

Biçimsel (Mantıksal- Matematiksel) Modelleme

Gereksinim Çözümleme Çalışması

- Geliştirilecek bilgi sistemi ya da yazılımla ilgili olarak tüm gereksinimlerin araştırılması, tanımlanması, ortaya çıkarılması ve bir gösterim biçimi ile açıklanması (modellenmesi) çalışmaları Sistem Çözümleme Çalışması olarak adlandırılır.
- Temel olarak çözümleme çalışması:
 - Mevcut Sistemin İncelenmesi
 - Önerilen Sistemin Modellemesi

- Mevcut sistemin modellenmesinden sonra, bilgisayar ortamında işlerin yapılabilmesi amacıyla önerilecek sistemin modeli oluşturulur. Bu model, önerilen sistemin işlevsel yapısı, veri yapısı ve kullanıcı ara yüzünü içerir. Bu model aynı zamanda “*mantıksal model*” olarak da tanımlanır. Mantıksal model, önerilen sistemin veri yapısını ve süreç yapısını hem genel hem de ayrıntılı olarak tanımlar.

•

- Detaylı bir sistem spesifikasyonu oluşturmak üzere kullanılan modellerdir. Sistemin işlevsel yapısı, veri yapısı ve kullanıcı arayüzünü içerir.
- Hata ve tutarsızlıkların tespiti için kullanılabilir.
- Denetleyici belgelendirme gerektiren emniyet kritik sistemlerde kullanılması faydalıdır. Döküman sistemin güvenliği açısından faydalıdır.
- Microsoft tarafından kullanılan sürücü doğrulama sistemleri gibi statik doğrulamada kullanılan araçların temelini oluşturur.

- Sistem tasarımcıları ve programcılara yöneliktir.
- Önerilen sistemin veri yapısını ve süreç yapısını hem genel hem de ayrıntılı olarak tanımlanır.
- Kolaylıkla fiziksel modellere dönüştürülebilir (Veri tabanı tabloları vb.).

- Biçimsel modelleme ile iki tür hata önlenabilir;
- Tanımlama ve tasarım hataları ile ihmal etmeler.
- Bir spesifikasyon ve bir program arasındaki tutarsızlıklar.

- Biçimsel yöntemler için başlangıç noktası, sistemin spesifikasyonu gibi davranan matematiksel bir sistem modelidir.
- Biçimsel spesifikasyon, sistemin ne yapması gerektiğine dair net bir tanımdır. Sistemi tasarlamamanın en kısa yoludur. Yanlış anlama olasılığı azalır.
- Biçimsel spesifikasyon ortaya koymak, gereksinimlerin ayrıntılı analizinin yapılmasını sağlar.

Biçimsel Spesifikasyonun Avantajları

- Gereksinimler detaylı analiz edildiğinden, sorunlar erken tespit edilir ve sonraki geliştirme maliyeti azalır.
- Semantik bir dille ifade edilen spesifikasyonlarda tutarsızlık ve eksikliklerin bulunabilmesi için otomatik olarak analiz edilir.
- Bir dizi doğruluğu koruyan dönüşümler yoluyla programa dönüştürülebilir. Böylece ortaya çıkan program kendi gereksinimlerinin karşılayabilir.
- Program test masrafları azalabilir.

Biçimsel Spesifikasyonun Dezavantajları

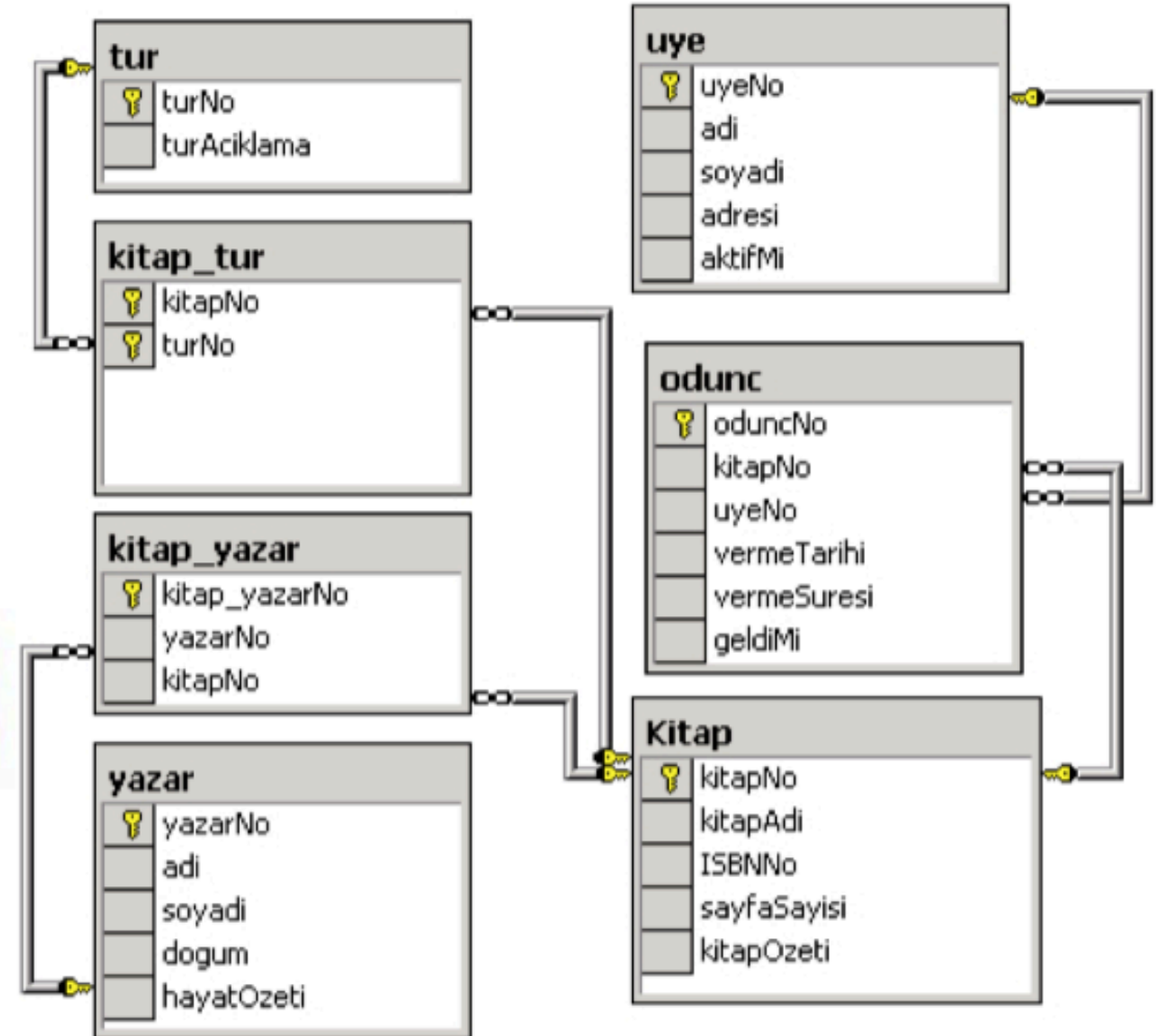
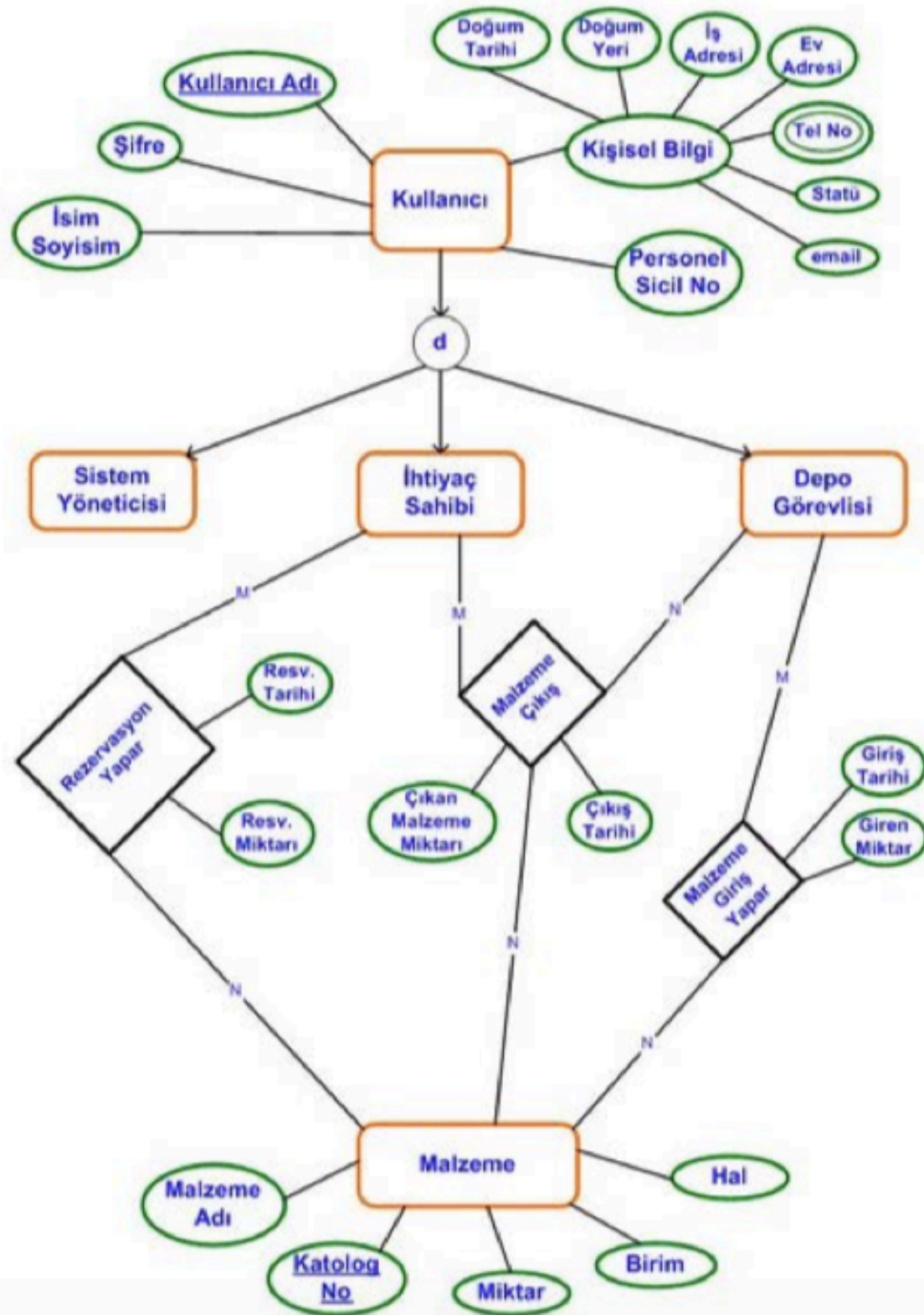
- Sorun ile karşılaşan uzman, biçimsel spesifikasyonu anlayamayabilir. Biçimsel spesifikasyondan anlayan yazılımcı da uygulama etki alanını anlamayabilir.
- Spesifikasyonun kullanımı sonucunda elde edilecek olası maliyet tasarrufu öngörülemez.
- Yazılımcılar biçimsel spesifikasyona hakim değillerse uzak durmak isteyebilirler.
- Büyük sistemlere ölçeklemek zordur. Sistemin tümü yerine kritik kısımlar betimlenebilir.
- Araç desteği sınırlıdır ve kullanımı zordur.
- Çevik geliştirme yöntemleri ile uyumlu değildir.

Veri Modelleme Yöntemleri

- Önerilen sistemin mantıksal modelinde veri yapısını açıklamak amacıyla 'Veri Modelleme' yöntemleri kullanılmaktadır. Bu yöntemler, veri yapısını çeşitli düzeylerde tanımlama (en soyut düzeyden en ayrıntı düzeye kadar) amacını güder. Sistem Çözümleme aşamasında en yaygın olarak kullanılan veri modelleme yöntemleri:
- Nesne İlişki Şemaları
- Veri Sözlüğü

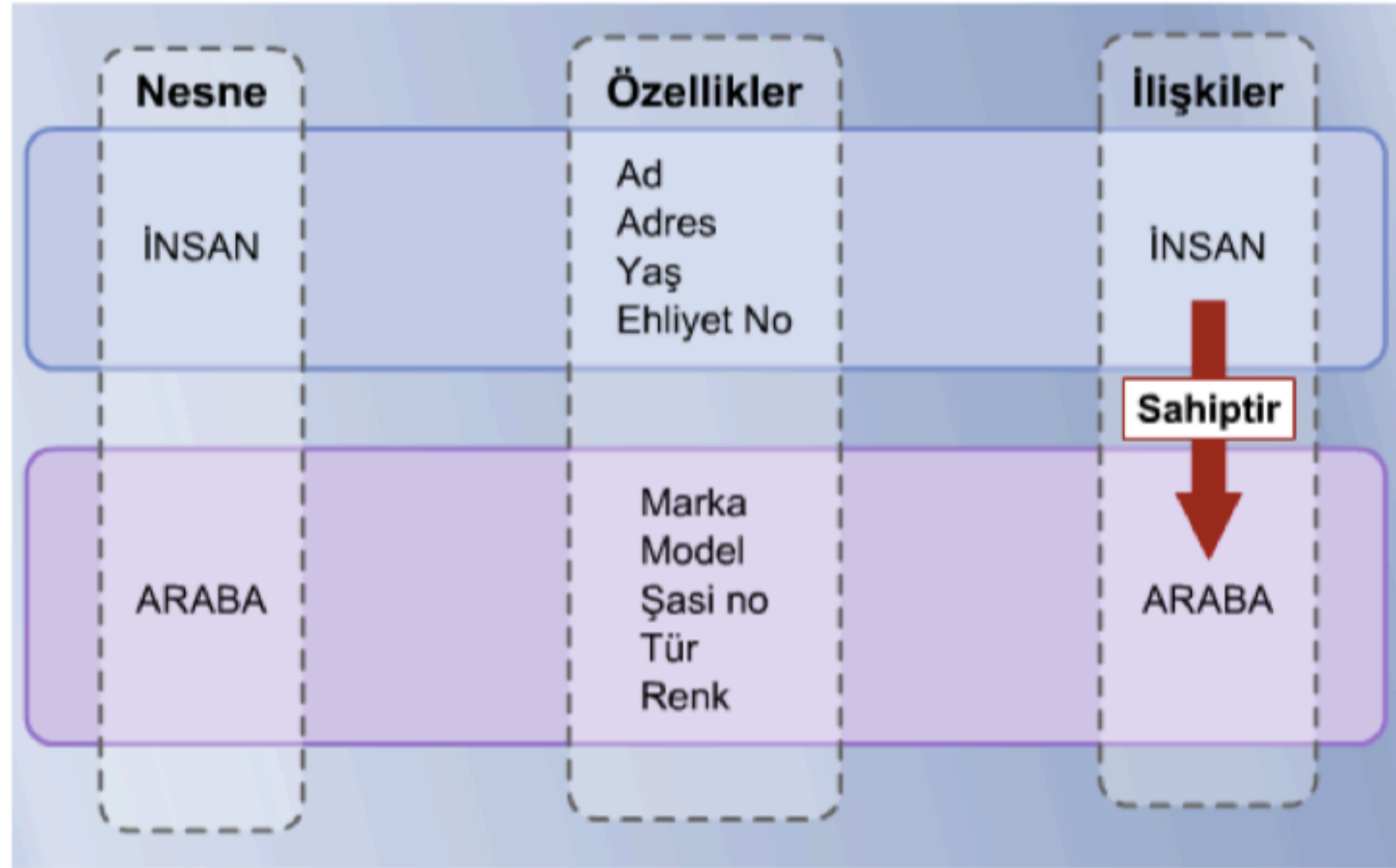
Nesne İlişki Şemaları

- Veri tasarımı açısından çok önemlidir.
- Sistemde ana veri nesneleri ve aralarındaki ilişkiyi belirtir.
- CASE araçları ile nesne ilişki şemaları otomatik veritabanı tablosu yapılarına dönüştürülebilmekte.



- Veri nesnesinin üç özelliği vardır;
 - Adı
 - Özellikleri
 - Veri nesnesi varlığının diğer veri nesnesi varlıklarına referansı
- Bu amaçla her bir veri nesnesini tek olarak belirleyen bir belirteç (anahtar) kullanılır. Söz konusu anahtar veri nesnesi varlığının ad özellikleri arasında yer alır.

Araba ve İnsan adlı iki veri nesnesi ele alalım.



Marka	Model	Şasi no	Tür	Renk	Sahibi
BMW	740L	XZ68...	Sedan	Siyah	AA
VOLVO	S80	A32XT..	Sport	Sarı	BT
Ford	Taurus	X24R56	Hatchback	Bej	NB

- Veri nesneleri arasındaki ilişkiler bire bir (1-1), bir den çoğa (1-N ya da N-1) ya da çoktan çokta (M-N) tanımlanabilir.

Veri Sözlüğü

- Nesne ilişki şemalarında belirtilen nesne özelliklerinin ayrıntılı tanımları Veri Sözlüğünde yer alır.
- Söz konusu ayrıntılı tanımlar genel olarak aşağıdaki bilgileri içerir:
 - Veri Adı
 - Veri Eş-adı (Aynı veri için kullanılan diğer ad)
 - Nerede/nasıl kullanıldığı
 - İçerik tanımı

Örnek: Kişi telefon bilgisinin tanımlanması

telefon no = [yer kodu | numara]

yer kodu = [212 | 242 | | 312]

numara = * yedi basamaklı herhangi bir sayı *

Veri Yapısı	Gösterim	Açıklama
Sıra Tanımlama	=	.. den oluşmuştur
Seçim	+	ve
Yineleme	[]	ya da
	{ } ⁿ	n kez yinelenir
	()	seçenek veri
	* *	açıklama

Süreç / İşlem Modelleme Yöntemleri

- Geliştirilecek sistemin süreç ya da işlemlerini ve bu süreçler arasındaki ilişkileri tanımlamak amacıyla kullanılan yöntemlerdir.
 - Veri Akış Diyagramları
 - Süreç Tanımlama Dili
 - Karar Tabloları
 - Karar Ağaçları
 - Nesne Şemaları

Veri Akış Diyagramları

Veri Akış Diyagramları kullanılarak geliştirilecek sistemin mantıksal modeli, '*Yukarıdan Aşağıya*' bir yaklaşımla oluşturulur.

Sistem önce en genel biçimiyle ele alınır, yalnızca dışsal ilişkileri incelenir. Daha sonra, sistemin iç yapısındaki süreçler ve bu süreçler arasındaki ilişkiler belirlenen bir ayrıntı düzeyine kadar modellenir.

4 temel sembol kullanılarak oluşturulur.



- VAD, bilgi sisteminin durağan yapısını gösterir.
- VAD, bilgi sisteminin süreçlerini, bu süreçler arasındaki veri akış ilişkilerini gösterir.
- VAD, bilgi sistemi süreçleri ile ilgili olan kurum birimlerini ya da dış birimleri bilgi kaynakları olarak gösterir.
- VAD, bilgi sistemi için gerekli olan ana veri depolarının neler olduğunu ve hangi süreçler tarafından kullanıldığını gösterir.
- VAD, bilgi sistemi süreçlerini, yukarıdan aşağıya doğru ayrıştırarak gösterir. Böylelikle süreçler ve aralarındaki ilişki, en soyut (genel) düzeyden en ayrıntılı düzeye kadar belirli bir sıra düzen içerisinde belirtir.

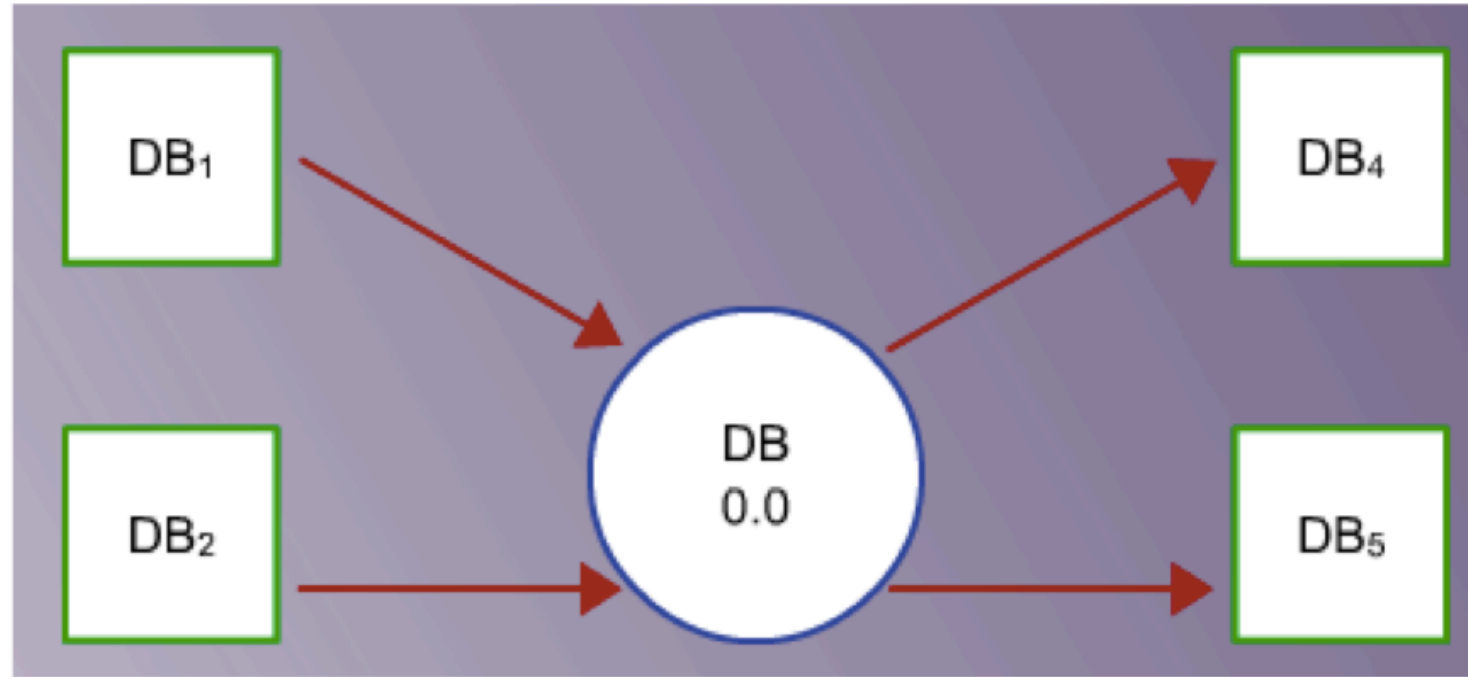
- VAD bilgi sistemi süreçlerinin zamana ilişkin durumunu ve bu durumla ilgili bilgileri göstermez.
- VAD, bilgi sistemi süreçlerinin kendi aralarındaki "karar" ilişkisini göstermez.
- VAD, gerek bilgi sistemi süreçleri, gerek veri akışları gerekse bilgi kaynakları ve bilgi depoları için ayrıntı içermez.

.

- Mantıksal veri yığınlarını göstermek amacıyla ucu açık dikdörtgenler kullanılmaktadır.
- Süreçler arası, süreçler ile veri kaynakları ve veri depoları arasındaki veri akış ilişkileri göstermek amacıyla oklar kullanılır.

- Temel olarak bir sistemin mantıksal modelinin süreç yapısı, üç tür Veri Akış Diyagramı çizilerek elde edilir.
 - Kapsam diyagramı: Geliştirilecek bilgi sisteminin dışsal ilişkilerini göstermek amacıyla kullanılır. Yalnızca bir çember ve gerekli sayıda veri kaynağı ile aralarındaki veri akışlarının belirtiminden oluşur.
 - Genel bakış diyagramı: Bir bilgi sisteminin ana işlevlerini, bu işlevlere ilişkin veri kaynaklarını, veri depolarını ve işlemler arasındaki ilişkileri içeren diyagramlardır. Kapsam diyagramının ayrıştırılmış ya da “patlatılmış” biçimidir. Bilgi sisteminin büyüklüğüne göre en çok bir kaç Genel bakış diyagramı çizilir.
 - Ayrıntı diyagramı: Genel bakış diyagramında belirtilen her ana işlev, belirli bir ayrıntı düzeyine kadar (7 +/-2) detay VAD oluşturularak ayrıştırılır. Veri kaynakları gösterilmez.

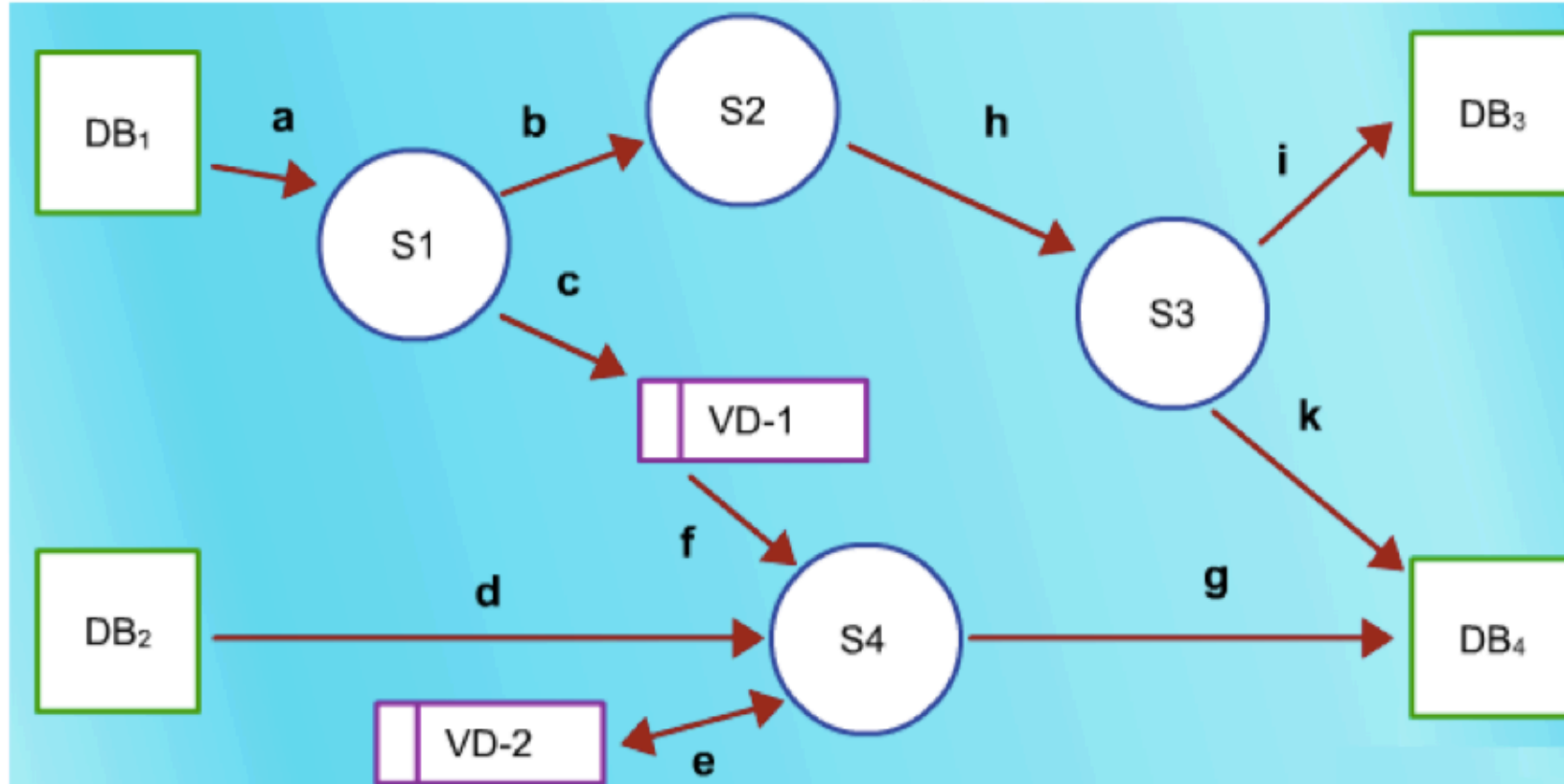
Kapsam Diyagramı



(DB: Dış Birim)

- **VAD**, bilgi sisteminin durağan yapısını **gösterir**.
- **VAD**, bilgi sisteminin süreçlerini, bu süreçler arasındaki veri akış ilişkilerini **gösterir**.
- **VAD** bilgi sistemi süreçlerinin zamana ilişkin durumunu ve bu durumla ilgili bilgileri **göstermez**.

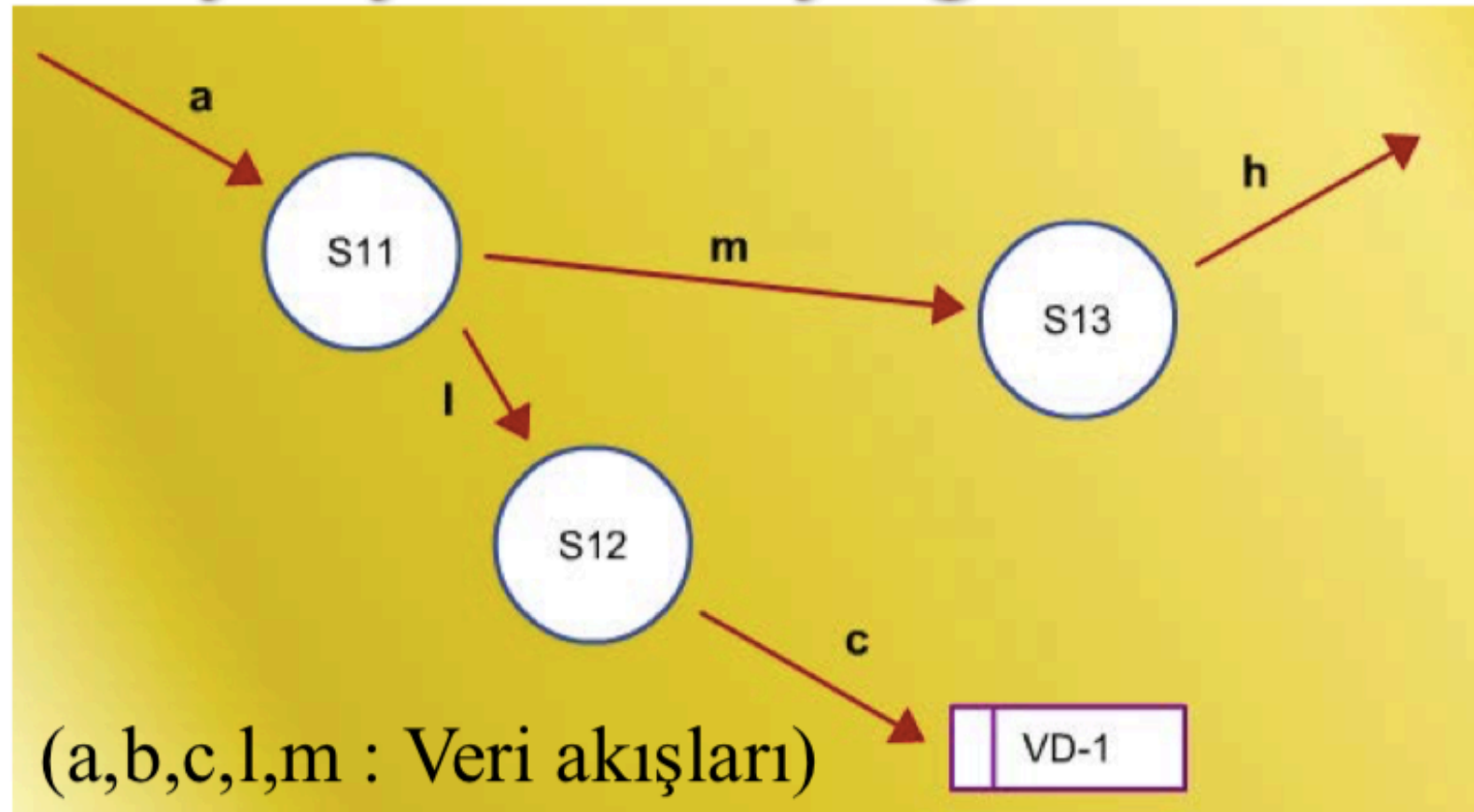
Genel Bakış Diyagramı



(S: Süreç, VD: Veri Deposu, a,b,c,d,e,f,g,h,i,k : Veri akışları)

- **VAD**, bilgi sistemi süreçleri ile ilgili olan kurum birimlerini ya da dış birimleri bilgi kaynakları olarak **gösterir**.
- **VAD**, bilgi sistemi süreçlerinin kendi aralarındaki "karar" ilişkisini **göstermez**.

Detay Ayrıntı Diyagramı



- **VAD**, bilgi sistemi için gerekli olan ana veri depolarının neler olduğunu ve hangi süreçler tarafından kullanıldığını **gösterir**.
- **VAD**, gerek bilgi sistemi süreçleri, gerek veri akışları gerekse bilgi kaynakları ve bilgi depoları için ayrıntı **içermez**.

Süreç Tanımlama Dili

- Veri akış diyagramlarında isimleri belirtilen, aralarındaki ilişkiler gösterilen ve yukarıdan aşağıya ayrıştırılmış olan bilgi sistemi süreçlerinin iç yapılarını belirtmek amacıyla kullanılan araç, yöntem ya da gösterim biçimleri Süreç tanımlama dili olarak tanımlanmaktadır.
- Düz Metin,
- Şablon,
- Yapısal İngilizce

Üçgeni İncele süreci üçgenin kenar boyutlarını (A,B,C) girdi olarak alır. Süreç önce bütün bu değerlerin pozitif olup olmadığını denetler. Eğer değerlerden biri negatif ise hata iletisi verir. Süreç, tüm kenar boyutlarının bir üçgeni belirleyecek biçimde geçerli olup olmadığını denetler. Eğer geçerli değerler ise üçgenin türünü (eşkenar üçgen, ikizkenar üçgen, çeşitkenar üçgen) belirler. Değil ise hata iletisi verir.

Süreç: Üçgeni İncele

Girdi: Üçgenin kenar boyutları (A,B,C)

Çıktı: Üçgen Türü, Hata iletisi

İşlem: A,B,C değerlerin pozitif olup olmadığını denetle. Eğer değerlerden biri negatif ise hata iletisi ver. A,B,C değerlerinin geçerli olup olmadığını denetle. Eğer geçerli değerler ise üçgenin türünü (eşkenar üçgen, ikizkenar üçgen, çeşitkenar üçgen) belirle. Değil ise hata iletisi ver.

Procedure: Üçgeni İncele

Üçgenin kenar boyutlarını oku;

if herhangi bir boyut negatif **then** hata iletisi ver; **endif;**

if en büyük kenar diğer ikisinin toplamından küçük **then**
begin

eşit kenar sayısını belirle;

if üç kenar eşit **then** tür eşkenardır; **endif;**

if iki kenar eşit **then** tür ikizkenardır; **endif;**

if hiç bir kenar eşit değil **then** tür çeşitkenardır; **endif;**

üçgen türünü yaz;

end;

else değerler üçgeni belirtmiyor iletisi ver; **endif;**

endproc



Tekrar

Kullanıcı Arayüz (Interface) Prototipleme

- Gereksinim tanımlama ya da sistem çözümleme çalışmasının önemli bir bileşeni, oluşturulacak bilgi sistemine ilişkin girdi ve çıktı gereksinimlerinin tanımlanmasıdır.
- KAP Yöntemi, gereksinim çalışmasının hemen sonunda kullanıcıya gösterilecek bir prototip yazılım hazırlanmasını içermektedir. Söz konusu prototipin gerçekten içsel olarak çalışmayan ancak ekranlar, menüler ve bunların aralarındaki geçişlerin çalıştığı bir yazılımdır.

.

1. KAP hazırlama süresi uygulamanın boyutu ne olursa olsun, ekranlar ve ekranlar arası geçişlerin hazırlanması için gerekecek iş gücü ve zaman, sistem çözümleme için ayrılan zamanın %5'ini aşmamalıdır.

2. Bir özellik yalnızca bir kez gösterilmelidir. Bir özellik birden fazla ekranda yer alsa bile yalnızca bir ekranda gösterilmelidir. Örneğin, şehir listesini içeren bir kombo listesi, bir kaç şehir bilgisi içermeli ve yalnızca bir ekranda kullanıcıya gösterilmelidir. Başka ekranlarda aynı liste varsa, gösterilen ile aynı olduğu belirtilmelidir.

3. KAP hiçbir içsel işlem içermemelidir. Örneğin: bir ekranda kişilerle ilgili sorgulama yapıldığını düşünelim. Adı soyadı bilinen bir kişinin, bu bilgi verildiğinde, diğer ayrıntı bilgilerinin ekranda görüntüleneceğini varsayalım. Kullanıcı, Ad Soyad bilgisini girecek ve ilgili sorgulama tuşunu etkinleştirdikten sonra ekranda hep aynı bilgiler görünecektir. Yani, KAP, Ad soyad bilgisine göre KİŞİ tablosundan arama yapmayacak, program içerisinde hep aynı sonuçları ekrana verecektir. Çünkü Kişi tablosunun yaratılması ya da kullanılması söz konusu değildir. Burada amaç, kullanıcıya, sorgulama biçimini ve sorgulama sonucunda elde edilecek bilgileri açıklamaktır. Sorgulanan bilginin içeriği bu aşamada önemli değildir. Kullanıcıdan, sorgulama sonucu gelen bilgi türlerine ilişkin, sorgulama işleminin işlem akışına ilişkin bilgi alınması hedeflenir.



Çözümleme Çalışması Nasıl Değerlendirilir

- Sistem çözümleme çalışması sonuçlandıktan sonra, elde edilen ara ürünün (mantıksal model) istenenleri karşılayıp karşılamadığının belirlenmesi amacıyla değerlendirilmesi gerekir. Bu nedenle temel olarak:
 - Kullanıcıların benimsemesi ve onaylaması
 - Tutarlılık, tamlık ve uygunluk
 - Tasarım çalışması için yeterliliği

- Tamlik ve Tutarlilik: Tamlik, bilgi sistemi ya da yazılımın tüm ögeleri ve bunların arasındaki ilişkilerin tanımlanmasını gerektirir. Tutarlilik ise, önerilen modelin kendi içerisinde hatasız, çelişkisiz olması anlamındadır. Tamlik ve tutarlilik denetimi, bir programlama dilinde yazılmış bir programın söz dizim kurallarının denetimine benzer. Örneğin, modelleme aracı olarak VAD kullanıldığında;

- Bütün süreçlerin girdi ve çıktıları var mı?
- Süreçler, veri akışları, dış birim ve veri depoları uygun şekilde adlandırılıp numaralandırılmış mı?
- VAD' de ki tüm isimler veri sözlüğünde yer almış mı?
- Planlama aşamasında öngörülen belirtilmelerin hepsi ele alınmış mı?

Olurluluk

- Sistem çözümleme sırasında yapılan çalışmanın, planlama aşamasında yapılan çalışmaya uygunluğunun belirlenmesi için yapılan çalışmaları içerir.
- Maliyet kestirim çalışması yinelenir, sapmalar saptanır, kaynaklar yeniden planlanır. Sapmalar oldukça fazla olursa, yapılan çalışmanın yeniden gözden geçirilmesi gerekir.
- Bu aşamada, maliyet kestirimi için daha fazla bilgi mevcuttur. VAD yöntemi, maliyet kestirim yöntemlerinin kolayca uygulanabilmesini olanaklı kılar.

- Örneğin İşlev Noktalarının belirlenmesinde VAD'den:

Girdi ve Çıktı Sayıları: Girdi ve Çıktı sayıları doğrudan genel bakış diyagramındaki veri akışlarının incelenmesiyle elde edilir.

Kütük Sayısı: Genel bakış diyagramlarındaki veri depoları sayılarak elde edilir.

Sorgu Sayısı: Sorgu sayısı genel bakış diyagramı ve detay diyagramlardaki süreçlerin incelenmesiyle elde edilir.

Arayüz Sayısı: Arayüz sayısı, kapsam diyagramındaki dış birimlerle olan ilişkiler incelenerek elde edilir.

•

Kaynakça

- Ian Sommerville, Yazılım Mühendisliği
- <http://w3.bilecik.edu.tr/bilgisayar/wp-content/uploads/sites/75/2019/02/ders-4.pdf>
- <http://muhammetbaykara.com>
- Ali Arifoğlu ders notları
- <http://www.kriptarium.com/gereksinim.html>