

YAZ16103

Yazılım Mühendisliğine Giriş

Dr. Öğr. Üyesi Bora ASLAN

İÇİNDEKİLER

BÖLÜM 5: SİSTEM MODELLEME System modeling

BAĞLAM MODELLERİ Context models

ETKİLEŞİM MODELLERİ Interaction models

YAPISAL MODELLER Structural models

DAVRANIŞSAL MODELLER Behavioral models

MODEL GÜDÜMLÜ MİMARİ Model-driven engineering

SİSTEM MODELLEME System modeling

- Sistem modelleme, bir sistemin soyut modellerinin geliştirilmesi sürecidir.
- Her bir soyut model söz konusu sistemin farklı bir görünümünü veya perspektifini sunar.
- Sistem modelleme günümüzde bir sistemin UML (Birleşik Modelleme Dili - Unified Modeling Language) diyagram tiplerinin kullanılmasına dayanan bir çeşit grafiksel gösterim ile temsil edilmesi anlamına gelmektedir.
- Modeller gereksinim mühendisliği sürecinde sistemin detaylı gereksinimlerini saptamak amacıyla; tasarım sürecinde sistemi gerçekleştiren mühendislere, sistemi betimlemek amacıyla ve gerçekleştirim sonrasında da sistemin yapısını ve işlevini belgelemek amacıyla kullanılırlar.

SİSTEM MODELLEME System modeling

- UML yazılım sistemlerini modellemede kullanılabilecek 13 farklı diyagram tipinden oluşmuştur.
- UML 1990'lardaki nesneye yönelik modelleme çalışmalarına dayanır.
- 2004'te geniş kapsamlı bir revizyon yapılmış ve UML 2 ortaya çıkmıştır.
- UML, yazılım sistemlerinin modellerinin geliştirilmesi için standart hale gelmiştir.
- Daha genel sistem modelleme görevleri için SysML gibi varyantları önerilmektedir.

<http://www.uml.org/>

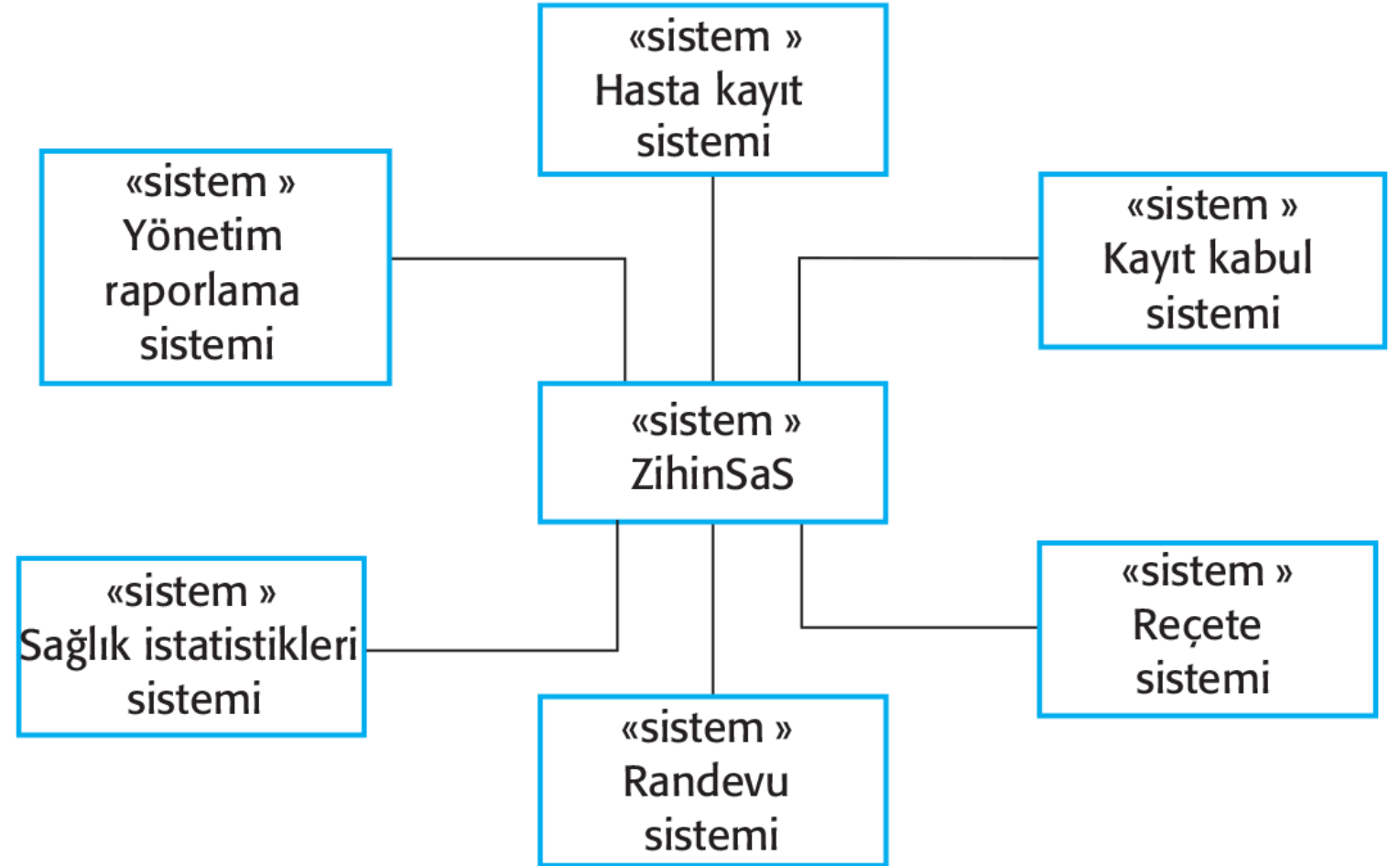
<https://sysml.org/>



BAĞLAM MODELLERİ - Context models

Sistem tanımlamanın erken bir safhasında sistemin sınırlarına karar vermeniz gerekir ki bu neyin geliştirilmekte olan sistemin içinde kaldığına, neyin ise dışarıda bırakılacağına karar vermektir. Bu iş, paydaşlarla birlikte sistemin hangi işlevleri içereceğine karar vermek ve sistemin operasyonel çevresindeki işleme ve çalışma biçimlerini belirlemek ile ilgilidir.

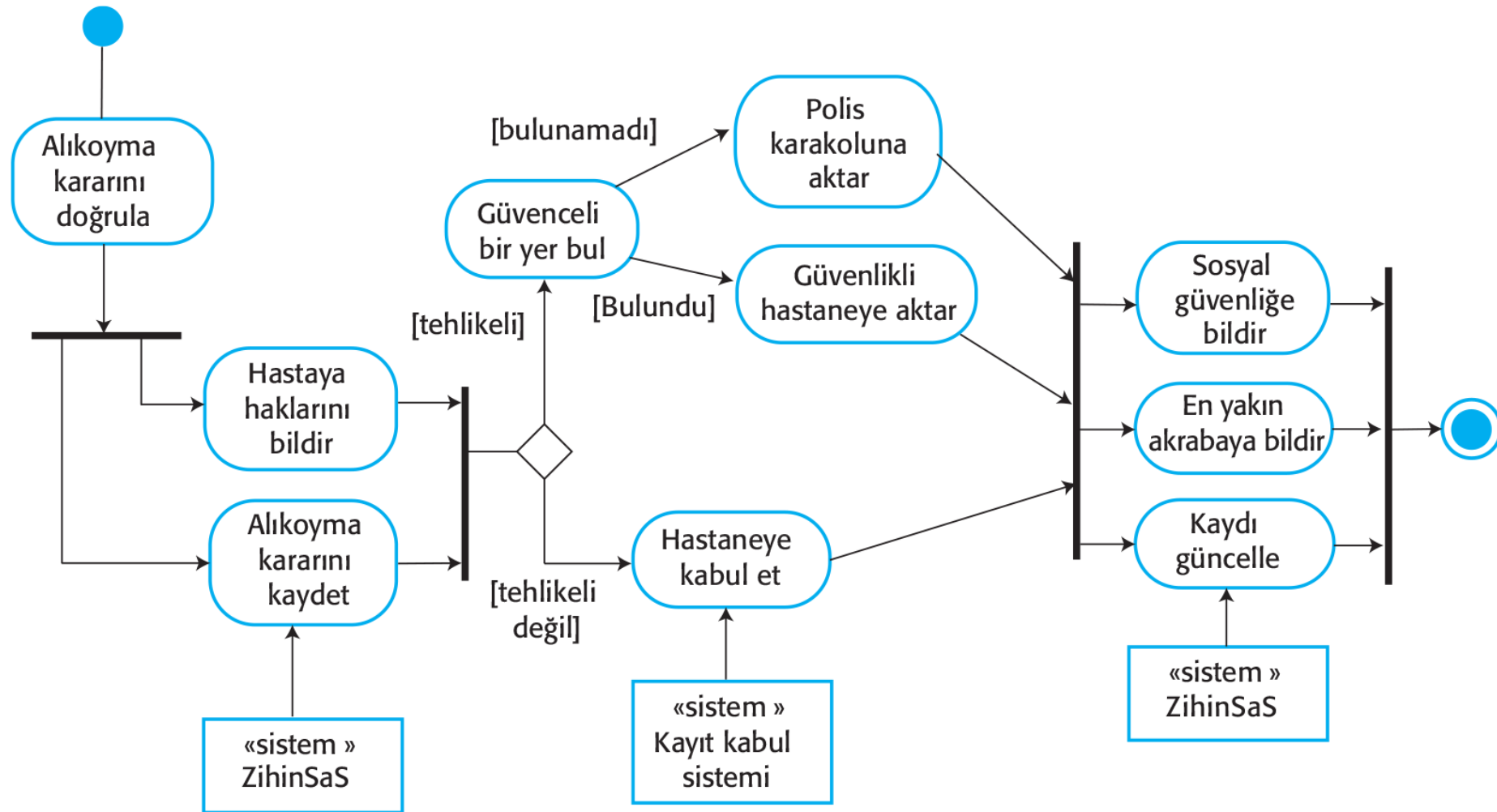
Bazı iş süreçleri için otomasyon desteği vermek isterken bazı başka süreçleri elle yapmak veya başka sistemlerle desteklemek isteyebilirsiniz. Sistemin işlevlerinin bazı var olan sistemlerle çakışmasını kontrol etmek ve yeni işlevlerin nerede gerçekleştirilmesi gerektiğine karar vermelisiniz. Maliyeti ve sistem gereksinimleri ile tasarımı anlamak için gereken zamanı azaltmak için bu kararlar sürecin başlarında alınmalıdır.



Şekil 5.1 ZihinSaS sisteminin bağlamı

BAĞLAM MODELLERİ - Context models

- Bağlam modelleri ortamdaki pek çok başka otomasyon sistemini kapsayabilir.
- Fakat betimlenen sistem ile çevredeki diğer sistemler arasındaki ilişkileri göstermezler.
- Dış sistemler, betimlenen sisteme veri sağlayabildiği gibi sistemin ürettiği veriyi alıp kullanabilirler.
- Tüm bunlar tanımlanan sistemin gereksinimlerini ve tasarımını etkileyebilir ve bu yüzden dikkate alınmak zorundadır.
- Bu sebeple basit bağlam modelleri iş süreç modelleri gibi modeller ile tasarlanır ve tanımlanır.



Şekil 5.2 İstemsiz alıkoymanın süreç modeli

activity diagram

- UML etkinlik diyagramları sistemin kullanıldığı iş süreçlerini göstermek için kullanılabilir.

ETKİLEŞİM MODELLERİ - Interaction models

Tüm sistemler bir tür etkileşim gerektirir. Etkileşim, kullanıcı girdi ve çıktılarını içeren kullanıcı etkileşimi olmanın yanı sıra geliştirilmekte olan sistem ile ortamdaki diğer sistemler arasında ve nihayet sistemin bileşenleri arasında yer alan etkileşimdir.

Kullanıcı etkileşimi modelleme kullanıcı gereksinimlerini belirlemeyi sağlaması nedeniyle önemlidir. Sistemler arası etkileşimi modellemek iletişim problemlerini göz önüne getirir. Bu kısımda etkileşim modellemeye yönelik birbirleriyle ilintili iki yaklaşım tartışılacaktır.

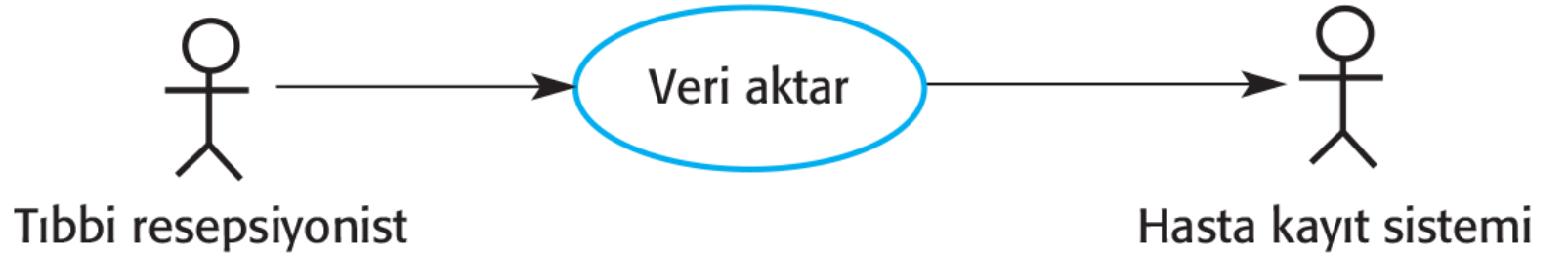
1. Kullanım durumu modelleme (**Use case modeling**): Genellikle sistem ile dış faktörler arasındaki (insan kullanıcı veya başka bir sistem) etkileşimin modellenmesidir.
2. Sıra diyagramları (**Sequence diagrams**): Sistem bileşenleri arasındaki (dış faktörler dâhil) etkileşimin modellenmesidir.

ETKİLEŞİM MODELLERİ - Interaction models

Kullanım durumu modelleme - Use case modeling

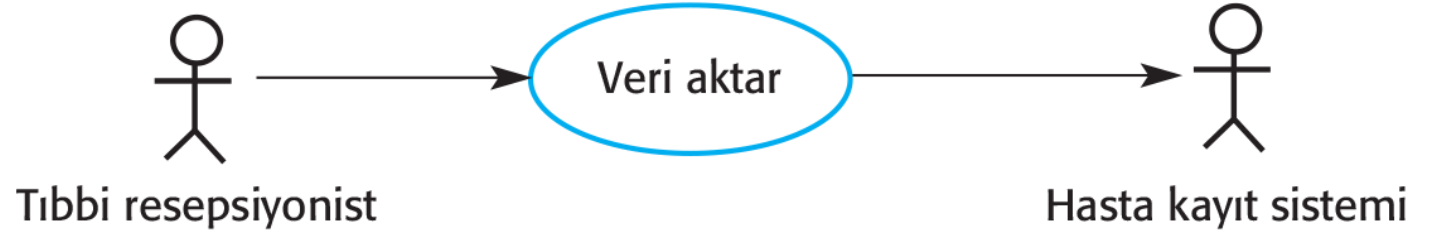
Kullanım durumu bir kullanıcının belirli bir etkileşim sonucu sistemden beklediklerinin basit bir tarifidir ve sistem tasarımının erken evrelerinde kullanılır.

Her bir kullanım durumu bir sistem ile dışarıdan ilişki kurmayı içeren ayrık bir görevi temsil eder.



Şekil 5.3 Veri aktar kullanım durumu

Şekil 5.3 Veri aktar kullanım durumu

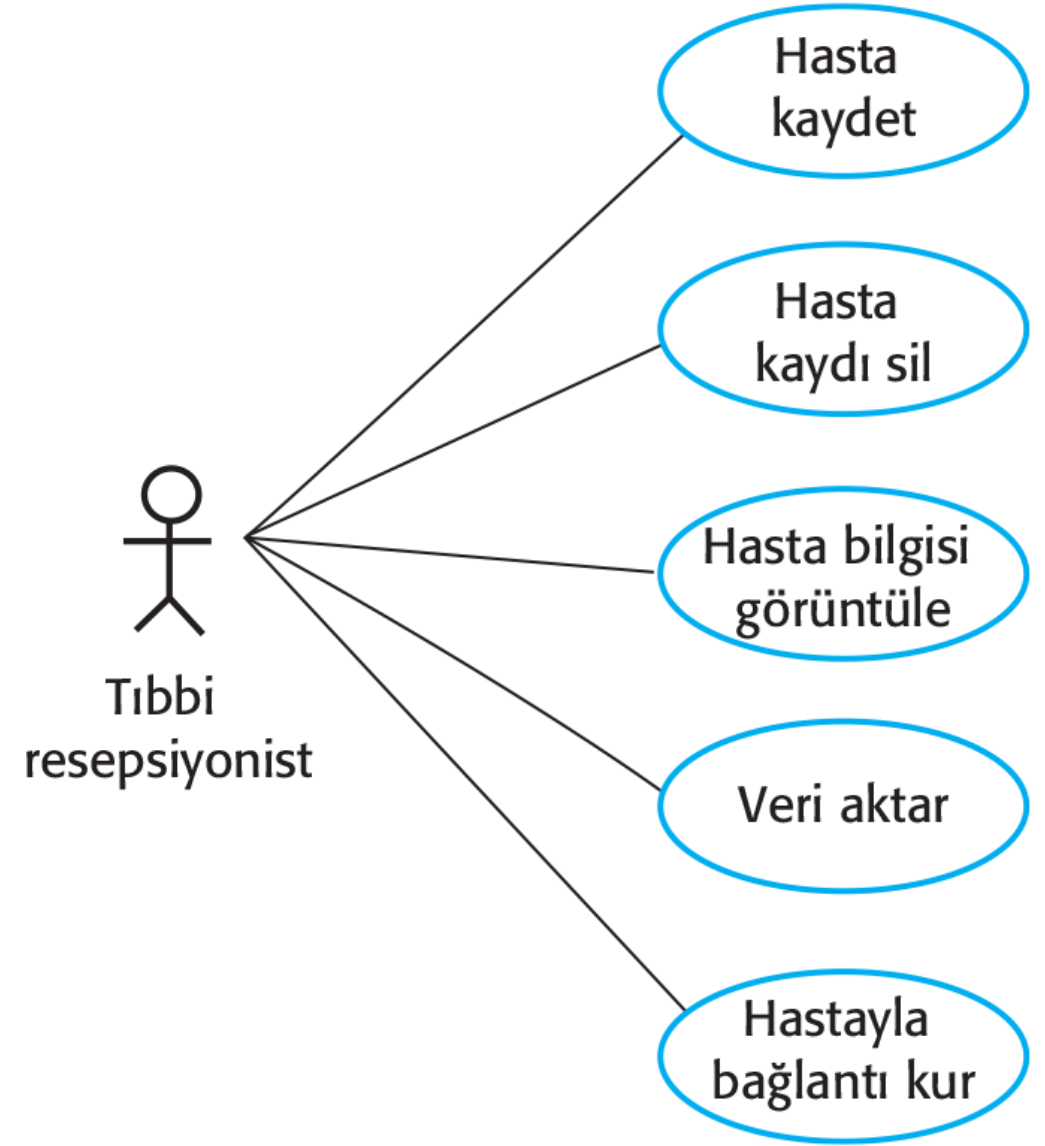


ZihinSaS sistemi: Veri aktar

Aktörler	Tıbbi resepsiyonist, Hasta kayıt sistemi (HKS)
Betimleme	Bir resepsiyonist ZihinSaS sisteminden bir sağlık otoritesince yürütülen bir hasta kayıt sistemine veri aktarabilir. Aktarılan bilgi güncellenmiş kişisel bilgiler olabilir (adres, telefon numarası vs.) veya hastaya ilişkin teşhis ve tedavinin bir özeti olabilir.
Veri	Hastanın kişisel bilgileri, tedavi özeti
Uyaran	Tıbbi resepsiyonist tarafından verilen kullanıcı komutu
Tepki	HKS'nin güncellendiğinin teyidi
Yorumlar	Resepsiyonist hasta bilgisine ve HKS'ye erişmek için gerekli güvenlik izinlerine sahip olmalıdır.

Şekil 5.4 Veri aktar kullanım durumunun tablo gösterimi

- Bileşik kullanım durumu diyagramları bir seferde birçok kullanım durumunu gösterebilirler.
- Bazen bir sistemde mümkün olan tüm etkileşimleri tek bir bileşik kullanım durumu diyagramında göstermek mümkün olabilir.
- Ne var ki kullanım durumlarının çokluğu nedeniyle bu mümkün olamayabilir.
- Bu durumda her biri ilgili kullanım durumlarını gösteren birçok diyagram yaratabilirsiniz.



ETKİLEŞİM MODELLERİ - Interaction models

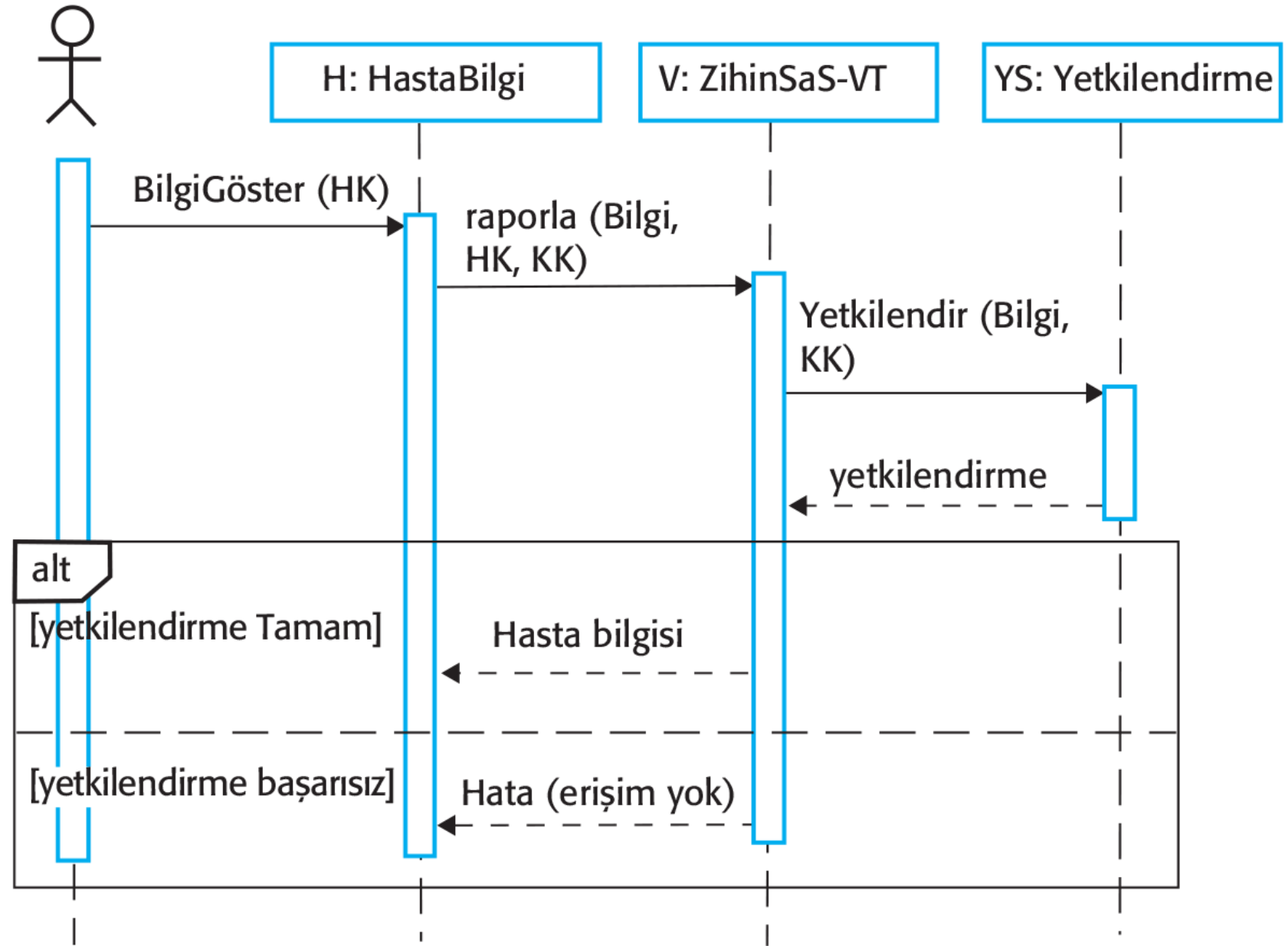
Sıra Diyagramları - Sequence diagrams

Sıra diyagramları genel olarak bir sistemin aktörleri ile nesneleri arasındaki ya da nesnelerinin kendi aralarındaki etkileşimleri modellemek amacıyla kullanılır.

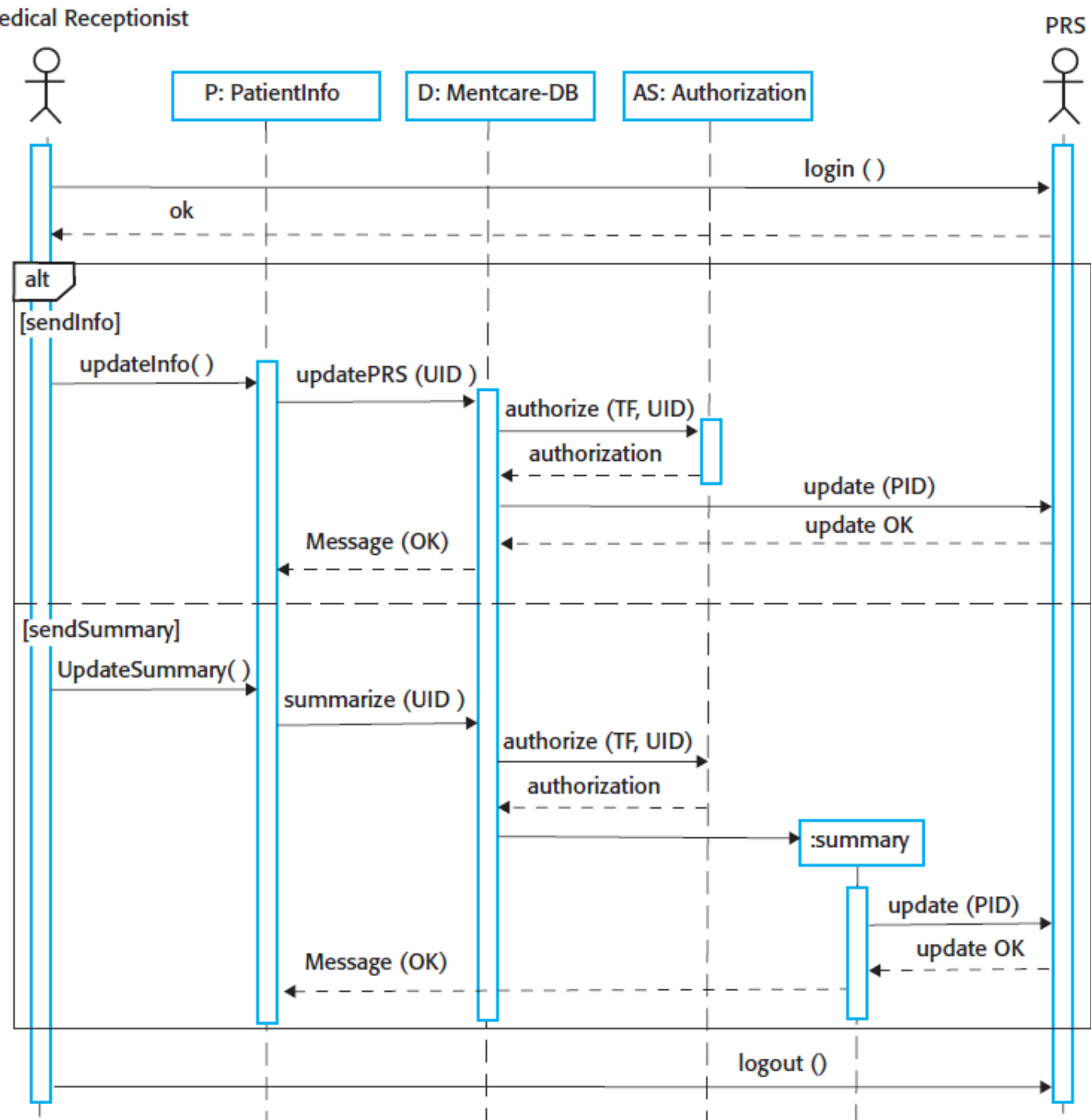
Sıra diyagramı bir kullanım durumu ya da kullanım durumu örneği içinde meydana gelen sıralı etkileşimler serisini gösterir.

Sıra Diyagramları Sequence diagrams

Tıbbi resepsiyonist



Şekil 5.6 "Hasta bilgisi görüntüle" için sıra diyagramı



YAPISAL MODELLER - Structural Models

- Yapısal modeller bir sistemin organizasyonunu o sistemi oluşturan bileşenler ve bileşenler arasındaki ilişkiler cinsinden gösterirler.
- Yapısal modeller sistem tasarımını gösteren statik modeller olabildikleri gibi sistemin çalışma esnasındaki organizasyonunu gösteren dinamik modeller de olabilir.
- Bu ikisi aynı şey değildir; bir sistemin etkileşimli iş parçacıkları biçimindeki dinamik organizasyonu sistem bileşenlerini içeren statik modelden çok farklı olabilir.
- Bir sistemin yapısal modellerini, sistem mimarisini tasarlarken ve tartışırken yaratırız. Bunlar tüm sistem mimarisinin modelleri olabilirler veya sistem içindeki nesneler ve bunların ilişkilerini içeren daha detaylı modeller olabilirler.
- Bir sistemin yapısal modellerini, sistem mimarisini tasarlarken ve tartışırken yaratırız.
- Bunlar tüm sistem mimarisinin modelleri veya sistem içindeki nesneler ve bunların ilişkilerini içeren daha detaylı modeller olabilirler.

YAPISAL MODELLER - Structural Models

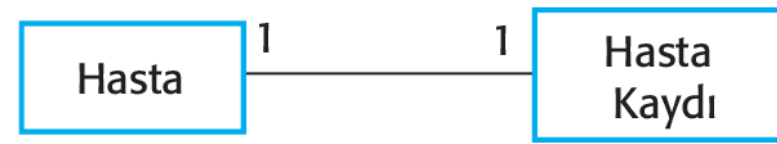
Sınıf Diyagramları - Class Diagrams

- Sınıf diyagramları bir nesneye yönelik sistem modeli geliştirirken sınıfları ve sınıflar arasındaki bağları göstermek için kullanılırlar.
- Yazılım mühendisliği sürecinin erken evrelerinde modeller geliştiriyorsanız, nesneler gerçek dünyadaki şeyleri temsil eder; Örneğin hasta, reçete veya doktor gibi.
- Bir model geliştirilirken ilk adım genellikle dünyaya bakmak, gerekli nesneleri tespit etmek ve bunları sınıflarla temsil etmektir.
- Bu diyagramların en basit halleri kutucuklar içine yazılmış sınıf isimlerini içerir.

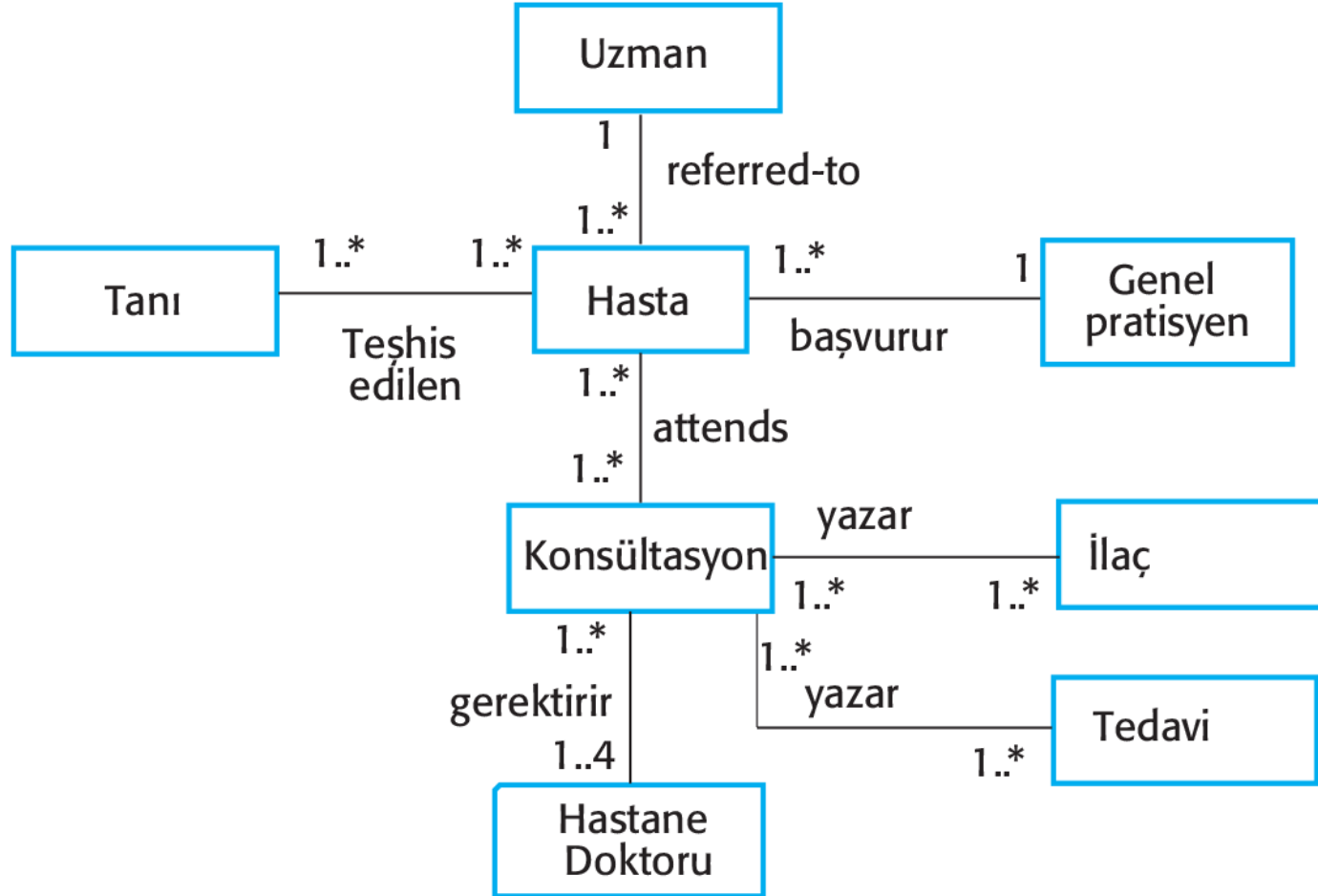
Şekil 5.8 UML Sınıfları ve bağlar



Şekil 5.8 UML Sınıfları ve bağlar



Şekil 5.9 ZihinSaS sistemindeki sınıflar ve bağlar



Sınıf Diyagramları - Class Diagrams

- UML'de özellik ve işlemleri göstermek için sınıfı temsil eden basit dikdörtgene yeni bilgiler eklenir.
 - Üst bölümde nesne sınıfının adı.
 - Orta bölümde sınıf özellikleri. Bu bölümde adlar ve seçimlik olarak özelliklerin tiplerini içerir.
 - Dikdörtgenin alt bölümünde nesne sınıfına ilişkin tüm işlemler (fonksiyon, metod, prosedür)

Konsültasyon

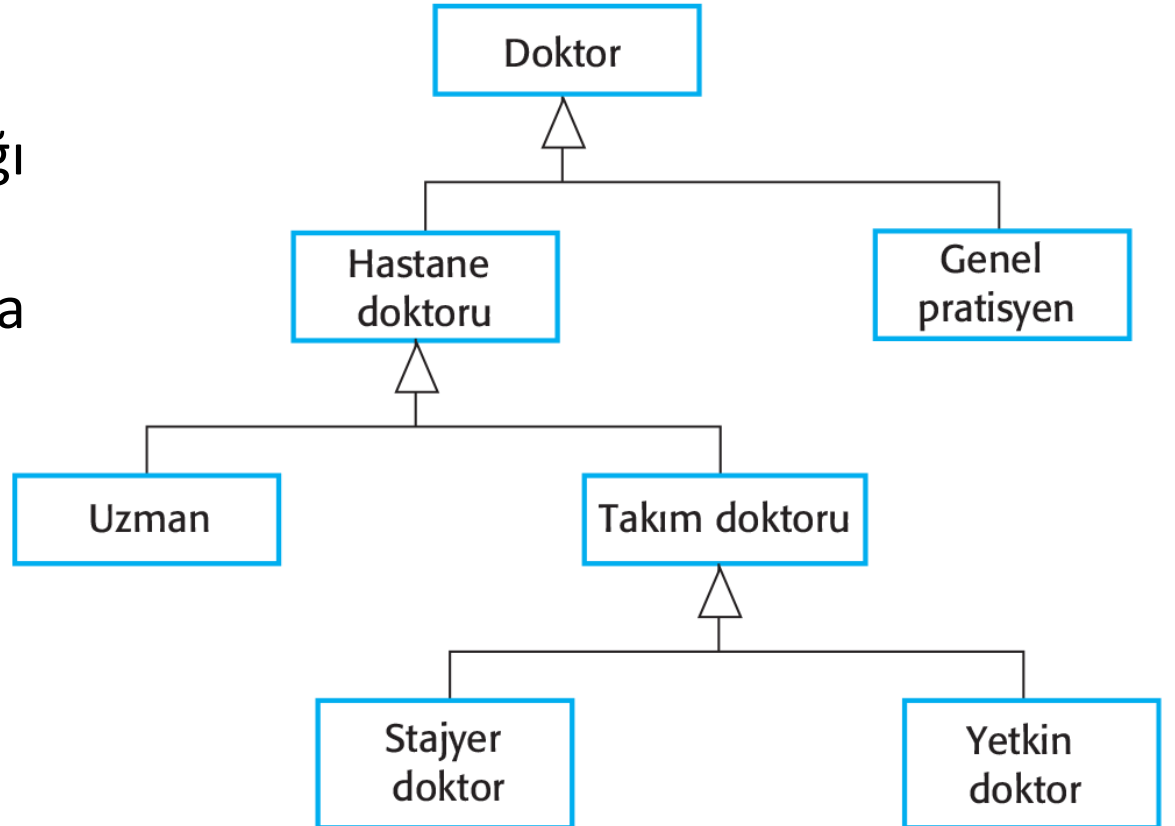
Doktorlar
Tarih
Saat
Klinik
Sebep
Yazılan ilaç
Yazılan tedavi
Sesli notlar
Transkript
...

Yeni ()
ReçeteYaz ()
NotKaydet ()
TranskriptYaz ()
...

YAPISAL MODELLER - Structural Models

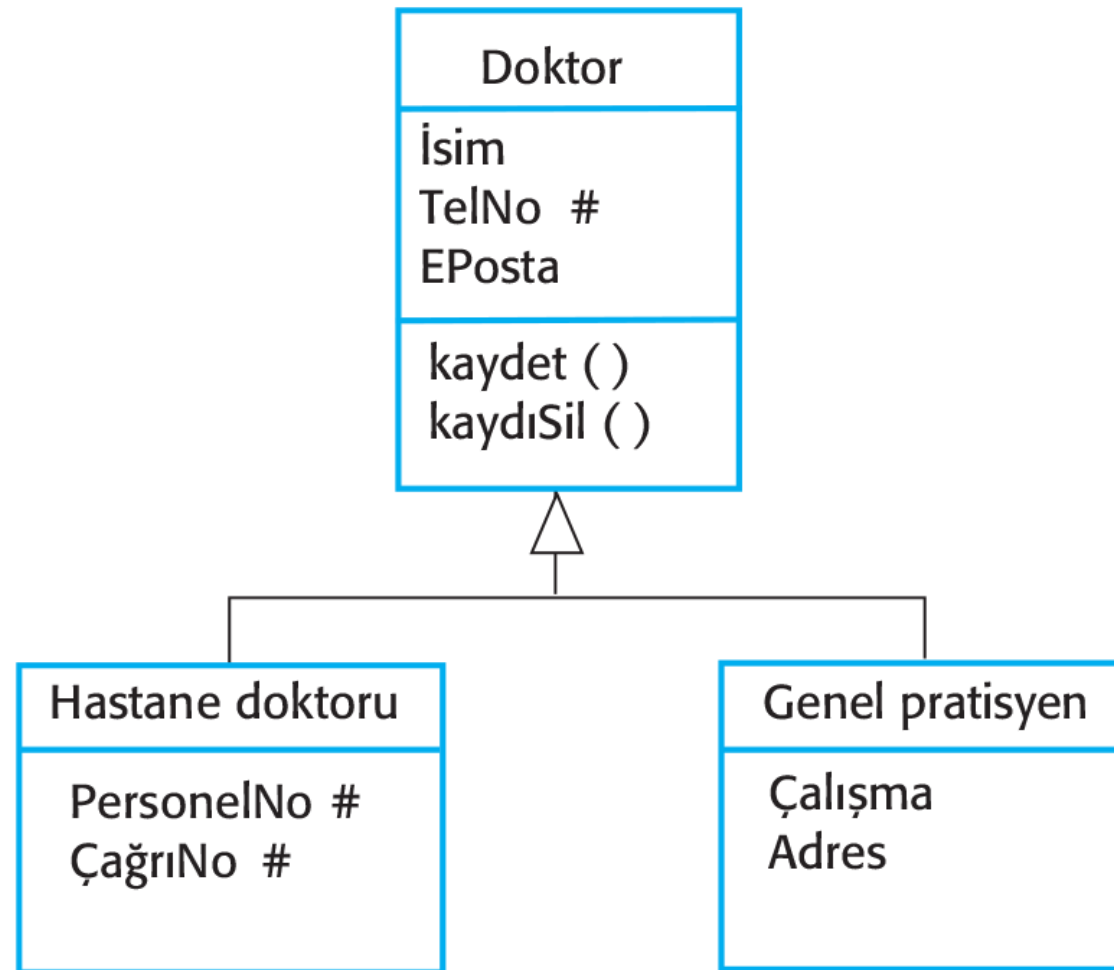
Genelleştirme - Generalization

- Sistemleri modellerken genelleme ve sınıf üretme imkanlarının olup olmadığını
- belirlemek yararlıdır.
- Bu ortak bilgilerin tek bir yerde tutulacağı anlamına gelir.
- Bu özellik uygulamada miras alma veya kalıtım (inheritance) tekniği ile sağlanır.



Şekil 5.11 Bir genelleştirme hiyerarşisi

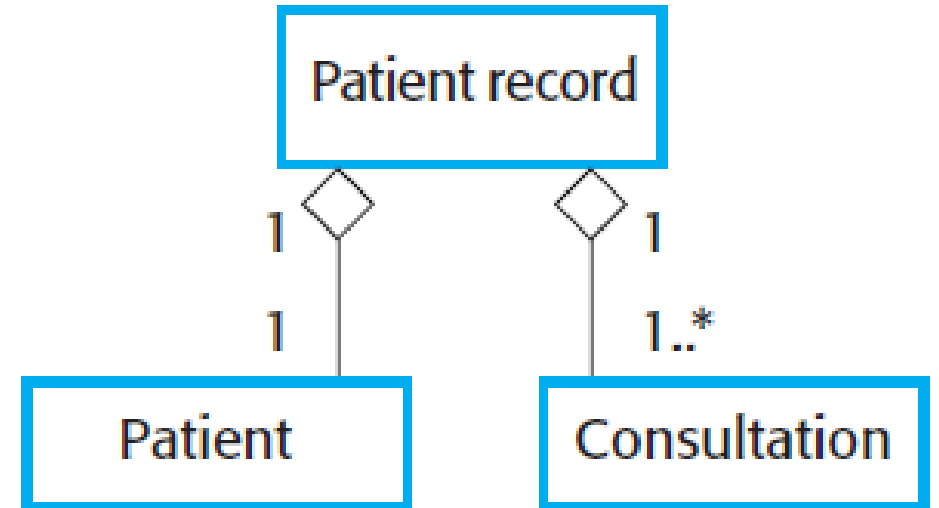
Şekil 5.12
Detaylandırılmış bir
genelleştirme hiyerarşisi



YAPISAL MODELLER - Structural Models

Birleştirme - Aggregation

- Gerçek dünyadaki nesneler genellikle farklı parçalardan oluşur.
 - Örneğin bir dersin çalışma paketi bir kitap, PowerPoint yansılar, sınavlar ve okuma tavsiyelerinden oluşabilir.
 - UML sınıflar arasında birleştirme adı verilen özel bir bağ türü sunar ki bu bir nesnenin (bütün) başka bazı nesnelerden (parçalar) meydana gelmesi anlamındadır.
 - Bunun için elmas şekli kullanılır.
-
- Hasta Kaydı, hasta ve sınırsız sayıda Konsültasyonlardan oluşmaktadır.
 - Kayıt hem özel hasta bilgisini hem de doktorlarla olan her bir konsültasyonun bilgisini saklar.



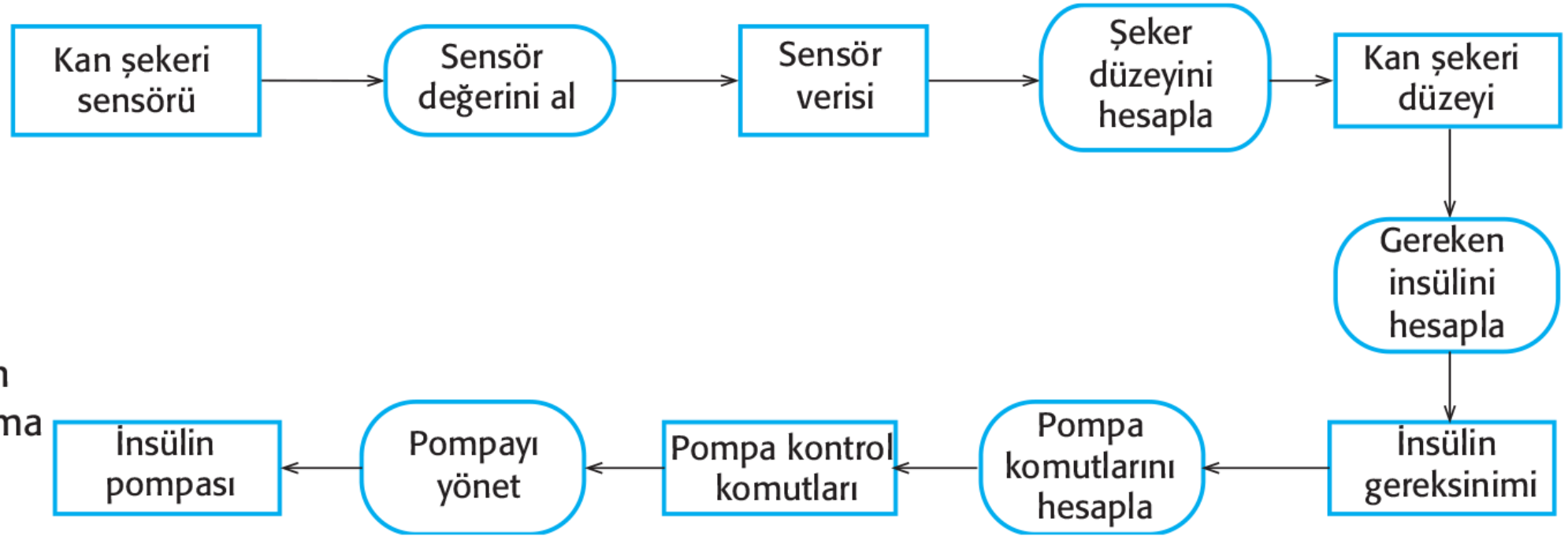
DAVRANIŞSAL MODELLER

Davranışsal modeller bir sistemin işleyişi sırasındaki dinamik davranışının modelleridir. Sistem, çevresinden gelen bir uyarana tepki verdiğinde ne olduğunu veya ne olması gerektiğini gösterirler. Bu uyarılar veri veya olaylar olabilir:

1. Sistem tarafından işlenecek olan veri erişilebilir hale gelir. Verinin mevcudiyeti işlemi tetikler.
2. Sistemin işleyişini tetikleyecek bir olay olur. Olaylar kendilerine bağlı verilere de sahip olabilirler fakat buna her zaman rastlanmaz.

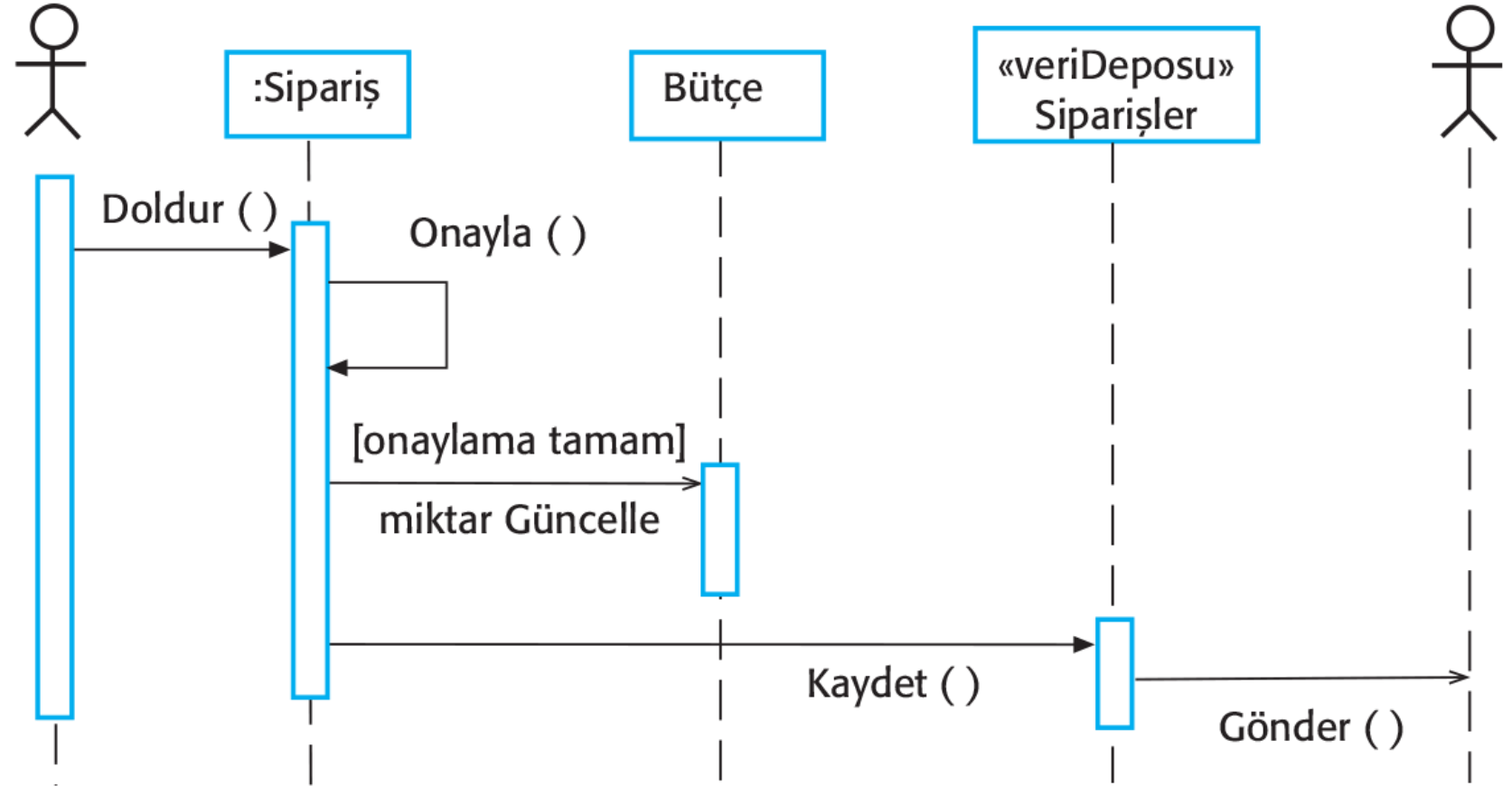
Pek çok iş sistemi veri tarafından güdülen veri-işleme sistemleridir. Göreceli az sayıda dış olay işleme gereksinimi ile birlikte genellikle sisteme girilen veri tarafından kontrol edilirler. İşlevleri veri üzerinde bir dizi etkinlik sonrasında çıktı üretmeyi içerir.

Şekil 5.14 İnsülin pompasının çalışma biçimini gösteren etkinlik modeli

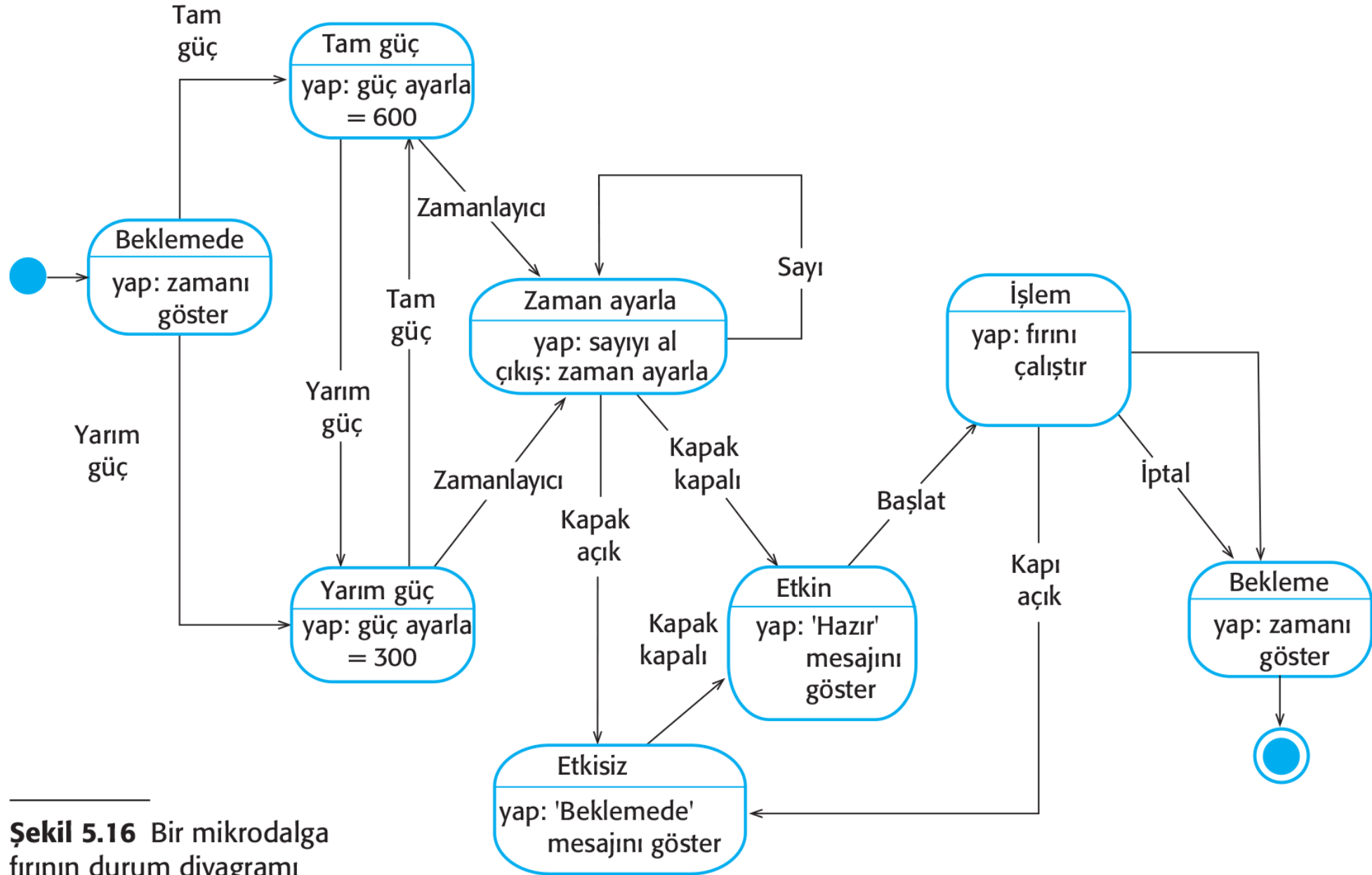


Satın alma sorumlusu

Tedarikçi



Şekil 5.15 Sipariş
işleme



Şekil 5.16 Bir mikrodalga fırının durum diyagramı

Durum	Durum	Açıklama
Bekle	Bekleme	Fırın, girdi beklemekte. Gösterge paneli saati gösteriyor.
Yarı	Yarım güç	Fırın 300 watt'a ayarlı. Gösterge paneli "Yarım güç" gösteriyor.
Tam	Tam güç	Fırın 600 watt'a ayarlı. Gösterge paneli "Tam güç" gösteriyor.
Zaman	Zaman ayarı	Pişirme zamanı kullanıcının girdiği değere ayarlanır. Gösterge paneli seçilen pişirme zamanını gösterir ve ayarlama sırasında güncellenir.
Engelli	Engelli	Fırının çalışması güvenlik için engellendi. İç fırın ışığı açık. Gösterge panelinde "Hazır değil" yazıyor.
Etkin	Etkin	Fırın etkin halde. İç fırın ışığı kapalı. Gösterge panelinde "Pişirmeye hazır" yazıyor.
Pişi	Pişirme	Fırın pişirme konumunda. İç fırın ışığı açık. Gösterge paneli zamanlayıcının geri sayımını gösteriyor. Pişirme bittiğinde 5 saniye boyunca zil çalar. Fırın ışığı yanar. Zil çalarken gösterge panelinde "Pişirme tamamlandı" yazar.
Uyarı	Uyaran	Açıklama
Yarı	Yarım güç	Kullanıcı yarım güç düğmesine bastı
Tam	Tam güç	Kullanıcı tam güç düğmesine bastı
Zaman	Zamanlayıcı	Kullanıcı zamanlayıcı düğmelerinden birine bastı
Sayı	Sayı	Kullanıcı sayısal bir tuşa bastı
Kapak	Kapak açık	Fırın kapağı düğmesi kapalı değil
Kapak	Kapak kapalı	Fırın kapağı düğmesi kapalı
Başlat	Başlat	Kullanıcı Başlat düğmesine bastı
İptal	İptal	Kullanıcı iptal düğmesine bastı.

Şekil 5.18
Mikrodalga fırın için durumlar ve uyarılar.

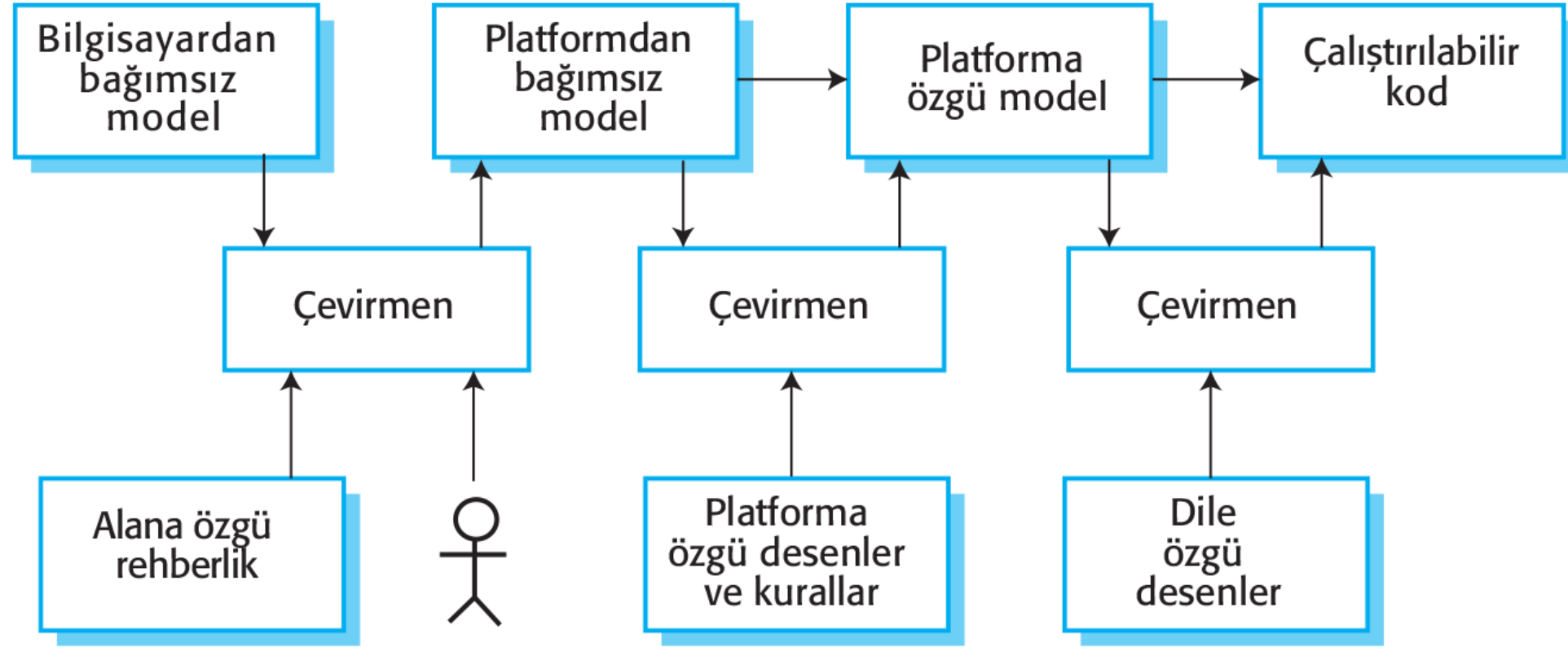
MODEL GÜDÜMLÜ MİMARİ

Model güdümlü mimari, UML modellerinin bir alt kümesini kullanarak bir sistemi betimleyen, model odaklı bir yazılım tasarım ve gerçekleştirim yaklaşımıdır. Burada farklı soyutlama derecelerindeki modeller yaratılır. Prensipte olarak yüksek düzeyli platformdan bağımsız bir modelden insan eli değmeden çalışan bir program oluşturmak mümkündür.

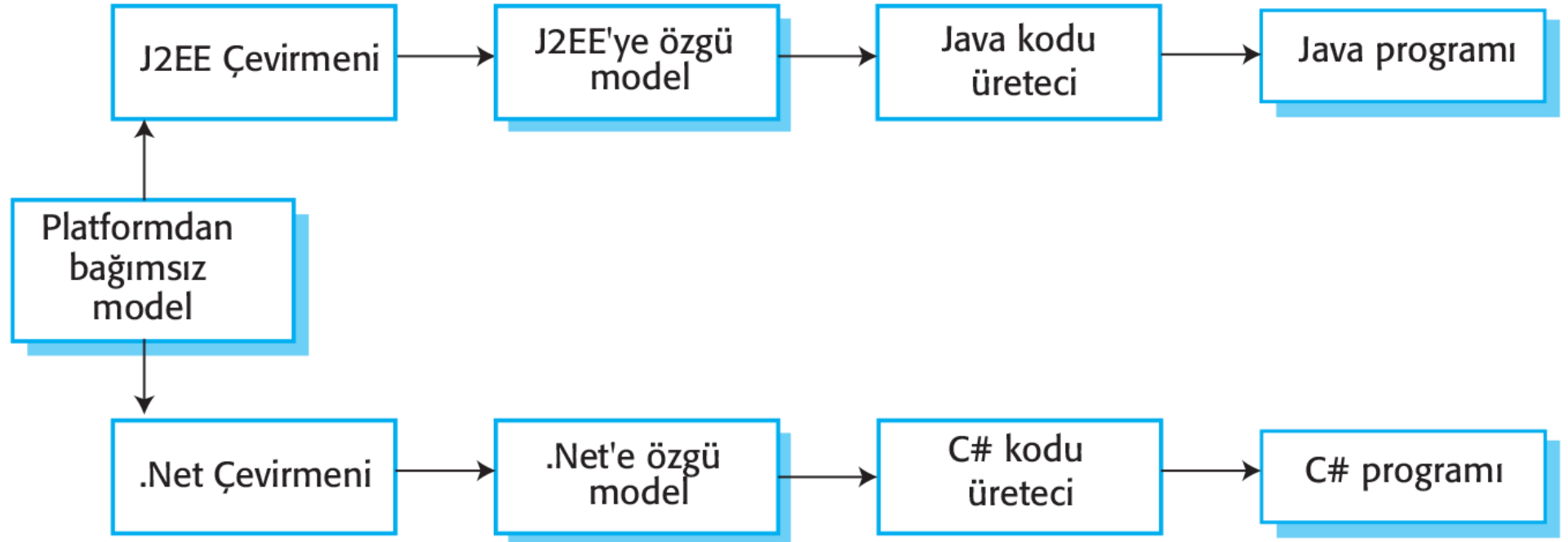
MDA metodu üç çeşit soyut sistem modelinin üretilmesini tavsiye eder:

1. Bir bilgisayardan bağımsız model (BBM)
2. Bir platformdan bağımsız model (PBM)
3. Bir Platforma özgü model (PÖM)

Şekil 5.19 MDA dönüşümleri



Şekil 5.20 Çoğul platforma özgü modeller





Executable UML

Model güdümlü mühendisliğin arkasındaki asıl fikir modellerden koda tam otomatik dönüşümün mümkün olabilmesidir. Buna ulaşabilmek için şekilsel modelleri, anlamları açık biçimde tanımlanmış ve çalıştırılabilir koda derlenebilecek biçimde kurabilmek gereklidir. Ayrıca şekilsel modellere bu modellerde tanımlanan işlemlerin nasıl gerçekleştirildiği hakkında bilgiler eklemek için bir yöntem ihtiyacı vardır. Bu, UML 2'nin çalıştırılabilir UML veya xUML adı verilen bir alt kümesini kullanarak mümkündür (Mellor ve Balcer 2002).