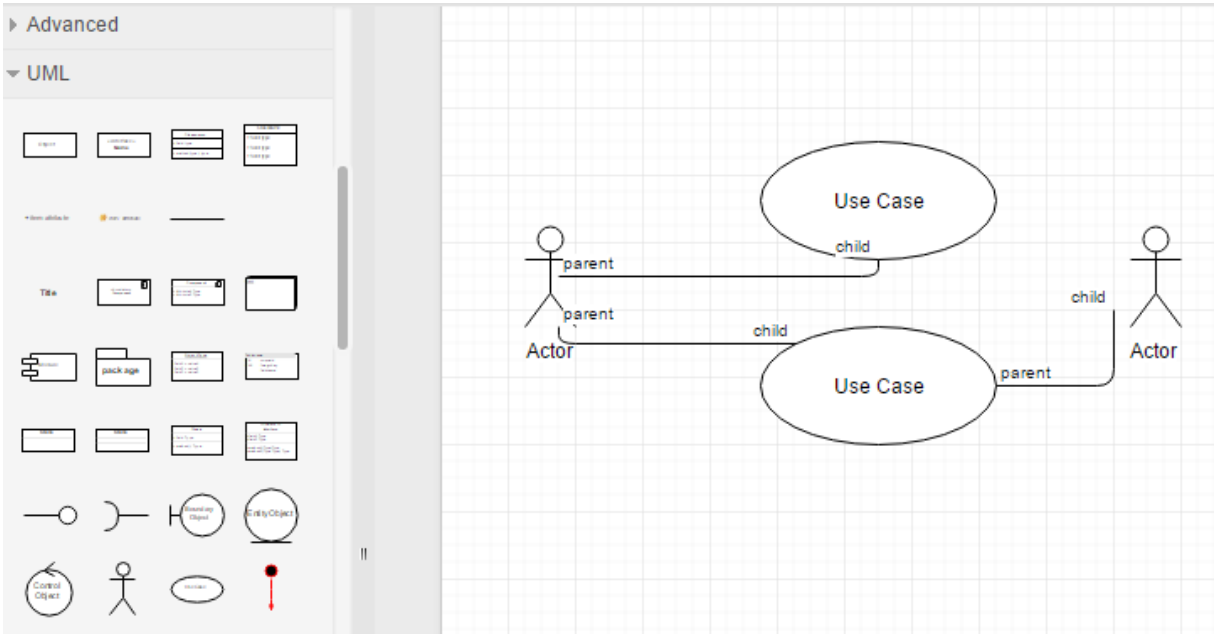
Kullanım senaryosu diyagramı çizimi

MS Visio 2003 için "Software" kategorisi altında "UML Model Diagram(metric)" seçilir. Açılan projede soldaki şekiller kısmında "UML Use case(metric)" seçeneğinin altındaki öğeler ile diyagramı çizebiliriz. Elemanların üstüne çift tıklayıp, açılan pencereden isimlendirme yapılabilmektedir. "Aktor" elemanı ile aktörler; "system boundary" ile davranışları kapsayan sistem; "use case" elemanı" ile kullanım senaryoları belirtilir. "connection" oku ile kullanıcıları, kullanım senaryolarına bağlayabiliriz. Oka çift tıklayıp "end1,end2" ile gösterilen değerleri değiştirerek ilişki türü tanımlanabilir.



www.draw.io adresinde UML klasörü altındaki kullanım diyagramına ait olan elemanlar ile kullanım senaryosu çizilmektedir; ancak sistem sınırı belirten eleman bulunmamaktadır. Association ile aktörler senaryolara bağlanır, uses(include) ve extend ile senaryolar arası içerme/genişletme belirtilebilir.

Şekil 7. 3 MS Visio 2003 kullanım senaryosu diyagramı çizimi



Şekil 7. 4 Draw.io adresinden kullanım diyagramı çizimi