

MIS (YÖNETİM BİLİŞİM SİSTEMLERİ)

SORU: Sistem (Düzenek, Nizam) hakkında bilgi veriniz.

Bu yazının amacı, en basit terimlerden birisi olan Sistem kelimesini açıklamaktır. Bir kelime ne kadar basit ve ne kadar sık kullanılıyorsa açıklaması da bir o kadar zor olur.

Kelime esas olarak İngilizcedeki System kelimesinden Türkçeye Sistem olarak girmiştir (veya diğer başka bir dilden de girmiş olabilir ancak Türkçe olmadığı kesin). Kelimenin girmesinden önce Türkçede aynı anlamı ifade eden “nizam” kelimesi kullanılmaktaydı. Ardından, sanırım sistem kelimesinin yabancı dilden girmesi ile birlikte “nizam” kelimesini de eski bulan birileri tarafından “düzen” veya “düzenek” gibi “düzmek” kökünden kelimeler üretilerek kullanılmaya başlandı. Ancak günümüzde kelime neredeyse tamamen Türkçeleşmiş ve dilin bir parçası haline gelmiştir denilebilir. Yine de “nizam” kelimesini çok da yabana atmamak gerek çünkü hala yoğun bir şekilde kullanılmakta.

Bu yazının amacı kelimeyi elbette bilgisayar bilimleri açısından incelemektir. Farklı alanlarda farklı anlamlara gelebilmekle birlikte genelde aynı özde buluşuyor. Ortak bir tanım yapılması gerekirse sanırım aşağıdaki tanım doğru olur:

1. Alt bileşenleri olan
2. Bileşenler arasında ilişkileri olan
3. Girdileri ve çıktıları olan
4. Bir sınırı ve ortamı olan

kümedir (set) demek doğru olur. Yani her sistemin var olabilmesi için bazı alt bileşenlerinin veya elamanlarının bulunması gerekir, bu eleman veya alt bileşenler de birer sistem olabileceği gibi böyle bir şart yoktur. Örneğin duruma göre basit bir sayı, basit bir bilgi veya herhangi bir araç olabilir. Ardından bu bileşenler arasında bazı ilişkilerin bulunması beklenir. Aslında sistemin özü bu bileşenlerin birbir ile nasıl çalıştığıdır. Elbette sistemin varlığının sebeplerinden birisi de girdileri ve çıktılarıdır. Ancak bunlar şart değildir. Yani hiç girdisi yahut hiç çıktısı olmayan sistemler olabilir. Son olarak her sistemin çalıştığı bir ortam ve bu ortamların ilişkisini belirleyici bir sınır bulunmalıdır. Diğer bir deyişle sistemler ortamlardan etkilenir ve her sistem her ortamda çalışmak zorunda değildir.

Yukarıdakiler ışığında sistem tanımını biraz daha ilerleterek, bir sistemin taşınması gereken 9 özelliği aşağıdaki şekilde listeleyebiliriz:

1. Bileşenler (Components)
2. İlişkili bileşenler (Interrelated components)
3. Sınır (boundary)
4. Amaç (purpose)

5. Ortam (environment)
6. Arayüzler (Interfaces)
7. Kısıtlar (Constraints)
8. Girdiler (Inputs)
9. Çıktılar (Outputs)

Yukarıdaki yaklaşımı biraz daha açarsak bir sistem üzerinde yapılabilecek 4 farklı işlem ortaya çıkar bunlar bilişim sistemlerinin yönetimi için (management of information systems , MIS) oldukça önemlidir.

1. Ayırıştırma (Decomposition)
2. Modülerlik (Modularity)
3. Birliktelik (Coupling)
4. Yapışkanlık (Cohesion)

Bu işlemleri bir örnek üzerinden açıklamaya çalışalım. Örneğimizi üniversite üzerinden verelim. Örneğin üniversiteyi bir sistem olarak düşünürsek, bu sistemin yönetilebilir alt parçalara ayrılmasına ayırıştırma (decomposition) ismini verebiliriz. Diyelim ki üniversitenin fakülte ve bölümlere ayrıştırılması ve alt birimlerinin yönetimini ayırıştırma olarak düşünebiliriz. Buradaki kritik nokta, bir sistemi alt sistemlere bölerken aşağıdaki hususlara dikkat edilmesidir:

1. Alt sistemlerin yönetilebilir ve anlaşılabilir olması gerekir. Örneğin üniversite sistemini alt sistemlerine bölüp sonrada ismi Ahmet olan bir alt sistem çıkarmak hata olacaktır.
2. Alt sistemler bölünürken, her alt sistem kendi içerisinde özel olarak ele alınmalı ve diğer sistemlerden bağımsız düşünülmelidir. Örneğin üniversiteyi fakültelere bölerken birbiri ile ilişkili eğitim ve araştırma içeriklerinin beraber bulunmasını düşünmek doğaldır. Her alt sistemin kendi eğitim ve araştırma içeriğini belirleyeceği buradan çıkarılabilir. Üniversiteyi alt fakültelere böldüğümüz bir senaryoda “eğitim dekanlığı” gibi bir birim anlamsızlaşır.
3. Sistemdeki kişiler düşünülerek ayırıştırma yapılmalıdır. Örneğin öğrencilerin bölünebildiği en alt grubun bölümler olduğu üniversitemizde bir fakülte oluştururken bölüm bu fakültelerden birisine bağlı olmalıdır. Bir bölümü birden fazla fakülteye bağlamak bir hata olacaktır.
4. Son olarak ayırıştırma işlemi sırasında farklı görüşleri gündeme alabilmek için farklı zamanlarda ve farklı kişilerle ayırıştırma alternatiflerinin tartışılmasının faydalı olduğu görülmüştür.

Modülerlik ise sistemin ayrıştırılmasının doğal bir sonucudur. Modülerlik çok farklı anlamlarda sonucu etkileyebilir. Örneğin ekleme çıkarma olarak düşünecek olursak, ayrıştırılma sonucunda elde edilen alt sistemlerden herhangi birisinin sistemden bağının koparılması veya yeni bir modülün sisteme eklenmesinin kolaylaştırılmasıdır. Veya yönetim açısından düşünülecek olursa, modülerlik her modülün diğer sistemlerden bağımsız olarak kendini yönetebilmesi olarak ortaya çıkacaktır.

Birliktelik (coupling) kavramı ise, sistemdeki bileşenlerin birbirine ihtiyacını ifade etmektedir. Örneğin sistemdeki bir parçada sorun yaşanması halinde, sistemdeki diğer bileşenlerin bir kısmı veya tamamında da sorun ortaya çıkabilir. Burada iki yeni kavram ortaya çıkar, yüksek birliktelik (high coupling) ve düşük birliktelik (low coupling). Anlamsal olarak yüksek birliktelik sistemdeki bir bileşenin diğer bileşenlere daha fazla bağlı olduğunu ifade etmektedir. Örneğin rektörlük ve fakülteler arasındaki ilişki yüksek birliktelik olarak görülebilir, rektörlükteki bir problem dekanlıkları etkileyebilir veya tam tersi mümkündür ancak örneğin güzel sanatlar fakültesi ile mühendislik fakültesi arasındaki ilişkiyi daha düşük birliktelik olarak düşünmek yerinde olur. Birbirlerini etkilemeleri daha düşüktür denilebilir.

Son olarak yapışkanlık (cohesion) kavramı, sistemin parçalarının birbirine ne kadar yapışmış olduğunu yani birbirine ne kadar ihtiyaç duyduğunu temsil etmektedir. Bir sistemin herhangi bir parçasının değiştirilmesi veya güncellenmesi sırasında yapışık ilişkide bulunduğu diğer parçaların da etkilenmesi beklenir. Bu açıdan düşük yapışıklık (low cohesion) kavramı önemlidir ve sistem analizinde beklenen özelliklerdendir. Yapışkanlık kavramını daha iyi anlayabilmek için şöyle bir örnek verilebilir, örneğin bir yazılımdaki sisteme kayıt ve giriş (login) ekranını ele alalım. Bu ekran diğer bütün modüllerden bağımsız olarak tek başına çalışabiliyor ve diğer modüllerle sadece giriş işleminin sonucuna göre iletişim kuruyorsa burada düşük yapışkanlık vardır denilebilir. Buna mukabil, sistemdeki bütün modüllerin içerisine gömülmüş bir kod parçası şeklinde çalışıyorsa yüksek yapışkanlık vardır denilebilir. Örneğin kodda değişikliğe gidilip kullanıcılardan giriş sırasında doğum yıllarını da sormaya karar verdik, bu durumda tek bir noktadan müdahale etmek yeterli midir yoksa bütün ilgili giriş ekranlarına müdahale mi gerekmektedir sorusunun cevabı yapışkanlık seviyemizi belirlemektedir.