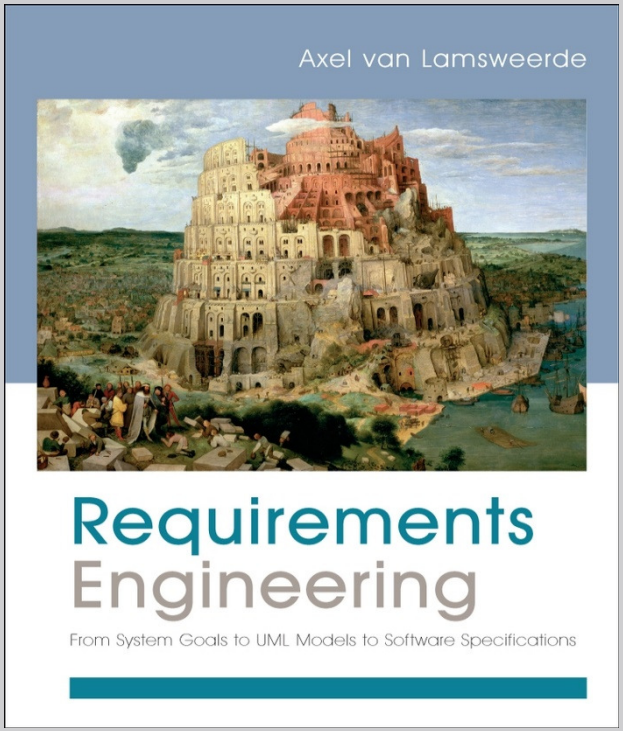
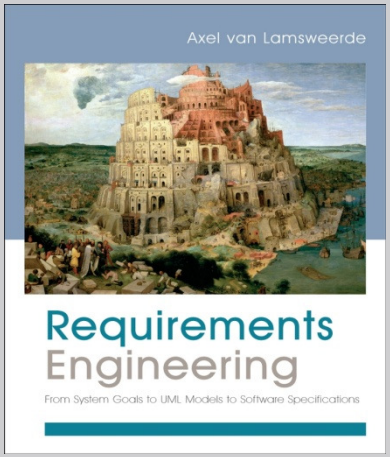
Requirements Engineering

From System Goals to UML Models to Software Specifications



Axel Van Lamsweerde



Gereksinimler Mühendisliğinin (GM) Temelleri

Bölüm 1

Gereksinimler Mühendisliği (GM) Dersine Giriş

Gereksinimler Mühendisliği nedir? Gerçek dünya problemine ait problemin makine çözümü gerçekleştirilir

$\Rightarrow$ real world problem and its machine solution

GM nin kapsamı Çözülecek probleme ait Niçin(WHY), Ne (WHAT) ve Kim (WHO) sorularına cevap aranır.

GM çalışmalarında çok kullanılan ifadeler: betimsel –tanımlayıcı (descriptive) ve kurallı (prescriptive) deyimler (cümleler)

Gereksinimin belirlenmesinde temel 2 kategori: fonksiyonel gereksinimler ve fonksiyonel olmayan gereksinimler

Gereksinimlerin yaşam döngüsü: aktörlerin gerçekleştirdikleri prosesler (süreçler) sonunda yazılım ürününün analizi gerçekleşir.

Problem Dünyası ve Makine Çözümü
Problem World and Machine Solution

- Geliştirilen bir yazılımın doğru ve kaliteli bir ürün olduğundan emin olmak için, öncelikle gerçek dünya problemi doğru çözülmüş olmalıdır. Bu da problemi ayrıntılı olarak **anlamak** ve **tanımlamak demektir**.
  - Gerçek dünyada çözülecek problemin gereksinimleri ne (what) sorusunun cevabıdır.
  - Bu cevaptan ortaya **içerik(** context) çıkar.
- Örnek: araba kontrolü

  - Problem: el freninin indirilmesi bazı durumlarda istenilen şekilde gerçekleşemeyebilir
  - İçerik (Context): araba sürme, fren, sürücünün amacı, güvenlik, kurallar…….

Problem Dünyası ve Makine Çözümü
Problem World and Machine Solution

- Problem Dünyası : Gerçek dünyanın çözülmesi istenen problem parçasının ifadesidir ve aşağıdaki bileşenlerle tanımlanır.
  - Beşeri Bileşenler (human components): organizasyon birimleri, personel(staff), operatorler, … VE
  - Fiziksel Bileşenler: aygıtlar, eski yazılım, doğa ,…….

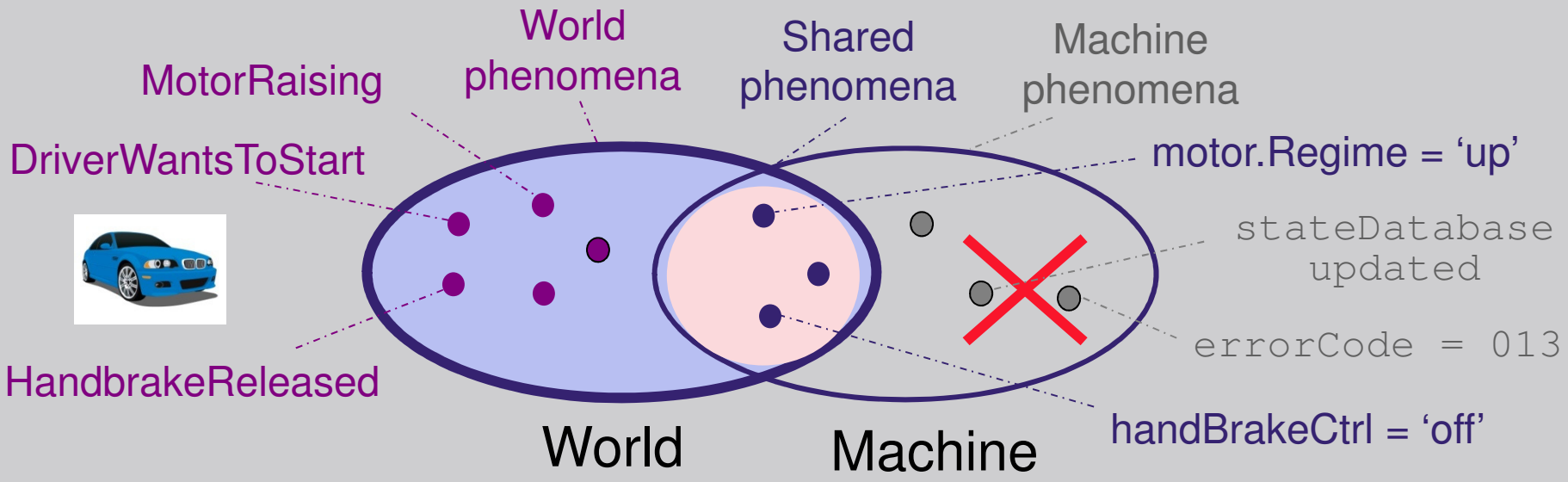
  içerir.
- Makine Çözümü : Problemin çözümü için düzenlenmesi gerekenler, yazılımın geliştirilmesi için gerekli gereksinimler belirlenir. Bunlar şöyle özetlenebilir:
  - Geliştirilmesi ve /veya satın alınması gereken yazılım
  - Donanım/yazılımın implementasyonun sağlandığı platform ,
  - Ortak giriş/çıkış aygıtları (sensorler-vericiler & çalıştırıcılar-alıcılar (actuators)

Problem Dünyası ve Makine Çözümü
Problem World and Machine Solution

- Gereksinimler Mühendisliği çalışmalarında başlangıçta 2 temel çalışma gerçekleştirilir.
  - Makinenin istenilen çözümünün **problem dünyasına etkisi belirlenir.**
  - Problem dünyası ile ilgili **varsayımlar** (assumptions) ve bununla ilgili çeşitli özellikler (relevant properties) tanımlanır

Problem Dünyası ve Makine Çözümü

- Dünya ve makine sözcüklerinin kendilerine ait olguları vardır.
- GM **dünya** olgusu (world phenomena) ile ilgilidir ve paylaşılan olguları (shared phenomena) içerir.
  - Oysa yazılım tasarımı machine phenomena (makine fenomeni) ile ile ilgilidir.



ÖRNEK:

Ödemenin gerçekleştirilmesi ile ürünün teslim edilmesi probleminin GM çözümüne ait bir özelliğin (bir deyimin) problem dünyası ve makine çözümü gösterimi

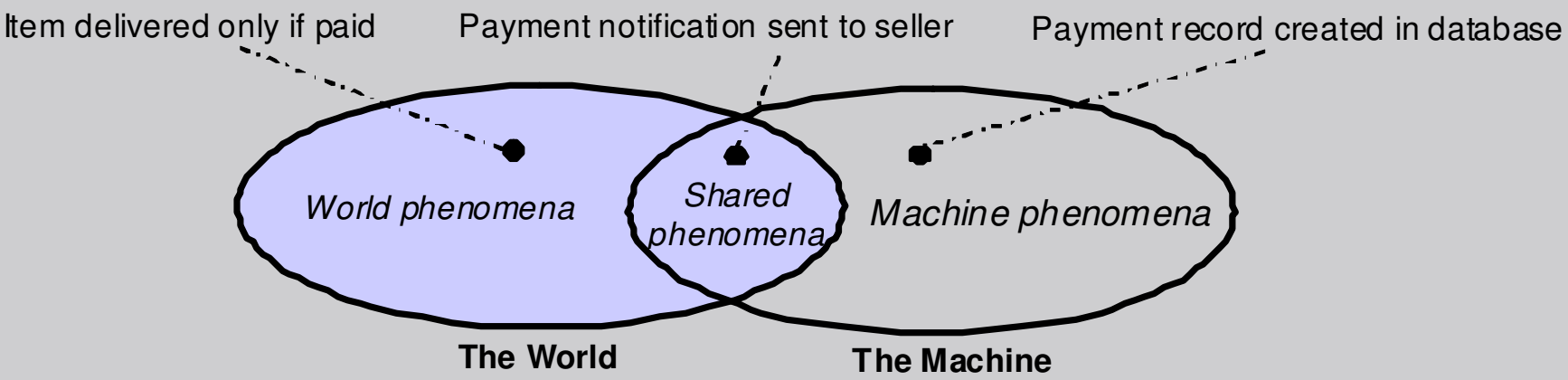


Figure 1.1 - The problem world and the machine solution

Yukarıdaki çözüme ait olası problem ifadesi :

Ödeme gerçekleştirildikten sonra satıcıya bir bildirim gönderilmelidir.

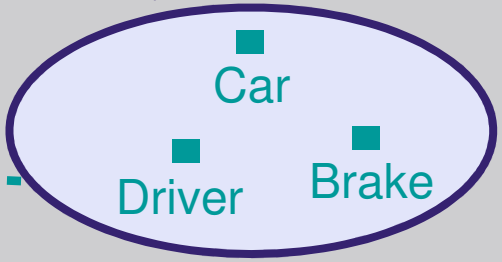
Problem Dünyasına ait Sistem Betimlemeleri System-as-is ve System-to-be

- System: Problem Dünyasına oluşturulmuş etkileşim halindeki bileşenlerin kümesidir. İki küme ile betimlenir:

i) System-as-is: Makine Çözümü oluşturulmadan önce sistemin içerdikleridir (mevcut sistem olarak yorumlanır)

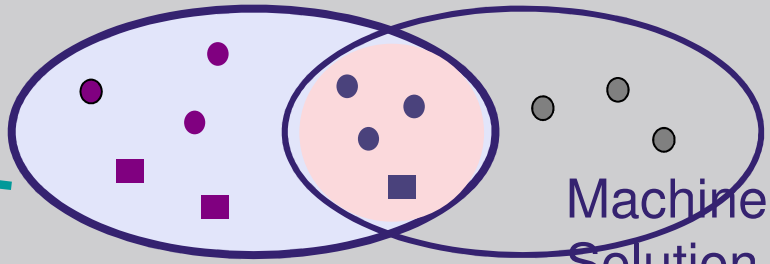
ii) System-to-be: Makine Çözümünün sistem çalıştırıldığında gerçekleştirecekleridir (istenilen çözümdür)

Araba fren sistemi :
kavramlar (concepts)
olgular (phenomena)
kuralların (rules)
betimlenmesi ile problem tanımlanır.



System-as-is

Otomatikleştirilmiş araba fren sistemine ait kavramlar , olgular ,kurallar



System-to-be

Gereksinimler Mühendisliğinin Tanımı

Birbiri ile bağlantılı, (eşgüdümlü) aktiviteler dizisi olarak tanımlanır.

Yazılım yoğun bir sistem üzerinde hedefler (objectives), yetenekler(capabilities), nitelikler (qualities) , kısıtlamalar (constraints) ve varsayımların (assumptions) **araştırılması** (exploring), **değerlendirilmesi** (evaluationg), **belgelenmesi** (documenting), **pekiştirilmesi -sağlamlaştırılması** (consolidating) , **düzeltilmesi** (revising) ve **uyarlanması** (adapting) gerçekleştirilir.

system-as-is ile tanımlanmış problemler yeni teknolojilerle sağlanan fırsatlarla (opportunities) çözümlenir.

Gereksinimler Mühendisliğinin Farklı Tanımlamaları...

Ross'77

- Gereksinimler tanımı mutlaka aşağıdakileri vurgulamalıdır.
  - Mevcut ve gelecek (tahmini) koşullara göre niçin (WHY) yeni bir sisteme ihtiyaç vardır ?
  - **Hangi** (WHAT) sistem özellikleri bu bağlamı sağlayacaktır ?
  - Sistem Nasıl (HOW) oluşturulacaktır?

Zave'97

- Gereksinimler gerçek-dünyanın **amaçları, fonksiyonları**, yazılım sistemindeki **kısıtları** ile ilgilidir ve
  - Yazılım davranışının kesin olarak belirlenebilmesi için bu kavramlarla bağlantı kurar
  - Zaman içerisinde bu kavramların evrimleşmesi -gelişmesi (evolution) sağlanır

Sistem Gereksinimleri (System Requirements)

- Software-to-be:(Geliştirilecek yazılım makinenin parçasıdır ve system-to-be 'nin bileşenidir.
- Environment (çevre) : system-to-be 'nin diğer tüm bileşenlerini insan (people), aygıtlar(devices) , önceden mevcut yazılımı (pre-existing software) içerir.
- System Requirements: çevredeki (environments) olgulara göre belirlenmiş olan **system-to-be NELERİ** karşılamalıdır?

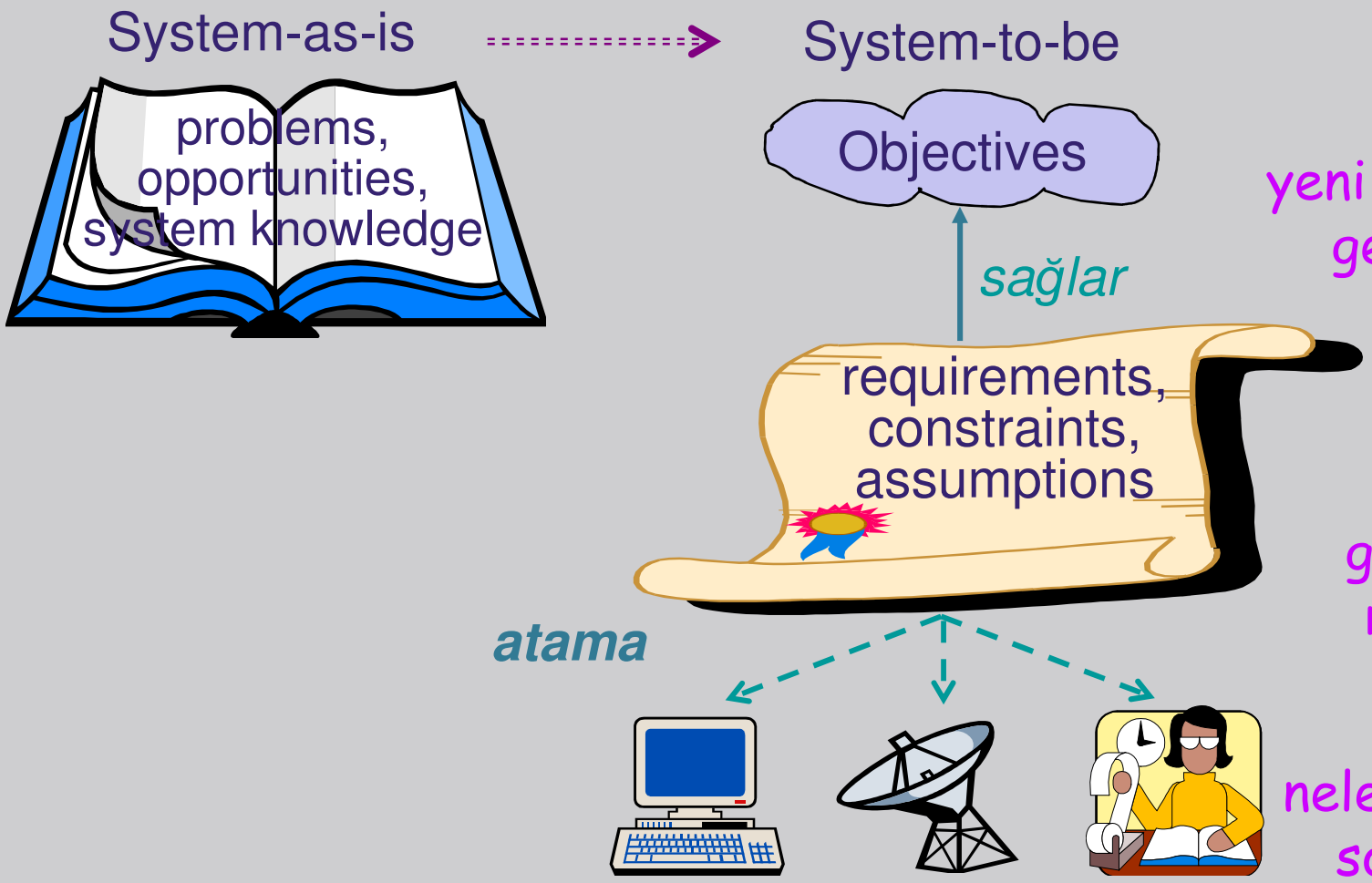
"El freni, sürücü hareket etmek istediğinde indirilmelidir.."

Yazılım Gereksinimleri (Software Requirements)

- Software Requirements: : Yazılım ve çevresi tarafından paylaşılan (shared) olgular cinsinden belirlenmiş olan **software-to-be** kendi başına **NELERİ** karşılamalıdır?

"Yazılım çıktı değişkeni olan **handBrakeCtrl** 'ın değeri , yazılım girdi değişkeni **motorRegime** değeri arttığında **kapalı** (off) olacaktır"

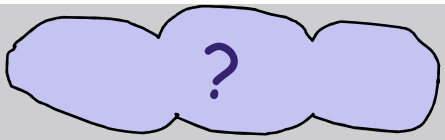
Gereksinimler Mühendisliğinin Kapsamı
Niçin (Why) , Ne(What) , Kim(Who) Boyutları



Niçin
yeni bir sisteme
gereksinim
vardır?

Ne
hizmetleri
gerçekleşti-
rilecektir?

Kimler
neler yapmakdan
sorumludur?

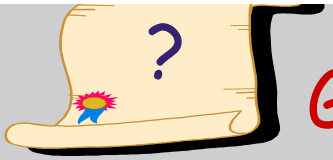


Gereksinimler Mühendisliğinin NİÇİN boyutu

- **System-to-be**'nin amaçlarının tanımlanması, analiz edilmesi ve gereksiz bilgilerden temizlenmesi (saflaştırılması) gerçekleştirilir
  - **system-as-is** 'in analiz edilmiş eksikliklerinin belirlenmesi
  - Ticari (işe ait -business) amaçlar ile uyum sağlanması
  - Teknolojik olanaklardan yararlanılması
- Örnek: havaalanı tren kontrol sistemine ait system-to-be şöyle tanımlanabilir.

  "Daha fazla yolcuya hizmet edebilmek"

  "Terminaller arasında transfer süresini azaltmak"
- Güçlükler
  - Alan bilgisinin (domain knowledge) elde edilmesi kolay değildir.
  - Alternatif seçeneklerin değerlendirilmesi (yani aynı amacı sağlamak için alternatif yolların sağlanması) kolay değildir.
  - Çelişen amaçlarla başa çıkmak kolay değildir.



Gereksinimler Mühendisliğinin NE Boyutu

- system-to-be'nin **fonksiyonel hizmetlerinin** tanımlanmasıdır. Bunlar yazılım hizmetlerini oluştururlar.
  - Tanımlanmış amaçların sağlanması (gerçekleşmek üzere tanımlanması)
  - Nitelik kısıtlarının güvenlik , performans,.... sağlanması
  - Çevre ile ilgili gerçekçi varsayımların probleme ilavesi
- Örnek: havaalanı tren kontrolü

  "trenin güvenli hızlanmasının hesaplanması"

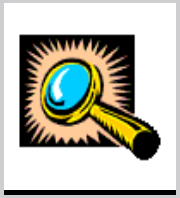
  "trendeki yolculara yararlı bilgilerin gösterilmesi"
- Güçlükler
  - Özelliklerin tanımlamalarının doğru yapılması kolay değildir.
  - Bunların parçaların tümünde açık olarak anlaşılabilirliği kolay sağlanamaz.
  - Sistemin amaçlarının geriye doğru izlenebilirliğini sağlamak güçtür.

?Gereksinimler Mühendisliğinin KİM Boyutu

- **System-to-be** bileşenlerini oluşturan amaçlar, servisler, kısıtlardan sorumlu olanların belirlenmesi gerekir.
  - Böylece yazılım-ortamının (software environment) sınırları belirlenecektir.
- Örnek: havaalanı tren kontrolü
  - "Trenin güvenli hızlanması" ... software-to-be 'nin doğrudan sorumluluğu altında (sürücüsüz seçenek) *veya* sürücünün yazılım işaretlerini takip etmesi olarak tanımlanabilir.
  - "Trenin hızı/pozisyonunun tahmini doğruluğu… ... İzleyen sistemin sorumluluğunda ***ya da*** önceki trenin sorumluluğunda tanımlanabilir.
- Güçlükler
  - Doğru otomasyon derecesine karar vermek için alternatif seçeneklerin değerlendirilmesi kolay değildir.

GM'de Deyimler (Statements)

- GM 'inde kullanılan deyim (statements) tipleri :



tanımlayıcı (descriptive)

öngörmeli - kurallı (prescriptive)

Gereksinimlerin sınıflandırması

fonksiyonel *ve* fonksiyonel olmayan gereksinimler

GM'de Deyimler (Statements)

- Tanımlayıcı –Niteleyici (Descriptive) deyimler sistemin nasıl bir davranış gösterdiğine bağlı olmadan sistem özelliklerini içerir (bildirme kipi - indicative mood)
  - Doğal kanun (natural law) , fiziksel sınırlayıcılar (physical constraint) vs.
  - " Trenin kapıları kapalı ise , açık değildir "
  - "Trenin hızı pozitif ise, hızı boş değildir"
- Kurallara bağlı (Prescriptive) deyimler olması istenilen özellikleri içerir ve sistemin davranışının nasıl olduğuna bağlı değildir. (istek kipi- optative mood)
  - "Tren hareket halinde iken kapılar daima kapalı kalacaktır"

GM 'de Kurallı (prescriptive) ve Tanımlayıcı (descriptive) Deyimler arasındaki önemli fark

- Kurallı deyimler (prescriptive statements)

  ihmal edilebilir (negotiated),

  daha farklı zayıf cümlelere dönüştürülebilir(weakened),

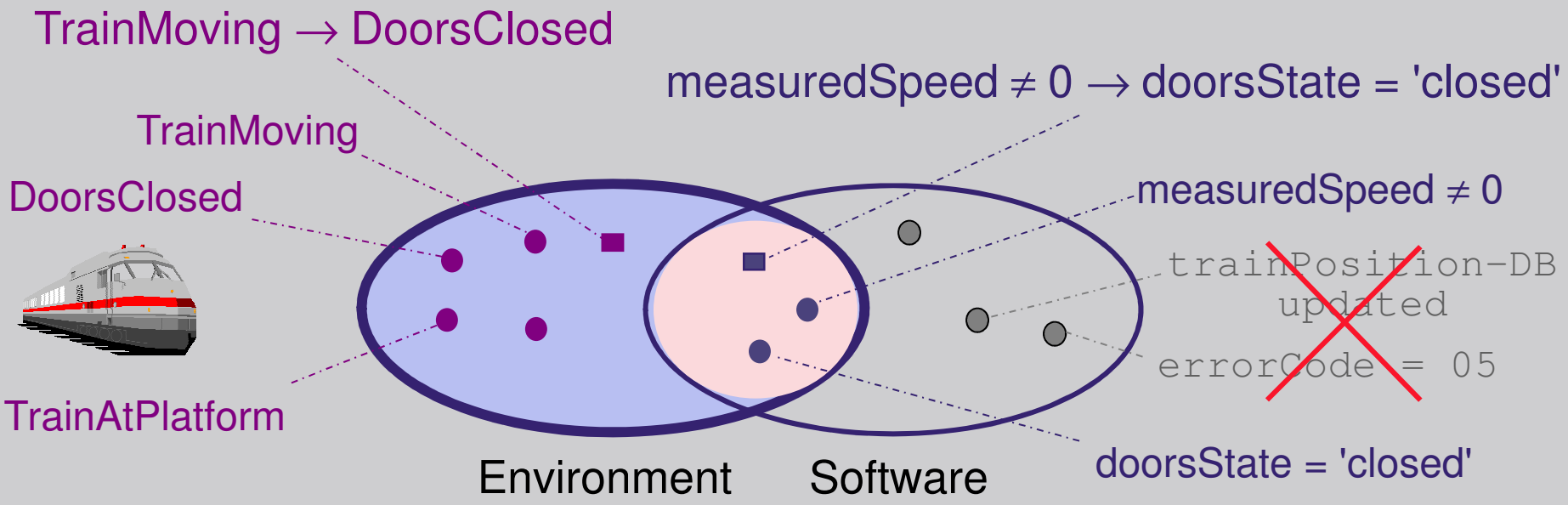
  başka cümlelerle değiştirilebilir (replaced by alternatives)

- Tanımlayıcı – niteleyici deyimler (descriptive statements) yukarıda yapılabilenlerin hiç birini gerçekleştiremez

Deyimler durumlarına göre değişebilir (Statements may differ in scope)

- Bir GM deyimi bir olguya (duruma) referans vermelidir. Bu:
  - Çevresel olabilir (owned by the environment) ya da
  - Çevre ve software-to-be arasında olabilir

  Biri ortaya çıkan olayı (phenomena) kontrol altında tutarken , diğeri ekranda görüntüler.



Diğer Deyim tipleri
Sistem Gereksinimleri, Yazılım Gereksinimleri

- Sistem Gereksinimi: : Kurallı (*prescriptive*) *deyimdir .*
  Paylaşılmış olmasına gerek yoktur ve çevre olayına işaret eder.
  - software-to-be yapısı ile birlikte diğer sistem bileşenlerini kullanır yürütür.
  - Sözcüksel olarak ifade edildiğinde tüm ayrıntılar verilmiş olur.

$$\text{TrainMoving} \rightarrow \text{DoorsClosed}$$

- Yazılım Gereksinimi : Kurallı (*prescriptive*) *deyimdir.* Paylaşılan olaylara işaret eder.
  - Sadece software-to-be tarafından yürütülür
  - Yazılı geliştiricilerin ifadeleri ile belirlenir.

$$\text{measuredSpeed} \neq 0 \rightarrow \text{doorsState} = \text{'closed'}$$

Diğer Deyim tipleri
Sistem Gereksinimleri, Yazılım Gereksinimleri

- Sistem Gereksinimi: : Kurallı (*prescriptive*) *deyimdir .*
  Paylaşılmış olmasına gerek yoktur ve çevre olayına işaret eder.
  - software-to-be yapısı ile birlikte diğer sistem bileşenlerini kullanır yürütür.
  - Sözcüksel olarak ifade edildiğinde tüm ayrıntılar verilmiş olur.

$$\text{TrainMoving} \rightarrow \text{DoorsClosed}$$

- Yazılım Gereksinimi : Kurallı (*prescriptive*) *deyimdir.* Paylaşılan olaylara işaret eder.
  - Sadece software-to-be tarafından yürütülür
  - Yazılı geliştiricilerin ifadeleri ile belirlenir.

$$\text{measuredSpeed} \neq 0 \rightarrow \text{doorsState} = \text{'closed'}$$