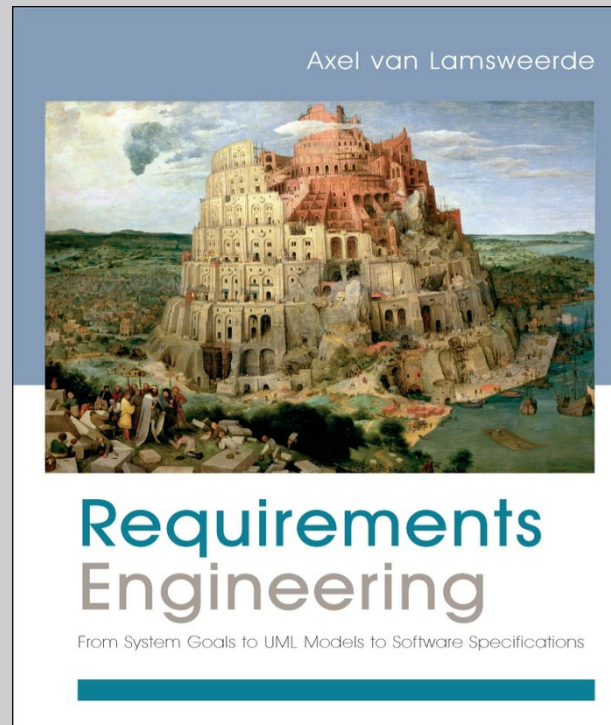
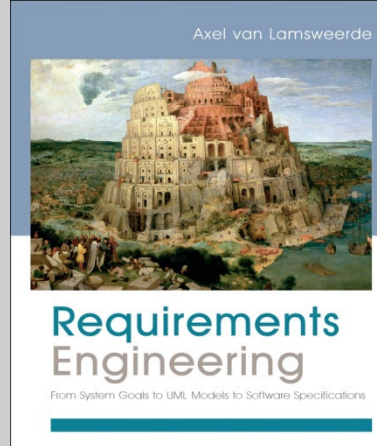


Requirements Engineering

From System Goals
to UML Models
to Software Specifications



Axel Van Lamsweerde



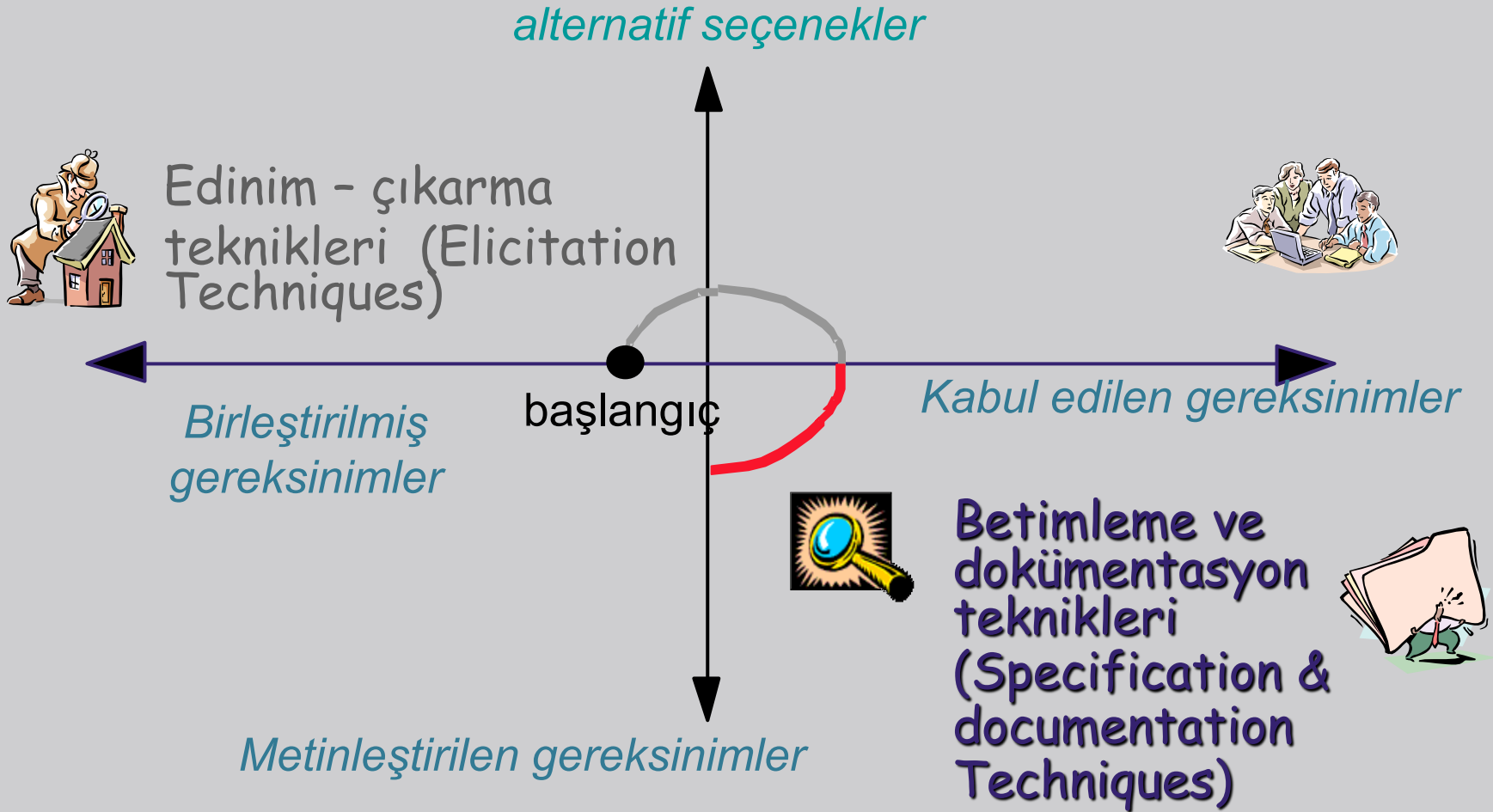
Gereksinimler Mühendisliğinin Temelleri

Bölüm 4

Gereksinimlerin Betimlenmesi ve Dokümantasyonu

Requirements Specification
& Documentation

RE ürünleri ve süreçleri (RE products and processes)



Diyagramların Kullanılması (UML- Devam)

Biçimsel Betimlemeler (Kavramsal Modelleme)

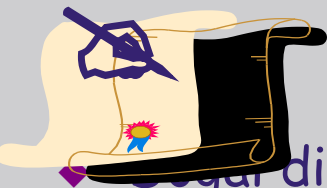


Gereksinimlerin Betimlenmesi ve Dokümantasyonu

Gereksinimlerin betimlenmesi ve dokümantasyonuna ilişkin 4 farklı yaklaşım incelenecektir.

- ◆ Herhangi bir koşul olmaksızın (unrestricted) gereksinimlerin dokümantasyonun doğal dilde (natural language) hazırlanması
- ◆ Belli kurallara uygun olarak yine doğal dilde gereksinimlerin disiplinli olarak dokümantasyonu
- ◆ Diyagram notasyonu ile gereksinimlerin belirlenmesi diğer bir ifade ile: Görsel Diyagramların Kullanımı
 - «use case» ve «sequence diyagramları» anlatıldı
 - Diğer görsel diyagramların incelenmesine devam edilecek.
- ◆ Biçimsel (formal) betimlemeler

Gereksinimlerin Belgelendirilmesinin Doğal Dilde Koşulsuz Gerçekleştirilmesi

- 
- ◆ Doğal dilde herhangi bir kısıtlama olmadan yazılmış gereksinim bildirimleridir (**statements**). Gereksinim olarak kabul edilen tüm deyimler ifade edilebilir.
 - ◆ **Avantajları** : Sınırsız ifade istenilenin diğer ilgililere iletimini kolaylaştırır, yazılması için de özel bir bilgi gerekmez.
 - ◆ **Dezavantajları:**
 - i) Problem dünyasının (sistem gereksinimleri - çevresel olgular) herhangi bir özelliği gereksinimlerde ifade edilmeyebilir. Ya da belirtilen amaç ya da gereksinim açık değildir. Ayrıca, bazı girdilerin yazılım karşılığı olmayabilir (**omission- ihmal**)
 - ii) Aynı gereksinimin farklı şekillerde ifade edilmiş olması belirsizlik (**ambiguity**) oluşturur.
 - iii) Herhangi bir istem ifadesi problem dünyasının özelliklerine uygun olmayabilir (**inadequacy**)
 - iv) Gereksinim ifadesi problem dünyasının özellikleri ile hiçbir bilgi vermez (**noise**) .
 - v) Herhangi bir gereksinimin beytimlenmesinde sorun yoktur. Fakat makine çözümü (software phenomena) olmayabilir(**overspecification**).

Yapılandırılmamış ve Doğal Dilde Betimlenmiş Gereksinim Bildirim Örneği

“Bir trende frenleme sistemi, herhangi bir hızlanma komutu alındığında **ya da** belirlenenden daha yüksek bir hızda trenin istasyon bloğuna girmesi ile **ve** öndeki trene belirlenmiş değerden daha yakın olması durumunda etkinleştirilecektir»

Bu kritik **güvenlik gereksinimi** iki farklı şekilde şöyle yorumlanabilir:

Tren istasyona çok hızlı girdiği durumda

i) Tam fren sistemi eski bir komut alındığında **ya da** öndeki tren çok yaklaştığında aktive edilecektir.

if Case1 then <Statement1> F1
or if Case2 then <Statement2>

Bu bildirim hatalı sonuç verir.

ii) Tam fren sistemi **sadece** önceki tren çok yaklaştığında etkinleştirilecektir.

Doğru bildirim If Case1 then <Statement1> **F2**
 and if Case2 then <Statement2> olmalıdır.

Yapılandırılmamış (Unstructured) Gereksinim Bildirim Örneği

- ◆ F1 bildirimi önermeler mantığı (propositional logic) ile yazıldığında

$(\text{not Case1 or Statement1}) \text{ or } (\text{not Case2 or Statement2})$

ifadesi

$\text{not (case1 and case2) or Statement1 or Statement2}$

$\text{not false or Statement1 or Statement2}$ önermesi **True** ile sonuçlanır.

F1 bildirimi, gereksinim bildiriminin (problem ifadesinde verilen) tümünü sağlamadığı için tanımlamadaki bazı koşullar sağlanmayacaktır.

Bu da problemin çözümü sırasında problem oluşturabilir.

Laws of Propositional Logic

De Morgan's laws	$\neg(p \vee q) \equiv \neg p \wedge \neg q$	$\neg(p \wedge q) \equiv \neg p \vee \neg q$
Absorption laws	$p \vee (p \wedge q) \equiv p$	$p \wedge (p \vee q) \equiv p$
Conditional identities	$p \rightarrow q \equiv \neg p \vee q$	$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

Yapılandırılmamış (Unstructured) Gereksinim Bildirim Örneği

- ◆ F2 bildirimi önermeler mantığı (propositional logic) ile yazıldığında

(not Case1 or Statement1) and (not Case2 or Statement2)

ifadesi

not (case 1 and not statement 1) and not (Case 2 and not
Statement2)

(not Case 1 or not not Statement 1) and

(not Case 2 or not not statement 2)

not (Case 1 or Case 2) or (Statement1 or Statement 2)

(Case 1 or case 2) and (Statement1 or Statement2)

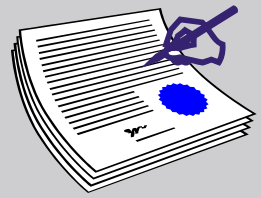
(Case 1 and Statement1) or (Case2 and Statement2)

önermesi **True** ile sonuçlanır.

Yapılandırılmamış Gereksinim Bildirim Örneği

- ◆ (Case 1 and Statement1) or (Case2 and Statement2) olarak elde edilen F2 bildiriminin sonucu da problemin gereksinim ifadesini tümüyle simgelememektedir.
- ◆ Çünkü sonuç olarak elde edilen or bildiriminin sadece tek tarafının gerçekleşmesi ifadenin T sonucu vermesi için yeterlidir.
- ◆ Bu da problem için tanımlanan
case 1: herhangi bir hızlanma komutu alınması
case 2: belirlenenden daha yüksek bir hızda istasyon bloğuna girilmesi
case 3: öndeki trene belirlenmiş değerden yaklaşma önermelerinden YA case 1 ya da case 2 True ise gereksinim sonucu T olacaktır; YA DA case 3 True ise gereksinim sonucu T olacak , fren sistemi devreye girecektir.

BU da ya sadece trenlerin yaklaşması koşulu ile ya da bu koşul hiç gerçekleşmeden gereksinim ifadesinin doğru sonuçlanması demektir.
Problem ifadesi ile ÇELİŞİR



Yapılandırılmış (Structured) olarak Doğal Dilde Gereksinim Bildirimi

- ◆ Yapılandırılmadan (unstructured) doğal dilde belgelendirilmiş gereksinimlerdeki problemlerin düzeltilmesi gerekir.

Bunun için öncelikle aşağıdaki tanımlamalar, **lokal ve global kurallar** olarak tanımlanır.

- ◆ **Lokal kurallar** deyimlerin doğal dilde nasıl yazıldığıdır.
- ◆ **Global kurallar** ise gereksinimler belgesinin nasıl düzenleneceğidir.

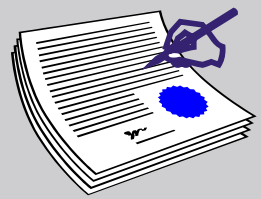
Gereksinimlerin Yazılmasında Lokal Kurallar

- ◆ Yeterli ve etkin teknik çözümlmeleri sağlayabilmek için kurallara uygun olarak, daha disiplinli bir gereksinim belgelendirmesi gerçekleştirilmesidir.
- ◆ Karar tabloları gereksinimlerin belgelendirilmesi için oluşturulabilir.
- ◆ Bunlar karmaşık kombinasyonların oluşturulması için **if-then** koşullarından ve **ön tanımlı deyim** şablonlarından genişletilerek elde edilir.

Doğal Dil Betimlemeleri için Biçimsel Kuralların Kullanımı

Teknik belgelerin hazırlanması için kullanılan standart ilkeler gereksinimler belgesinin hazırlanmasına uyarlanır.

- Bildirimi ifade etmeden önce ne olduğunun açıklanması
- Önce motivasyonun (ifade edilecek olan bildirimin) sağlanması, daha sonra özetlenmesi
- Her kavram kullanılmadan önce tanımlanmış olmalı
- Bildirimin uygunluğunun (relevant) , anlaşılabilirliğinin (comprehensible) yeterliliğinin (enough) kontrol edilmesi
- Bir cümlede hiçbir zaman birden fazla gereksinim, tahmin (assumption) ya da alan özelliği (domain property) olmaması.
- Cümlelerin kısa olması
- Gereksiz kısıtlamalar kullanılmamalı
- Soyut ifadeleri açıklığa kavuşturmak için akılcı örnekler kullanılmalı
- Parçalar arasındaki karmaşık ilişkiler için diyagram oluşturulması



Gereksinim Bildirimlerinin Kurallı Yazıldığı Gereksinim İfadesi

- ◆ Koşulların karmaşık kombinasyonu için karar tabloları (decision tables)
- ◆ Karar tabloları karmaşık if-then koşullarını yapılandıran standart bir tekniktir.

Girdi if-koşulları

Doğruluk değerlerinin girilmesi

Train receives outdated acceleration command	T	T	T	T	F	F	F	F
Train enters station block at speed $\geq X$ mph	T	T	F	F	T	T	F	F
Preceding train is closer than Y yards	T	F	T	F	T	F	T	F
Full braking activated	X		X		X			
Alarm generated to station computer	X	X	X	X				

Çıktı then-koşulları

bir durum= AND-kombinasyonu

- ◆ Sistemattir, oluşturulması basittir ve yorumlanmasında avantajları vardır

Tablonun ve Yorumlamanın Basitleştirilmesi

Tamlık Kontrolü (Completeness check): Tüm tablo için 2^N sütun gereklidir.

Tablonun basitleştirilmesi (Table reduction):

- i) Giriş koşullarının (input conditions) AND kombinasyonu, alan özelliklerini belirlemediği durumda o sütun göz önüne alınmayabilir.
- ii) Giriş koşullarının aynı çıktı kombinasyonları şeklinde sonuçlanması durumunda iki sütun birleştirilebilir. Örnekte birinci ve üçüncü sütunlar birleştirilebilir. Çünkü her iki sütunda ikinci koşullardan biri T diğeri F dir. Aynı doğruluk tablosu vermemektedir.

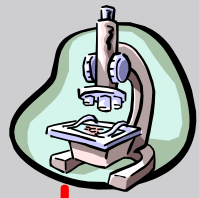
Karar tabloları (Decision Table) **KABUL TESTİ**ni (Acceptance Test) sağlar. Her sütun farklı bir **giriş-çıkış test verisidir**. Bu test **Kara Kutu** (Black-Box) sınıflandırmasına aittir.

Kabul Testinin Gereksinimler Mühendisliğindeki Önemi

Kabul testi (acceptance test) , bir sistemin gereksinimlerini karşıladığını doğrulamak için kullanılan kritik bir süreçtir.

Kabul testleri, müşterinin veya son kullanıcının sistemin gereksinimlerini karşıladığını onaylaması için yapılır. Müşteri, kabul testlerini geçmeden ürünü kabul etmez.

- ◆ **Gereksinimlerin Doğrulanması** : Belirlenen gereksinimlerin gerçek sistemde uygulanıp uygulanmadığı kontrol edilir.
- ◆ **Müşteri Memnuniyeti**: Müşterinin ihtiyaçlarının karşılandığı garanti edilir. Bu da müşteri memnuniyetini artırır.
- ◆ **Sürekli İyileştirme**: Gereksinimlerin ve süreçlerin sürekli olarak gözden geçirilmesine olanak tanınır. Bu, sistemin zamanla daha iyi hale gelmesini sağlar.

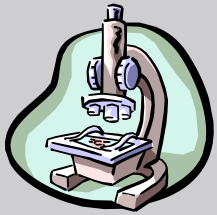


Gereksinimlerin Betimlenmesi ve Dokümantasyonu için Biçimsel Betimlemeler

Biçimsel Betimlemeler (Formal Specification) aşağıdaki gibi farklı betimlemelerle gerçekleştirilir.

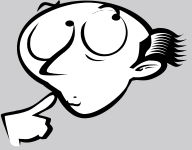
- Bildirimlerin gösterimi için temel mantıksal bilgiler (Logic as a basis for formalizing statements)
- Hikaye odaklı betimleme (History-based specification)
- Durum odaklı betimleme (State-based specification)
- Olay odaklı betimleme (Event-based specification)
- Cebirsel betimlemeler (Algebraic specification)





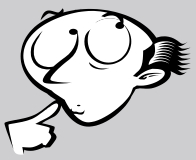
Biçimsel Betimleme Teknikleri (Formal Specification Techniques)

- ◆ Logic : bildirimlerin formüle edilmesinin temelidir
 - Propositional logic (önermeler mantığı)
 - First-order predicate logic (birinci dereceden yüklem mantığı)
 - First-order specification languages (birinci dereceden betimleme dilleri)



Biçimsel Betimleme Nedir ve Niçin Gereklidir?

- ◆ Biçimsel = makinede işlenebilen dil
 - Çoğunlukla matematiksel mantıkla ifade edilir.
 - Yeni bilginin sentaksı, semantiği ve çıkarım kurallarıdır
- ◆ Yararları
 - Bildirimin formüle edilmesinde daha yüksek kesinlik
 - Yorumlamada daha kesin kurallar
 - Daha karmaşık kontroller ve türetmelerin otomasyonu



Biçimsel Betimleme Niçin Gereklidir?



- ◆ Biçimsel betimleme doğal dildeki ve diyagram olarak verilen (görsel) betimlemeleri tamamlar.
- ◆ Özellikle kritik çözümler için önemlidir.
- ◆ Gereksinimlerin Dokümantasyonu 2 kısımda gerçekleştirilir
 - **Bildirim Bölümü (Declaration Part):** Parçanın yapısıdır (Örneğin diyagramlar...)
 - **Beyan kısmı (Assertion Part):** Parçanın özellikleri - (Örneğin «prescriptive», «descriptive» tanımlamalar...)
 - Bu tanımlamalarla büyük parça betimlemelerden (coarse grained) daha küçük parçacıklı (fine grained) betimlemelere geçiş mekanizmasının yapılandırılması sağlanır.



Önermeler Mantığı : Sentaks

Propositional logic: syntax

- ◆ Mantıksal sistem en basit biçimsel sistemdir.
- ◆ Önermeler, and ($\wedge$), or ($\vee$), not ($\neg$), implies ($\rightarrow$), equivalent to ($\leftrightarrow$) gibi mantıksal birleştiriciler ile rekürsif(özyineli) olarak birleştirilir.
 - İfade edilebilirlik sınırlıdır: Değişken içermez
Miktar ,çokluk (quantification) içermez

Dilbilgisi kurallarına uygun olarak well-formed bildirimler **sözdizim (sentaks) kurallarıdır.**



Önermeler Mantığı: İspat Teorisi

Propositional logic: proof theory

- ◆ Mevcut deyimlerde çıkarım kuralları (inference rules) ile yeni deyimler oluşturulması
 - **sound rule**: herhangi bir yorumlamaya (interpretation) göre eğer sonuç doğru ise, varsayım-önermeyi (premise) true yapar.

- ◆ Çıkarım kuralları (inference rules)

$$\frac{P \rightarrow Q, P}{Q}$$

$$\frac{P \rightarrow Q, Q \rightarrow R}{P \rightarrow R}$$

$$\frac{P \vee Q, \neg P \vee R}{Q \vee R}$$

- ◆ Yukarıdaki 3. kuralın uygulaması

$\neg \text{trainMoving} \vee \text{doorsClosed}, \text{trainStopped} \vee \text{trainMoving}$
kurallarından

$\text{doorsClosed} \vee \text{trainStopped}$ sonucu elde edilir.



TABLE 1 Rules of Inference.		
<i>Rule of Inference</i>	<i>Tautology</i>	<i>Name</i>
$\frac{p}{\therefore p \vee q}$	$p \rightarrow (p \vee q)$	Addition
$\frac{p \wedge q}{\therefore p}$	$(p \wedge q) \rightarrow p$	Simplification
$\frac{p}{q}$ $\frac{q}{\therefore p \wedge q}$	$((p) \wedge (q)) \rightarrow (p \wedge q)$	Conjunction
$\frac{\dot{p}}{p \rightarrow q}$ $\frac{p \rightarrow q}{\therefore q}$	$[p \wedge (p \rightarrow q)] \rightarrow q$	Modus ponens
$\frac{\neg q}{p \rightarrow q}$ $\frac{p \rightarrow q}{\therefore \neg p}$	$[\neg q \wedge (p \rightarrow q)] \rightarrow \neg p$	Modus tollens
$\frac{p \rightarrow q}{q \rightarrow r}$ $\frac{q \rightarrow r}{\therefore p \rightarrow r}$	$[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$	Hypothetical syllogism
$\frac{p \vee q}{\neg p}$ $\frac{\neg p}{\therefore q}$	$[(p \vee q) \wedge \neg p] \rightarrow q$	Disjunctive syllogism

Gereksinim Cümlelerinin Çıkarım Kuralları ile Çözümlemesi

- ◆ Kullanıcı, üyelik oluşturmak istediğinde, kullanıcı adı daha önce alınmamış olmalı ve şifrenin en az 8 karakter uzunluğunda olması gerekmektedir.
 - ◆ Ayrıca, kullanıcı e-posta adresini doğrulamalıdır.
- cümlelerine ait biçimsel (formel) betimlemenin önermeler mantığı ile oluşturulması.

Önermeler;

- ◆ O : Kullanıcı adı orijinal P : Şifre uzunluğu ≥ 8
- ◆ Q : E-posta doğrulandı R : Üyelik oluşturuldu

Çözüm:

$$O \wedge P \wedge Q \Rightarrow R$$

Bu örnekler, daha karmaşık gereksinimlerin mantıksal ifadelerle nasıl ifade edilebileceğini ve çıkarım kurallarının nasıl uygulanabileceğini göstermektedir..

Gereksinim Cümlelerinin Çıkarım Kuralları ile Çözümlemesi

- ◆ Kullanıcı, bir etkinliğe katılmak için kayıt yaptırdığında, etkinliğin tarihinin gelecekte olması ve kontenjanın dolu olmaması gerekmektedir.

cümlesine ait biçimsel (formel) betimlemenin önermeler mantığı ile oluşturulması

Önermeler:

- ◆ F : Etkinlik tarihi gelecekte
- ◆ G : Kontenjan dolu
- ◆ H : Kayıt başarılı

Çözüm:

- ◆ $F \wedge \neg G \Rightarrow H$

Gereksinim Cümlelerinin Çıkarım Kuralları ile Çözümlemesi

- ◆ Bir kullanıcı, bir ürünün iadesini talep ettiğinde, ürünün kullanılmamış olması gerekmektedir.

Cümlesine ait biçimsel (formel) betimlemenin önermeler mantığı ile oluşturulması

Önermeler

- ◆ *D: Ürün kullanılmamış,*
- ◆ *E: İade talebi kabul edildi*

Çözüm:

$$\neg D \Rightarrow \neg E$$

Eğer ürün kullanılmamış değilse, o zaman iade talebi kabul edilmez.

Not: Önerme negatif verildiğinde çıkarım kuralına dikkat.

Gereksinim Cümlesinin Modus Tollens Çıkarım Kuralı ile Çözümlemesi

- ◆ Eğer bir kullanıcı, bir ürünü satın almak istiyorsa, ürünün stokta olması gerekmektedir. Eğer ürün stokta değilse, kullanıcı satın alma işlemini gerçekleştiremeyecektir.

cümlelerine ait biçimsel (formel) betimlemenin önermeler mantığı ile oluşturulması.

Önermeler

- ◆ P : Ürün satın almak istiyor
- ◆ Q : Ürün stokta

Moduls Tollens:

- ◆ $P \Rightarrow Q$
- ◆ $\neg Q$
- ◆ Sonuç: $\neg P$ (Kullanıcı ürünü satın alamaz.)

Modus Tollens Çıkarım Kuralı

Eğer:

◆ $P \Rightarrow Q$ (Eğer P doğruysa, Q doğrudur.)

Ve

◆ $\neg Q$ (Q yanlıştır)

Bu taktirde: (sonuç /conclusion)

◆ $\neg P$ (P yanlıştır)

Gereksinim Cümlesinin Modus Polens Çıkarım Kuralı ile Çözümlemesi

- ◆ Eğer bir kullanıcı, bir oturum açma işlemi gerçekleştirdiyse, kullanıcının erişim izni olmalıdır. Eğer kullanıcı oturum açmışsa, erişim izni verilecektir.

Cümlelerine ait formel çözümlemenin çıkarım kuralları ile yapılması

Önermeler:

- ◆ P :Kullanıcı oturum açtı , Q :Erişim izni var

◆ Mantıksal İfade:

- ◆ $P \Rightarrow Q$ (Eğer kullanıcı oturum açtıysa, erişim izni vardır.) VE
- ◆ P (Kullanıcı oturum açtı.) İSE, BU TAKTİRDE
- ◆ Q (Erişim izni vardır.) (SONUÇ/ CONCLUSION)

gereksinim cümlesi, Modus Ponens kuralıyla çözümlenmiştir..
Kullanıcı oturum açtığında, otomatik olarak erişim izni verilir.

Gereksinim Cümlesinin Disjunctive Syllogism Çıkarım Kuralı ile Çözümlemesi

Disjunctive Syllogism, mantıksal bir çıkarım kuralıdır ve şu şekilde ifade edilir:

Eğer:

- ◆ $P \vee Q$ (P veya Q doğrudur) VE
- ◆ $\neg P$ (P yanlıştır) . BU TAKTİRDE
- ◆ Q (Q doğrudur.) (SONUÇ /CONCLUSION)

Gereksinim Cümlesinin Disjunctive Syllogism Çıkarım Kuralı ile Çözümlemesi

- ◆ Bir kullanıcı, bir etkinliğe katılmak istiyorsa, ya etkinlikte yer kalmalı ya da etkinliğe katılmak için gerekli olan ücret ödenmelidir. Eğer etkinlikte yer kalmadıysa, kullanıcı etkinliğe katılamayacaktır.

Önermeler

- ◆ P : Etkinlikte yer kaldı.
- ◆ Q : Ücret ödendi

Disjunctive Syllogism Çıkarım Kuralı

- ◆ $P \vee Q$ (Etkinlikte yer kalıyor ya da ücret ödendi.)
- ◆ $\neg P$ (Etkinlikte yer kalmadı.)
- ◆ Sonuç:
- ◆ Q (Ücret ödendi.)