6.3. Yapısal Şemalar

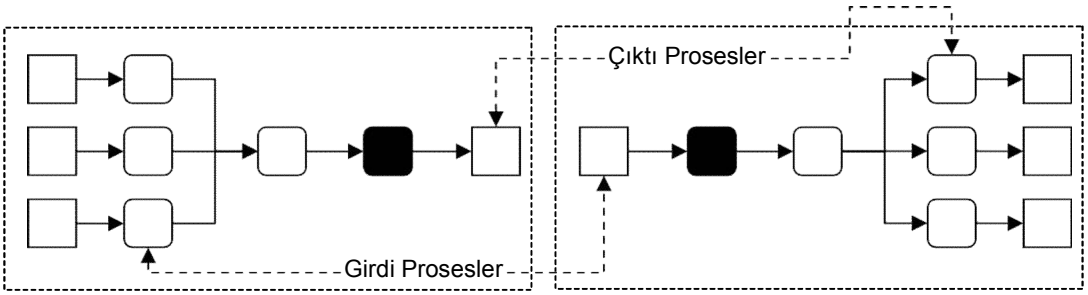
Yapısal şemalar (YŞ), hiyerarşik bir yapı içerisinde program modüllerinin ve bu modüllerin diğerleriyle ilişkisinin grafiksel gösterimidir. Bir yapısal şemanın en üstünde (root), tek bir yönetici modül bulunur.

Bir sonraki seviyede ise, modül çağrılmalarını koordine eden diğer modüller bulunmaktadır. En alt seviyedeki modüller, herhangi bir başka modül çağırmazlar, sadece belirli görevleri yaparlar.

VAD sistemin NE yaptığını gösteren bir analiz aracıdır, YŞ ise sistemin bilgisayar programcılarınca NASIL yürütüleceğini gösteren bir tasarım aracıdır.

Yapısal şemalarda iki teknik kullanılır: [1] Dönüştürme (Transform) Analizi ve [2] İşleme (Transaction) Analizi. Bilgi sistemleri genelde ya dönüştürme merkezli ya da işleme merkezlidir. Yapısal şemalar çizilirken önce sistemin dönüştürme ya da işleme merkezi tespit edilmelidir.

İşleme merkezli sistemde, veri sistemin merkez modülüne (işleme merkezi) gelir, değerlendirilir ve oradan uygun lokasyonlara dağıtılır. Dönüştürme merkezli sistem yeni değerler üreten bir merkezi dönüştürme modülüne sahip olmakla beraber, bu sistemlerin girdileri fazla çıktıları azdır. Bu iki tip sistem şekil 6.3'de gösterilmiştir.



Şekil 6.6 – Dönüştürme ve İşleme Merkezli Sistemler

6.4. Yapısal Dil

Çoğu durumlarda, bilgi sistemi tasarımı için kullanılan akış şemaları, karar tabloları ve HIPO gibi araçlardan gerçek programlara geçmek oldukça zor olabilir.

YD, VAD'da bulunan proseslerdeki dönüşüm işlemlerinin nasıl yapılacağını tarif etmek için kullanılır. YD, bir nevi normal konuşma dilini kullanarak bilgisayar programları yazmaya benzer. YD, Sahte Kod (SK) (Pseudocode) olarak da bilinir.

Bu iki kavram arasında temelde bir fark olmamakla beraber YD'nin konuşma diline, SK'nın ise programlama diline daha yakın olduğu düşünülebilir.
Aşağıda bir YD örneği verilmiştir:

Örnek:
Firmada Ayda brüt 250 dolardan fazla kazananların listesi

1. PRINT Rapor Başlığı

2. READ Her bir Personel Verisi

3. Brüt Ödemeyi Hesapla

4. Brüt Ödeme 250 Dolar'dan Fazla mı?

    a. Evet ise, PRINT Numara, Oran, Brüt Ödeme

    b. Hayır ise, Hiçbir şey Yazma

5. Tüm personel için 2-4 adımları tekrarla

Yapısal dil normal olarak READ, WRITE, SORT, MOVE, MERGE, SUBTRACT, MULTIPLY, DIVIDE, DO, FINE gibi İngilizce fiilleri kullanır. Ayrıca değişkenler için MUSTERI_ADI, MUSTERI_ADRESI gibi etiketler kullanılır. Sıfat ve zarf kullanılmaz ve kullanılan isimler *veri sözlüğünden (ileride anlatılacak)* alınırlar.

.

Bilgisayar programlarındaki tipik 3 prosesin gösterilmesinde YD kullanılabilir.

Bunlar:

[1] Sıra (sequence),
[2] şartlı ifadeler (conditional statements),
[3] tekrar (repetition).

Sıra, programlamada özel bir yapı gerektirmeyen sıralı işlemleri ifade eder;

İşlem 1...
İşlem 2...
İşlem 3... gibi.

Şartlı ifadeler ise herhangi bir mantıksal ya da matematiksel ifadenin farklı durumlarında ne yapılması gerektiğini gösterir.

Örneğin;

```
IF şart A doğruysa
        İşlem A'yı yap
ELSE İşlem B'yi yap
END IF
```

Veya

```
IF Stoktaki_Miktar, Min_Stok_Miktarı ndan küçükse
        THEN Yeni sipariş üret
ELSE Bir şey yapma
END IF
```

Şartlı ifadelerin bir diğer şekli de programın izleyebileceği birçok farklı yol olduğu zamanlarda kullanılır.

Bunun için CASE ifadesi kullanılır, şöyle ki;

```
SELECT CASE
CASE 1 (Şart 1)
        Şart 1 için yapılacak işlemler
CASE 2 (Şart 2)
        Şart 2 için yapılacak işlemler
:
:
CASE n (Şart n)
        Şart n için yapılacak işlemler
END CASE
```

Bir örnek verecek olursak;

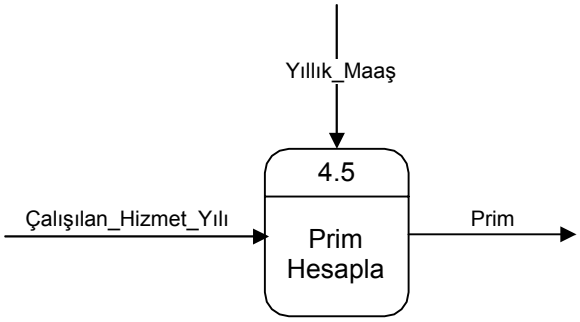
```
READ Stoktaki_Miktar
        SELECT CASE Stoktaki_Miktar
                CASE 1 (Stoktaki_Miktar, Min_Sipariş_Miktarı ndan büyük ise)
                        Hiçbirşey yapma
                CASE 2 (Stoktaki_Miktar, Min_Sipariş_Miktarı na eşit ise)
                        İlgili kişiye e-posta gönder, durumu bildir
                CASE 3 (Stoktaki_Miktar, Min_Sipariş_Miktari ndan küçük ise)
                        Yeni Sipariş Üret
                CASE 4 (Stok Yoksa)
                        Acil olarak yeniden sipariş yöntemi çalışması başlat
        END CASE
```

Tekrarlı işlemlerde ise DO-UNTIL ya da DO-WHILE benzeri ifadelerle döngüler oluşturulur.

Örneğin;

```
DO
        READ Stok Kayıtları
        IF Stoktaki_Miktar, Min_Sipariş_Miktari ndan küçük ise
                THEN Yeni sipariş üret
        ELSE Hiçbir şey yapma
        END IF
UNTIL EOF (Dosyada Kayıt Kalmayana Kadar)
```

Aşağıda bir VAD içerisinde bulunan bir proses için hazırlanmış bir YD örneği verilmiştir:




```
IF Çalışılan_Hizmet_Yılı 25 ya da daha büyükse THEN
        Yıllık_Maaş la 0.05 i Çarparak Prim i Hesapla
ELSE (Çalışılan_Hizmet_Yılı 25 den azdır)
        Yıllık Maaş la 0.025 i Çarparak Prim i Hesapla
END IF
```

6.5. Karar Tabloları

Eğer proses mantığında birçok şartlı durum söz konusuysa ve her bir durum farklı işlemlerin yapılmasını gerektiriyorsa, o zaman karmaşık mantığın anlaşılması, YD ile oldukça zordur. Yapılan araştırmalar, çok sayıda iç içe yuvalanmış IF ifadelerinin yorumlanmasında, insanları bunları birbirine karıştırdıklarını göstermiştir. Proseslerin böyle karmaşık olduğu durumlarda Karar Tabloları durumu YD'den daha iyi ifade edebilir.

Karar tablosu, sistemin mantığını adım adım yerine tablo biçiminde gösteren bir araçtır. Akış şemalarının alternatifi olabileceği gibi birlikte de kullanılabilirler.
Karar tabloları **4** ana bölümden oluşur:

1. Tablonun sol üst bölümüne mümkün olan tüm şartlar yazılır.
2. Sol alt bölümde, şartların kombinasyonları sonucu yapılan tüm mümkün faaliyetler listelenir.
3. Kurallar, sağ üst bölümde oluşturulur. Şartın durumuna bağlı olarak şartın karşılığındaki olasılıkları ifade eden gösterimler kullanılır. Örneğin; Evet, Hayır'ı temsil için E ve H harfleri kullanılır.
4. Sağ alt bölümde ise, verilen bir kural için geçerli olan faaliyetlerin gösterilmesini sağlayan ve onay anlamında olan "X" işaretleri yerleştirilir.

Karar tablosundaki geçerli teorik kural sayısı 2^n 'dir.
Burada n şart sayısıdır.
Her şartın karşılığında sadece 2 olasılık olduğu durumlarda geçerli olan bu formüle göre eğer 3 şartımız varsa toplam kural sayımız $2^3 = 8$ olur.

Tablo 6.1'de bir karar tablosu örneği verilmiştir. Bu teorik olarak tüm kuralları içeren bir karar tablosudur.

İndirim Hesapla	**KURALLAR**																															
ŞARTLAR	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
Satın alma<100$	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
Özel Teklif	E	E	E	E	E	E	E	E	H	H	H	H	H	H	H	H	E	E	E	E	E	E	E	E	H	H	H	H	H	H	H	H
İndirim<2$	E	E	E	E	H	H	H	H	E	E	E	E	H	H	H	H	E	E	E	E	H	H	H	H	E	E	E	E	H	H	H	H
İndirim Sonrası>45$	E	E	H	H	E	E	H	H	E	E	H	H	E	E	H	H	E	E	H	H	E	E	H	H	E	E	H	H	E	E	H	H
7 Gün içinde ödeme	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H
FAALİYETLER																																
2$ indirim	X	X	X	X																												
%5 indirim									X	X	X	X	X	X	X	X																
%7.5 indirim					X	X	X	X																								
%8 indirim																	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Ekstra %1 indirim				X				X				X					X		X		X		X		X		X		X		X	

Tablo 6.1 – İndirim Hesaplama Prosesi İçin Karar Tablosu

Tablo 6.2'de ise aynı sisteme ait sadece pratikte mümkün olan kuralları içeren bir karar tablosu verilmiştir. Bu indirgenmiş karar tablosudur.

İndirim Hesapla	**KURALLAR**								
ŞARTLAR	1	2	3	4	5	6	7	8	9
Satın alma<100$	E	E	E	E	E	E	E	H	H
Özel Teklif	E	E	E	E	H	H	H	-	-
İndirim<2$	E	H	H	H	-	-	-	-	-
İndirim Sonrası>45$	-	E	-	H	-	E	H	-	-
7 Gün içinde ödeme	-	E	H	-	H	E	E	E	H
FAALİYETLER									
2$ indirim	X								
%5 indirim					X	X	X		
%7.5 indirim		X	X	X					
%8 indirim								X	X
Ekstra %1 indirim		X				X		X	

Tablo 6.2 – İndirgenmiş Karar Tablosu

Karar tablosunda bazı şartların birden fazla karşılığı olabilir örneğin;

	KURALLAR					
ŞARTLAR	1	2	3	4	5	6
Çalışan Tipi	M	S	S	A	S	A
Çalışılan saatler	<40	<40	40	40	>40	>40
FAALİYETLER						
Taban ücreti öde	X		X		X	
Saatlik ücret hesapla		X		X		X
Fazla mesai ücreti hesapla					X	X
Devamsızlık raporu üret		X				

M: Maaşlı S: Saat ücretli

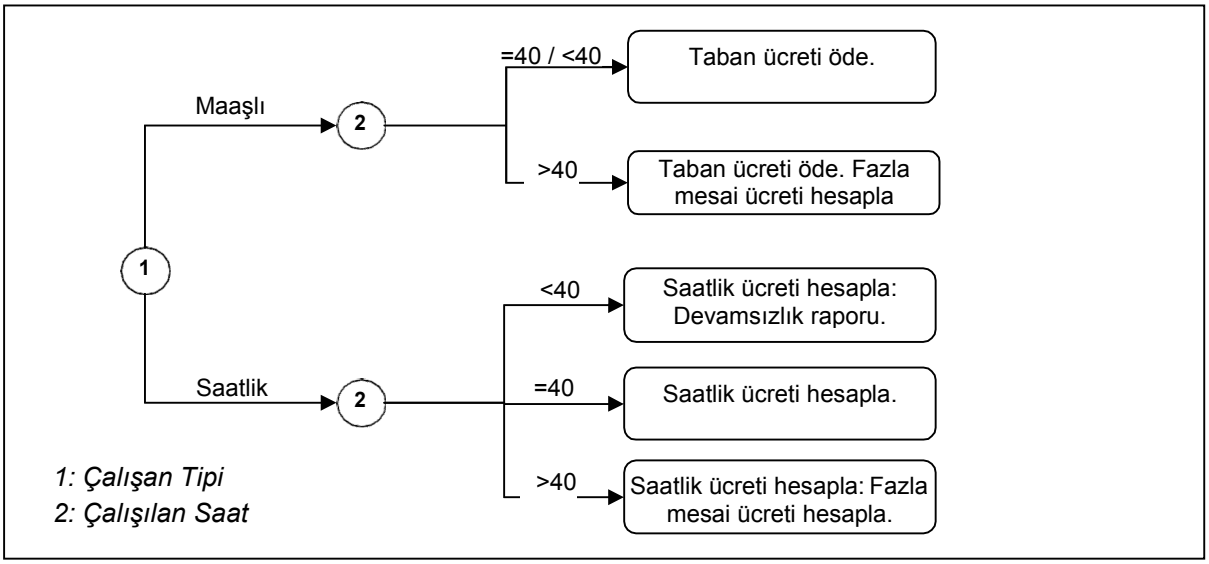
Tablo 6.3 – Şartların ikiden fazla olasılık taşıdığı durum için örnek (Karar Tablosu)

6.6. Karar Ağaçları

Karar ağaçları da karar tabloları gibi çok sayıda şartlı ifadelere sahip proseslerin tanımlanmasında kullanılabilir. Karar ağaçları, karar tabloları için bir alternatif araçtır.

Programların daha etkin yazımında karar tablolarından ziyade karar ağaçlarının daha faydalı olduğunu gösteren araştırmalar vardır. Bunun nedeni grafiksel gösterim ve adımların akışının görülebilmesidir.

Bu karar ağaçları yönetim biliminde kullanılan karar ağaçlarından farklı olarak olasılık içermezler. Çünkü sistem analizindeki karar ağaçlarının amacı karar prosesindeki şartlı durumları ve faaliyetleri tanımlamak ve organize etmektir.
Bir karar ağacı örneği şekil 6.4'te verilmiştir.



Şekil 6.7 – Karar ağacı örneği

6.7. HIPO

Bilgi sistemi geliştirme araçlarından bir diğeri de IBM tarafından büyük ve karmaşık çalışma sistemleri için geliştirilmiş olan HIPO (Hierarchy Plus Input-Processing- Output) tekniğidir. “Nasıl” dan ziyade “Ne” yapılacağı üzerinde yoğunlaştığı için akış şemalarından farklıdırlar.

HIPO’nun **3** temel amacı vardır:

1. Sistem fonksiyonlarının parçalara ayrılmış hiyerarşik yapısını göstermek

2. Sistem fonksiyonlarının ayrıntılarını herhangi bir programlama diline bağlı kalmaksızın göstermek.

3. Sistem fonksiyonları düzeyinde, girdiler ve çıktıları görsel olarak tarif etmek.

HIPO, iki ayrı diyagramdan oluşur:

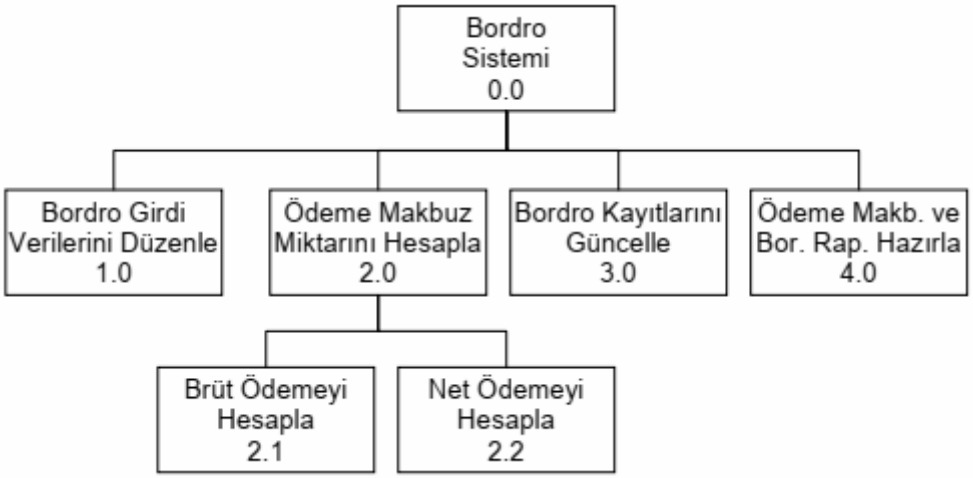
a. <u>Görsel İçerik Tablosu:</u>

Hiyerarşi diyagramı olarak da bilinir. İngilizce kısaca VTOC (Visual Table Of Contents) olarak ifade edilir. Sistemi yukarıdan aşağıya hiyerarşik bir yapıda modüler olarak ifade eden bir şemadır.

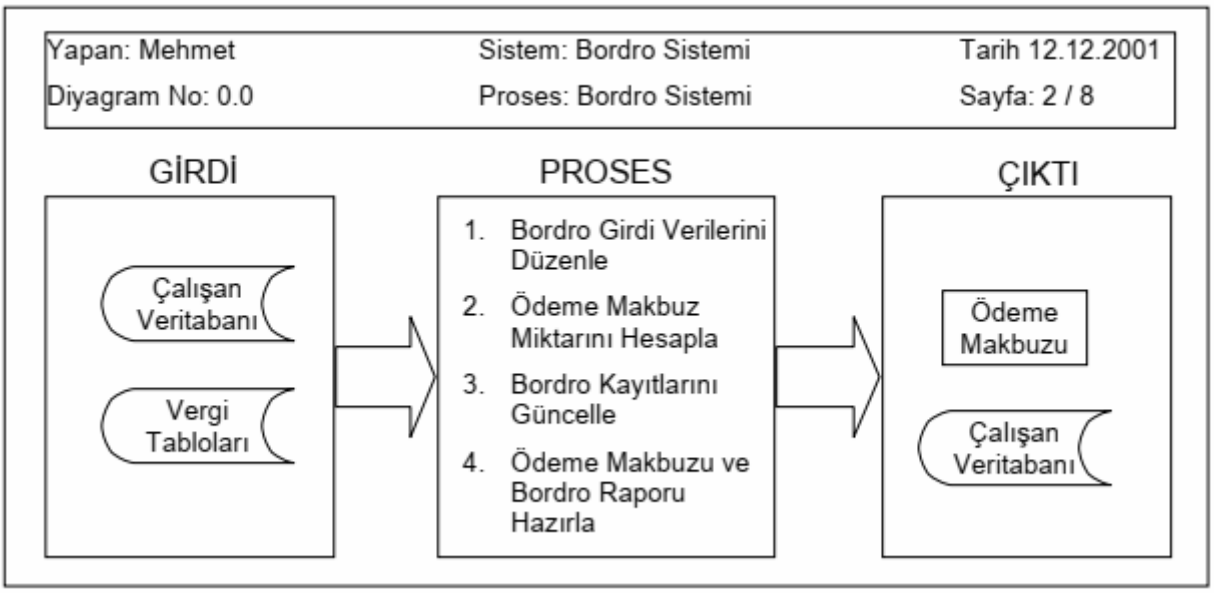
b. <u>HIPO Özet Diyagramı:</u>

İngilizce HIPO Overview Diagram olarak bilinir. VTOC'taki her bir kutu (modül) için girdi, çıktı ve ana prosesleri gösterir.

Şekil 6.8 ve 6.9'da bir bordro sistemine ait VTOC ve HIPO özet diyagramı örneği verilmiştir.



Şekil 6.8 – Bordro Sistemi için HIPO Hiyerarşi Diyagramı



Şekil 6.9 – HIPO Özet Diyagramı

KAYNAKÇA:

1. Sistem Analizi (Doç. Dr. Haluk Erkut – Kıyı Yayınları 1989)

2. İşletme Yönetiminde Sistem Yaklaşımı (Prof.Dr. H.Öner Esen – Alfa Basım Yayın Dağıtım 1998)

3. Yönetim Bilgi Sistemleri (Doç. Dr. Hadi Gökçen – EPİ Yayıncılık 2002)