

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BASİT TÜRKÇE METİNLERDE OLAY SIRALAMASI

Yüksek Mühendis Şadi Evren ŞEKER

**FBE Bilgisayar Mühendisliği Anabilim Dalında
Hazırlanan**

DOKTORA TEZİ

Tez Savunma Tarihi	: 21 Temmuz 2010
Tez Danışmanı	: Yrd. Doç.Dr. Banu DİRİ(YTÜ)
Jüri Üyeleri	: Prof.Dr. Coşkun SÖNMEZ (YTÜ)
	: Doç.Dr. Tunga GÜNGÖR (BOUN)
	: Yrd. Doç. Dr. Mustafa Cem KASAPBAŞI (İTİCÜ)
	: Yrd. Doç. Dr. M. Fatih AMASYALI (YTÜ)

İSTANBUL, 2010

İÇİNDEKİLER

1.	GİRİŞ.....	1
2.	KLASİK MANTIKTAN ZAMANSAL MANTIĞA GEÇİŞ VE ZAMANSAL MANTIKLAR	3
2.1	Sıfır Seviye Mantığı.....	3
2.2	Birinci Derece Mantık	5
2.2.1	Haber mantığından birinci derece mantığa geçiş.....	6
2.3	İkinci Derece Mantık	7
2.4	Kipler Mantığı	9
2.4.1	Axiological (Axiomatic, Belitsel) Mantık	9
2.4.2	Alethic (Gerekirlik) şekli.....	11
2.4.3	Deontic (Zorundalık) Mantığı	11
2.4.4	Epistemic (Bilgisel) şekli.....	13
2.4.5	Zamansal Mantıklar	14
2.4.6	Kipler mantıkları arasındaki ilişkiler	17
2.5	Zamansal Mantıklarda Etki Alanı Yapısı (Temporal Domain Structure)	20
3.	ZAMANSAL MANTIKLAR.....	25
3.1	LTL (Linear Temporal Logic, Doğrusal Zaman Mantığı).....	26
3.2	Kesit Mantığı (Choppy Logic)	28
3.3	CTL (Computation Tree Logic) Hesaplama Ağacı Mantığı.....	28
3.4	BTTL (Branching Time Temporal Logic) Zaman Çatallanmalı Zamansal Mantık.....	29
3.5	ITL (Interval Temporal Logic) Aralıklı Zaman Mantığı.....	30
3.6	Aralık Mantığı (Interval Logic).....	32
3.7	Genişletilmiş Aralık Mantığı (Extended Interval Logic)	33
3.8	Rtl (Real Time Interval Logic) Gerçek Zamanlı Aralık Mantığı	34
3.9	Allen Zamansal Mantığı (Allen Interval Logic).....	34
3.10	Reichenbach Zamansal Mantığı	37
4.	İŞLENEBİLİR ZAMANSAL DİLLER.....	39
4.1	STTL.....	39
4.2	Html Time.....	40
4.3	Owl Time	42
4.3.1	Sürelerin İfade edilmesi.....	43
4.3.2	Tarih ve Saat kavramı.....	46
4.3.3	OWL Time kullanarak zaman aralığı gösterimi	49
4.4	TTML	52
5.	İŞLENEBİLİR ZAMANSAL MANTIKLARIN KULLANIM ALANLARI.....	53
5.1	Soru Cevaplama.....	53

5.2	Kronoloji Çıkarımı	53
5.3	Metin İşleme Çalışmaları.....	54
6.	TIMEML	56
6.1	Timeml Etiketlerinin Ve Anlamlarının Açıklaması	56
6.1.1	<EVENT> etiketi.....	56
6.2	<TIMEX3> Etiketi	69
6.2.1	TIMEX3'lerin nasıl anlatılacağı.....	69
6.2.2	TIMEX3 etiketi için BNF	75
6.2.3	TIMEX3 için nitelikler	76
6.3	<SIGNAL> Etiketi	89
6.3.1	SIGNAL'lerin nasıl anlatılacağı.....	89
6.3.2	SIGNAL etiketi için BNF	90
6.3.3	SIGNAL için nitelikler	90
6.4	<MAKEINSTANCE> Etiketi.....	90
6.5	MAKEINSTANCE'ların Nasıl Anlatılacağı	91
6.6	MAKEINSTANCE Etiketi İçin BNF	93
6.6.1	MAKEINSTANCE Nitelikleri	94
6.7	Bağlantı Etiketleri: <TLINK>, <SLINK> Ve <ALINK>	104
6.8	<TLINK> Etiketi	104
6.8.1	TLINK'lerin nasıl anlatılacağı.....	106
6.8.2	TLINK etiketi için BNF	110
6.8.3	TLINK'ler için Nitelikler	110
6.9	<SLINK> Etiketi	111
6.9.1	SLINK'lerin nasıl anlatılacağı.....	114
6.9.2	SLINK etiketi için BNF	116
6.9.3	SLINK'ler için Nitelikler.....	117
6.10	<ALINK> Etiketi.....	118
6.10.1	ALINK'lerin nasıl anlatılacağı	118
6.10.2	ALINK etiketi için BNF	119
6.10.3	ALINK'ler için Nitelikler	120
6.10.4	Tüm LINK etiketleri için ek nitelikler.....	120
6.11	Karmaşık Timex Örnekleri	120
6.12	Karmaşık TLINK Ve SLINK Örnekleri	127
6.13	Nedensel Örnekler	134
7.	TÜRKÇE İÇİN ZAMANSAL MANTIK VE TIMEML.....	137
7.1	Timeml Neleri Yapamaz	137
7.1.1	Terslik ve Tekrarlılık ilişkisi	138
7.1.2	Kendine atıf durumu	139
7.1.3	Zamanda tekrar durumu.....	140
7.1.4	Türkçedeki zamanlar ve fiil yapılar	141
7.2	Yaprak Seviyesi İyileştirmeler	146
7.3	İlişki Seviyesi İyileştirmeler	149
8.	DERLEM ÇALIŞMASI	152

9.	TIMEML ÜZERİNDE UYGULAMA GELİŞTİRİLMESİ	159
9.1	Kronoloji Çıkarımı	159
10.	SONUÇLAR VE GELECEK ÇALIŞMALAR	168

SİMGE LİSTESİ

$\wedge$	ve
$\vee$	veya
$\Rightarrow$	ise
$\Leftrightarrow$	ancak ve ancak
$\neg$	değil (ters)
D veya T	doğru.
Y veya $\perp$	yanlış
$\exists$	bazı
$\forall$	her
$\diamond$	muhtemel(possibility)
$\square$	zorunda(necessity)
Gp	Globally, p önermesinin her zaman için doğru olduğu anlamındadır
Fp	Future p önermesinin gelecekte herhangi bir anda doğru olduğu anlamındadır
pUq	Until, p önermesinin q önermesinin doğruluğuna kadar doğru olduğu anlamındadır. Diğer bir deyişle, p olayı q olayından öncedir
Xp	Next p önermesinin bir sonraki önerme olduğunu belirtir
O	Sonraki (Next)
ξ	Kesit operatörü (Chop)
$\odot$	RTIL için başlangıç ve bitiş zamanı gösterir

KISALTMALAR LİSTESİ

AZM	Allen Temporal Logic	Allen Zamansal Mantığı
BNF	Backus-Naur Form	Backus-Naur Şekli
BTTL	Branching time temporal logic	Zaman çatallanmalı zamansal mantık
CTL	Computation Tree Logic	Hesaplama Ağacı Mantığı
EST	Eastern Standard Time	Doğu Standart Zamanı
FOL	First Order Logic	Birinci Seviye Mantığı
GMT	Greenwich Mean Time	Greenwich Yerel Saati
HTML	Hyper Text Markup Language	Hiper Metin İşaretleme Dili
IL	Interval Logic	Aralık Mantığı
ITL	Interval Temporal Logic	Aralık Zamansal Mantığı
LTL	Linear Temporal Logic	Doğrusal Zaman Mantığı
MathML	Math Markup Language	Matematik İşaretleme Dili
NP	Nondeterministic Polynomial	Belirsiz Çokterimli
OWL	Web Ontology Language	Web Ontoloji Dili
PTL	Propositional Temporal Logic	Önerme Zamansal Mantığı
PST	Pacific Standard Time	Pasifik Standart Zamanı
RTIL	Real Time Interval Logic	Gerçek Zamanlı Aralık Mantığı
SOL	Second Order Logic	İkinci Derece Mantık
STTL	Standard Time Tabling Language	Standart Zaman Tablolama Dili
TimeML	Time Markup Language	Zaman İşaretleme Dili
TTML	Time Tabling Markup Language	Zaman Çizelgeleme İşaretleme Dili
VP	Verb Phrase	Fiil Kelime Grubu
WFF	Well Formed Formula	İyi Şekillendirilmiş Formül
XML	Extendible Markup Language	Genişletilebilir İşaretleme Dili

ŞEKİL LİSTESİ

Şekil 2.1 Mantık sisteminde kullanılan temsili çıkarım	4
Şekil 2.2 Zamansal mantık operatörlerinin zaman çizgisi üzerinde gösterimi.....	15
Şekil 2.3 N operatörünün p önermesi üzerindeki etkisi	16
Şekil 2.4 F operatörünün p önermesi üzerindeki etkisi	16
Şekil 2.5 G operatörünün p önermesi üzerindeki etkisi	16
Şekil 2.6 –e kadar (until) operatörünün p ve q önermeleri üzerindeki etkisi	17
Şekil 2.7 Sonra operatörünün p ve q önermelerine etkisi.....	17
Şekil 2.8 Zamansal mantık özelliklerinin ilişki görüntülemesi (Bellini, 2000).....	19
Şekil 2.9 Klasik mantık ve zamansal mantıklar arasındaki seviye ilişkisi	20
Şekil 2.10 Mantık seviyeleri.....	20
Şekil 2.11 Zamansal mantıklarda doğrusallık ve kesinlik ilişkiler,.....	22
Şekil 2.12 Zamansal mantıklardaki olay ilişkileri ve AZM ilişki şekilleri	23
Şekil 2.1 Zamansal mantıklardaki olay ilişkileri ve AZM ilişki şekilleri	24
Şekil 3.1 Zamansal mantık ve hesaplanabilir mantık ilişkisi	25
Şekil 3.2 Örnek olay ilişki şeması	27
Şekil 3.3 Örnek olay ilişki şemasına ilave bir olay eklenmesi	27
Şekil 3.4 Örnek olay durum ilişkisi.....	35
Şekil 3.5 Örnek olay durum ilişkisi.....	36
Şekil 3.6 Örnek olay durum ilişkisi.....	37
Şekil 7.1 Örnek tekrarlı cümle modellemesi	139
Şekil 7.2 Örnek emir sıralaması	140
Şekil 7.3 Kendine atıfta bulunan örnek olay sıralaması	140
Şekil 7.4 Zamanda geri dönülmesi durumu.....	141
Şekil 7.5 Örnek zaman olay ilişkisi.....	144
Şekil 7.6 Örnek zaman olay ve yazar rivayet ilişkisi	144
Şekil 7.7 Örnek zaman olay ve yazar rivayet ilişkisi	145
Şekil 7.8 Örnek zaman olay ve yazar rivayet ilişkisi	145
Şekil 7.10 Türkçe için iyileştirilmiş AZM	150
Şekil 9.1 Tarsqi Kontrol Paneli Görüntüsü	160
Şekil 9.2 TimeML dönüşümü.....	161

Şekil 9.3 Tango ekran görüntüsü.....	165
Şekil 9.4 T-Box Ekran Görüntüsü	165
Şekil 9.5 TANGO örnek çalışma.....	166
Şekil 9.6 Örnek çalışmanın TBox Çıktısı.....	167

ÇİZELGE LİSTESİ

Çizelge 2.1 Kipler mantığındaki sembollerin karşılaştırılması	18
Çizelge 2.2 Zamansal mantık özelliklerin	19
Çizelge 3.1 Reichenbach zaman ilişkileri.....	38
Çizelge 7.1 Reichenbach zamansal mantığı ile Türkçe karşılaştırılması	142
Çizelge 7.2 Reichenbach Zamansal Mantığının Türkçeye Uygulanması.....	147
Çizelge 7.3 Reichenbach Zamansal Mantığının Türkçeye Uygun Hali	148
Çizelge 8.1 Derlem, Cümle, Kelime ve Fiil Sayıları.....	153
Çizelge 8.2 Derlem, Türkçe Zamanlara göre fiil sayıları	154
Çizelge 8.3 Derlem, Türkçe Zamanlara göre fiil oranları	155
Çizelge 8.4 Derlem, Birleşik Cümle Sayıları	155
Çizelge 8.5 Derlem Reichenbach Seviyesi iyileştirme.....	156
Çizelge 8.6 İyileştirme Sonrası Reichenbach Seviyesi Başarılar	157

ÖNSÖZ

Bu tez dokümanı, doktora çalışmasının sonuçlarını içermektedir. Bu sonuçlar, konu hakkında bilgisi olmayan bir okuyucuya yönelik olarak aşağıdaki sıra ile tasnif edilmiştir:

- Zamansal mantığın (temporal logic) tanımı ve kullanım alanları
- Zamansal mantık şekilleri
- İşlenebilir zamansal mantıkları (computable temporal logics)
- TimeML
- Türkçe için TimeML iyileştirmesi
- Derlem çalışmaları ve iyileştirmenin uygulanması

Yukarıdaki bu listeden de anlaşılacağı üzere bu tezin getirmiş olduğu bilimsel yenilik, Türkçe için işlenebilir bir zamansal mantık oluşturulması ve bu zamansal mantığın bir bilgisayar yazılımı tarafından kullanılmasıdır.

Bu konunun açıklanması için öncelikle zamansal mantık konusu anlatılıp güncel örnekleri gösterilecek ardından işlenebilir zamansal mantıklardan bahsedilerek bu konuda Türkçe doğal dil işleme çalışmalarına en uygun olarak seçtiğimiz TimeML detaylıca anlatılacaktır.

Ardından bu çalışmanın kapsamına giren zamansal mantıklara Türkçedeki zamansal kavram ve mantıkların eklenmesi ve dolayısıyla TimeML üzerinde yapılan değişiklikler anlatılıp son olarak bu çalışmanın üzerinde test edildiği derlem ve bu çalışmadan üretilen bilgisayar yazılımı açıklanacaktır.

Bu tezin hazırlanmasının her aşamasında sınırsız desteği ile gerek çalışmalarımda gerekse motivasyonumda önemli rol oynayan değerli hocam Yrd. Doç. Dr. Banu DİRİ'ye teşekkürü bir borç bilirim.

Bilgisayar mühendisi olmam ve akademik çalışmalara yönelmem konusundaki emeklerinden dolayı Prof. Dr. Şebnem BAYDERE'ye, seçmiş olduğum konuda tecrübeleri ile beni yönlendiren Prof. Dr. Kemal OFLAZER hocama ve doğal dil işleme çalışmalarında ulaştığım noktada emekleri ile katkısı çok büyük olan Prof. Dr. A. C. Cem SAY hocama teşekkür ederim.

ÖZET

Bu doktora çalışmasında, basit Türkçe Metinlerde olay sıralaması hedeflenmiştir. Literatürde olay sıralaması (event ordering) veya kronoloji çıkarımı (chronology extraction) olarak geçen bu işlem için metinler üzerinden deneysel bir yolla karşılaşılan problemleri çözmek yerine, olay sıralama işleminin bilimsel olarak ele alınıp, Türkçedeki bütün dilbilgisi kurallarını içeren bir şekilde çözülmesi hedeflenmiştir.

Bu anlamda, daha önceden var olan zamansal mantıklar incelenmiş, bu mantıkların çalışma şekilleri ve alanları ortaya konulmuştur. Ayrıca bu mantıkların bilgisayarlar tarafından işlenebilir olan uygulamaları da incelenerek, amaca yönelik literatür taraması yapılmıştır. Bu tarama sonucunda en uygun dil olarak TimeML isimli dil seçilmiş ve bu dil üzerinde Türkçe için iyileştirme yapılmasına karar verilmiştir.

TimeML dilinin üzerine kurulu olduğu iki zamansal mantık (Reichenbach ve Allen Zamansal Mantık) için Türkçedeki dilbilgisel zaman ifadeleri ile karşılaştırma yapılmıştır. Çıkan eksik noktalar için gerek zamansal mantıklara, gerek ise TimeML üzerine, ekleme yapılarak Türkçede bulunan, ancak henüz literatüre girmemiş olan, bu farklar ortaya konulmuş ve mantıksal olarak modellenerek, bir işlenebilir dil üzerinde ifade edilmiştir.

Yapılan bu ekleme çalışmalarının başarısını bulabilmek için, ayrıca bir derlem oluşturulmuş ve bu derlem üzerinde TimeML dilinin eski ve yeni halleri karşılaştırılarak, başarı oranı ölçülmüştür.

Yeni tasarlanan TimeML dilinde, Türkçedeki bütün zamansal ifadeler kapsanmış ve modellenmiştir. Ayrıca dilbilgisel çalışmanın getirdiği bir özellik olarak bu derlemde karşılaşılmayan bazı durumlar da bu doktora çalışması kapsamında ele alınmıştır.

Son olarak TimeML üzerinde yapılan bu değişiklikleri kapsayan bir yazılım hazırlanmış, hazırlanan bu yazılımda yapılan zamansal çıkarımlar görsel olarak işlenmiş ve bir çizelge olarak gösterilmiştir.

Anahtar Sözcükler: Olay Sıralama (Event Ordering), Kronoloji Çıkarımı (Chronology Extraction), Doğal Dil İşleme (Natural Language Processing), İşlenebilir Diller (Computable Languages), Zamansal Mantıklar (Temporal Logics).

ABSTRACT

Event Ordering for Simple Turkish Texts

This PhD. study aims to order events in simple Turkish texts. In the literature this study is classified as event ordering or chronology extraction and there are some studies approaching the problem in empirical ways and some using methodological linguistic approach. In this study a linguistic approach has been applied to cover all possible cases in Turkish.

This thesis study can be divided into two sub parts. In the first part, the current literature about Temporal logics has been researched and also the computable languages developed over these logics has been reported. This research has showed that the most suitable computable language is TimeML.

TimeML is built over two Temporal logics which are Reichenbach and Allen's Temporal logics. These Temporal logics has been compared by Turkish linguistic and the missing parts are added into both logical and computable language layers. At the best of author knowledge, in the study, carried out within thesis, the Turkish Temporal logic is first time formally modelled by a Temporal logic and this logic is implemented on a computable language.

Also a corpus is created to test the success of these additions on TimeML and both the classical TimeML and enhanced TimeML versions are tested on the corpus and the success of tagging is measured.

In the enhanced version of TimeML, all the Temporal models are covered and a full success is achieved. Also, some of the enhancements suggested on TimeML, can not be noticed on corpus, because this research is built on a formal linguistic methodology and some cases are rarely seen in Turkish texts.

Finally a software working on the computable language, enhanced TimeML and the outputs are visualized by a chart on this software.

1. GİRİŞ

Bu doktora tezinin amacı, zamansal mantıkların araştırılması ve Türkçe doğal diline uyarlanmasıdır. Basit bir Türkçe metinde geçen olayların ve zaman kavramlarının sıralanması ve bu sıralamanın modellenmesi hedeflenmiştir. Bu hedef için basit çocuk hikayeleri hedef seçilmiş ancak zaman içerisinde çalışma, genişletilerek bütün Türkçe dilbilgisini kapsayacak hale gelmiştir.

Bu çalışma bilgisayar bilimlerinin bir alt kolu olan yapay zeka konusundaki doğal dil işleme çalışmaları kapsamında düşünülebilir. Hedeflenen işlem literatürde olay sıralaması veya kronoloji çıkarımı olarak geçmekte olup, bu çalışmaya kadar henüz Türkçe için yapılmış farklı bir çalışma bulunmamaktadır.

Günümüzde bilgisayar bilimlerinde yapılan çalışmalar insanlara özgü pek çok konuda ilerlemeler sağlamaktadır. Örneğin doğal dillerin matematiksel olarak modellenmesi ne yazık ki mümkün değildir. Ayrıca dilbilgisi çalışmaları altında dillere özgü kuralların çok sayıda istisnaları bulunmaktadır. Hatta doğal dildeki bazı cümleler insanlar için bile karmaşık olabilmekte ve anlama güçlükleri barındırabilmektedir.

Bir insan başarısı ile, doğal dilde cümleleri anlayabilen ve anlamlı sonuçlar üretebilen makinelerin üretilmesi henüz mümkün olmasa bile çok yakın bir gelecekte bu konuda yeterli ilerlemenin sağlanacağı tahmin edilebilir. Örneğin bu çalışma sayesinde doğal dil çalışmalarında önemli yer tutan zaman kavramı ve zamansal bilgilerin analiz edilmesi ve modellenmesi daha kolay bir hale getirilmiştir.

Doğal dil çalışmalarının kullanıldığı önemli alanlar olarak verilen bir metinden test üreten otomatik soru yazılımları, bir metinde belirli bir sorunun cevabını arayıp otomatik olarak bulan soru cevap çalışmaları (Hirschman, L., 2001), arama motorlarının iyileştirilmesinde kullanılan dil çalışmaları, metin özetleme, şekillerin doğal dilde anlatılmaları ve şekil okuma gibi pek çok güncel konu sayılabilir.

Bu tezde, yukarıda bahsedilen doğal dil çalışmalarına, zamansal anlamda, yardımcı olacak bir temel oluşturulmuştur. Örneğin soru cevap çalışmalarında zamansal soruların üretilmesi veya cevaplanması, arama motorlarında zamansal soruların cevabının aranması veya zamansal ontolojilerin kurulması gibi çalışmaların hepsinde bir modelleme ihtiyacı bulunur.

Bu döküman kapsamında öncelikle literatür araştırmasına ve zamansal mantıkların

literatürdeki sınıflandırılmasına yer verilecektir. Doktora çalışmasının önemli bir kısmı mantık biliminin altında sayılabilecek bu zamansal mantıkların anlaşılması ve üzerlerine Türkçedeki bazı ifadeleri eklemek için iyileştirilme yapılmasını gerektirmektedir.

Ardından bu zamansal mantıkların bilgisayar bilimlerinde kullanılması ve dolayısıyla bilgisayarlar tarafından işlenebilir zamansal mantıkların açıklanmasına yer verilecektir.

Bu tez çalışmasının, üzerinde yoğunlaştığı ve yeni kısımlar eklenerek üzerinde kod geliştirilen TimeML dili ise özel bir bölümde detaylıca açıklanacaktır.

Son olarak doktora çalışması kapsamında üzerinde işlem yapılan derlem ve bu derlem üzerinde elde edilen başarılarından bahsedilecektir.

2. KLASİK MANTIKTAN ZAMANSAL MANTIĞA GEÇİŞ VE ZAMANSAL MANTIKLAR

Bu bölümün amacı zamansal mantık yaklaşımını açıklamak, bu alanda literatürde bulunan çeşitli zamansal mantıkları anlatmak ve klasik mantık yaklaşımıyla zamansal mantık yaklaşımını karşılaştırmalı olarak incelemektir. Buradaki amaç, tez çalışmasının da konusu olan zamansal mantık kavramını açıklamak ve çözümlemeli bir model ortaya koymaktır.

Klasik mantıkta üç seviyeden bahsetmek mümkündür bu üç seviye, literatürdeki farklı isimleri ile aşağıda listelenmiştir:

1. Sıfır seviye mantığı (zeroth order logic), Edat Mantığı, Haber mantığı veya önermeler mantığı (Propositional Logic, Propositional Calculus) (Boolos, 1975)
2. Birinci seviye mantığı (first order predicate calculus) (Boolos, 1975)
3. İkinci seviye mantık (second order calculus) (Väänänen, 2001)

Yukarıda sıralanan bu üç seviyeyi kısaca karşılaştırmamız gerekirse, sıfır seviye mantığında klasik mantık operatörlerine ilave olarak herhangi bir etki alanı (domain) tanımı söz konusu değilken, birinci seviye mantıkta, mantığın niceleyicilerinde (quantifiers) bir etki alanı tanımı yapılabilmektedir. İkinci derece mantıkta ise hem niceleyiciler (quantifier) hem de haberler (predicates) üzerinde etki alanı tanımlanabilmektedir.

Bu üç seviye aşağıda daha detaylı olarak anlatılacaktır:

2.1 Sıfır Seviye Mantığı

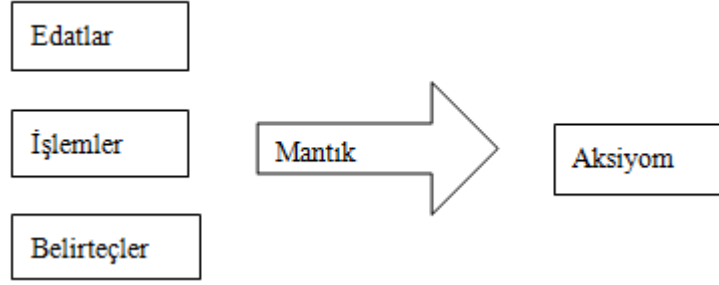
Literatürde, sıfır seviye mantığı, aynı zamanda, haber mantığı (edat, önerme veya önerme kelimeleri de haber yerine kullanılmaktadır) önerme mantığı veya haber matematiği (propositional logic, propositional calculus) olarak geçmektedir (Fischer, 1979).

Bu mantıkta amaç, “doğru” veya “yanlış” sonuçları ile biten bir mantıksal düzenek hazırlamak ve bu düzeneğe uygun bir şekilde problemleri modelleyerek çözmektir. Bu anlamda sonuçların doğru veya yanlış ile bitmesinden dolayı doğruluk fonksiyonu mantığı (truth functional propositional logic) ismi de verilmektedir.

Sıfır seviye mantığının da içinde bulunduğu mantık sistemleri birer matematiksel model olarak görülebilir ve bu matematiksel modeller için gerekli olan dört önemli unsur aşağıdaki şekilde sayılabilir:

1. Mantığın üzerine kurulu olduğu edatlar (önergeler, haberler, yüklem, predicates)
2. Mantıkta tanımlı olan işlemler (operators)
3. Mantıkta kullanılan belirteçler (quantifiers)
4. Yukarıdaki 3 özelliğin kullanılması ile çıkarılan aksiyomlar (axioms)

Bu dört özellik arasındaki ilişki Şekil 2.1 ile ifade edebilir.



Şekil 2.1 Mantık sisteminde kullanılan temsili çıkarım

Yukarıdaki durumu bir örnek üzerinden açıklamak gerekirse.

“Benim elmam var, benim elmam kırmızı, Ayşenin de elması var, o halde Ayşe’nin elması da kırmızı”

$\text{sahip}(\text{elma}, \text{ben}) \wedge \text{kırmızı}(\text{elma}, \text{ben}) \wedge \text{sahip}(\text{elma}, \text{Ayşe}) \Rightarrow \text{kırmızı}(\text{elma}, \text{Ayşe})$

Yukarıdaki gösterimde belirteçler kümesi {elma, ben, Ayşe}, işlemler kümesi { $\wedge$, $\Rightarrow$ } ve önermeler kümesi { sahip, kırmızı } olarak sıralanabilir. Yukarıdaki bu dizilimin tamamına ise bir aksiyom ismi verebiliriz. Aksiyom kelimesi yerine çeşitli kaynaklarda düzgün şekillendirilmiş formül (well formed formula, wff) tabiri de kullanılmaktadır (Andrews., 2002).

Sıfır seviye mantığında, doğru veya yanlış sonucuna ulaşmayı etkileyen işlemlerin kendileridir. Bu işlemler, üzerine kurulu oldukları önermelerin doğru veya yanlış olmasına göre sonuç üretir.

Sıfır seviye mantığında bulunan ikili işlemler (binary operators) aşağıdaki şekilde sıralanabilir:

$\wedge$ ve

$\vee$ veya

$\Rightarrow$ ise

$\Leftrightarrow$ ancak ve ancak

Ayrıca mantıkta bulunan tekli işlem (unary operator) ise bir tanedir

$\neg$ değil (ters)

Kullanılan önermelerin aldığı değerler, doğru veya yanlış olabilmektedir. Bu durum iki farklı şekilde $\{ D, Y \}$ veya $\{ T, \perp \}$ sembolleri ile gösterilir (Copi, 1971).

2.2 Birinci Derece Mantık

Bu modelin özelliği kıyas ve tümden gelim yaklaşımına göre ispatlanabilir özellikte olmasıdır (deductive). Yani bir makine, veya matematiksel model tarafından bu mantık modelindeki gösterimlerin ispatlanması (verification) veya reddedilmesi (falsification) mümkündür.

Birinci derece mantık, literatürde çeşitli isimlerle adlandırılmaktadır. Örneğin, mantığın üzerine inşa edildiği yüklem ve haberlerden (predicate) ismini alan “birinci derece haber mantığı” (first order predicate logic) veya bu mantık üzerinde matematiksel işlemlere izin vermesinden dolayı birinci derece yüklem aritmetiği (first order predicate calculus) veya yine birinci derece kelimesinin sebebi olan ve mantıkta önermeler mantığına (önermeler, propositional logic) yakın olması dolayısıyla alçak haber mantığı (lower predicate calculus) isimleri gibi farklı isimler verilmektedir.

Birinci derece mantığın, klasik önerme mantığından (önermeler mantığı, propositional logic) en önemli farkı, bu mantıkta kullanılan niceleyicilerin (quantifiers) veya bilgisayar bilimlari gözüyle değişkenlerin (variable) belirli bir tanım alanlarının olmasıdır (domain). Dolayısıyla birinci derece mantıkta yapılan modellemelerde kullanılan bütün niceleyiciler, tanımlı bir alanda geçerlidir ve mantık modellerimiz bu alan için doğru veya yanlış olarak işlenir.

Bu tanımlı olma özelliğinin getirdiği bir zafiyet olarak, birinci derece mantıkta sonuz kümeler olan, doğal sayılar veya Kartezyen uzay gibi varlıkların modellenmesi ve çözülmesi mümkün olmamaktadır. Bu tür sonsuzluk içeren modellemeler için ikinci derece mantık benzeri daha kuvvetli mantık modellemelerinden yararlanmak mümkündür.

2.2.1 Haber mantığından birinci derece mantığa geçiş

Birinci derece mantık konusunu anlamak için öncelikle klasik, önermeler mantığının (propositional logic) bazı özelliklerinin bilinmesi gerekir.

Örneğin “Şadi mühendistir.” cümlesi, klasik mantık açısından bir önermedir ve doğru veya yanlıştır.

Benzer şekilde “Ali mühendistir.” Cümlesi de klasik mantık açısından bir önermedir ve tek başına ele alınarak doğruluğu veya yanlışlığı ortaya konulabilir.

Ancak klasik önerme mantığında bu iki ayrı cümle yine bu mantıkta tanımlı olan klasik işlemler (operators) dışında birleştirme ve modelleme yapılamaz.

Örneğin bu önermelerden birisine p ve diğerine q sembolü verilip mantıkta bulunan aşağıdaki durumlar modellenenabilir:

$p \rightarrow q$ (Şadi mühendis ise Ali de mühendistir)

$p \wedge q$ (Şadi de Ali de mühendistir)

$p \vee q$ (Şadi veya Ali mühendistir)

$\neg p$ (Şadi mühendis değildir)

Yukarıdaki bu modeller sonuçta doğru (True (T)) veya yanlış (False (F, $\perp$)) sonuçlarından birisini üretir.

Birinci derece mantık, yukarıdaki bu modellemeyi öncelikle niceleyiciler (quantifier) üzerine taşır. Bu niceleyiciler, bilgisayar bilimlerindeki değişkenler (variables) olarak düşünülebilirler ve kabaca bir varlığa verilen ve mantık modelinde kullanılan isim olarak tanımlanabilir.

Yukarıdaki “Şadi mühendistir” cümlesini ele alacak olursak birinci derece mantıkta aşağıdaki şekilde göstermek mümkündür:

$\text{Şadi}(a) \rightarrow \text{Mühendis}(a)$

Modeli okurken “öyle bir “a“ vardır ki bu “a”nın ismi Şadi’dır ve bu “a” mühendistir” şeklinde okuyabiliriz. Burada kullanılan a bir niceleyicidir (quantifier) ve bu niceleyici üzerinde bir haber (predicate) bildirilmiştir. Bu modelde, klasik modele göre fark net bir şekilde görülmektedir. Tam bu noktada klasik mantıkta soramayacağımız ama birinci derece

mantıkta sorulması gereken “bu a nasıl bir a’dır?” sorusu farkı ortaya koyar. Birinci derece mantıkta bu sorunun anlamı, bu bütün a’lar için geçerli midir, anlamındadır. Yukarıdaki tanımda da anlatıldığı üzere birinci derece mantıkta niceleyicilerin (quantifiers) bir etki alanının olması söz konusudur. Yani a ile sembolize edilen bu gösterici acaba her durum için geçerli bir sembol müdür?

Bu sorunun cevabını farkı anlayabilmek için açık bir şekilde hayır diye verebiliriz. Yukarıdaki a niceleyicisi her durumda doğru olsaydı, bütün ismi Şadi olanlar mühendistir sonucuna varılması gerekirdi ki bu doğru değildir.

İşte bu ayrım bize birinci derece mantıkta kullanılan iki önemli sembolü gösterir. Bunlardan birincisi “her” anlamına gelen ($\forall$) sembol iken diğeri “öyle bazı” şeklinde okunabilecek ($\exists$) semboldür.

Yukarıdaki örneğimizi düzenleyecek olursak aşağıdaki gösterim doğrudur ve sonucu da doğrudur:

$\exists a (\text{Şadi}(a) \wedge \text{Mühendis}(a)) \rightarrow \text{Öyle bazı } a\text{'lar vardır ki, bu } a\text{'nın ismi Şadi'dir ve bu } a \text{ mühendistir.}$

$\forall a (\text{Bilgisayar Mühendisi}(a) \rightarrow \text{Mühendis}(a)) \rightarrow \text{Bütün } a\text{'lar için, } a \text{ şayet bilgisayar mühendisiyse; } a, \text{ aynı zamanda mühendistir.}$

Yine, yukarıdaki gösterim kullanılarak söylenebilecek doğru bir model de aşağıdaki gibi olabilir:

$\exists a (\text{Şadi}(a) \wedge \neg \text{Mühendis}(a)) \rightarrow \text{Öyle bazı } a\text{'lar vardır ki, bu } a\text{'nın ismi Şadi'dir ve bu } a \text{ mühendis değildir.}$

2.3 İkinci Derece Mantık

Bu mantığa göre modellenecek sistemdeki varlıklar ve bu varlıklar üzerine uygulanacak olan işlemler ayrı ayrı ele alınır ve sistemde modellenen hem varlıkların hem de işlemlerin geçerli olduğu bir etki alanından söz edilebilir.

İkinci derece mantığı anlamak için basitçe birinci derece mantık (first order predicate calculus, first order logic) ile farkı anlaşılmalıdır.

Birinci derece mantıkta, klasik mantıktan farklı olarak niceleyiciler (quantifiers)

kullanılmakta ve bu niceleyicilerin geçerli olduğu bir etki alanından (domain) bahsetmek mümkündür. Ancak bu niceleyiciler üzerindeki tanımları ve işlemleri ifade eden haberler (predicates) üzerinde herhangi belirleyici bir işlem yapılamıyordu.

Örneğin aşağıdaki birinci derece mantıkta modellenmiş iki gösterimi ele alalım:

$\exists a (\text{\textit{\$adi}}(a) \wedge \text{\textit{Mühendis}}(a)) \rightarrow \text{\textit{Öyle bazı a'lar vardır ki, bu a'nın ismi \$adi'dir ve bu a mühendistir.}}$

$\forall a (\text{\textit{Bilgisayar Mühendisi}}(a) \Rightarrow \text{\textit{Mühendis}}(a)) \rightarrow \text{\textit{Bütün a'lar için, a şayet bilgisayar mühendisiyse; a, aynı zamanda mühendistir.}}$

Yukarıdaki bu gösterimlerde dikkat edileceği üzere niceleyici (quantifier) olan a üzerinde bir etki alanı tanımlanmıştır buna mukabil haberler (predicates) üzerinde bir etki alanı tanımlanmamış ve bilgisayar mühendisi tanımı olduğu gibi bırakılmıştır. İşte ikinci derece mantık tam burada devreye girer ve bu haberlerin (predicates) üzerinde de etki alanı tanımını mümkün kılar.

Bu durumu aşağıdaki örnek üzerinden anlaşılabilir:

$\exists \text{\textit{\$ekil}} (\text{\textit{\$ekil}}(a) \wedge \text{\textit{\$ekil}}(b)) \rightarrow \text{\textit{Öyle bir şekilden bahsedilebilir ki, hem a hem de b aynı şekildedir.}}$

Yukarıdaki bu durum birinci seviyede yazılamaz. Bunun sebebi şekil belirten haberin (predicate) her durumda aynı olması gerekliliğidir. Diğer bir deyişle birinci seviye mantıkta tek bir şekilden bahsedilebilir, özel bir şekil olamaz.

Bu durum biraz daha ileri götürülürse, haberler (predicates) üzerinde de haberler tanımlamak mümkündür. Yani haberin haberi şeklinde bu durumu modelleyebiliriz.

$\exists \text{\textit{\$ekil}} (\text{\textit{Köşeli}}(\text{\textit{\$ekil}}) \wedge \text{\textit{\$ekil}}(a) \wedge \text{\textit{\$ekil}}(b))$

Yukarıdaki bu tanımda, bir şeklin köşeli olacağı, ve bu durumda a ve b niceleyicilerinin bu şekilden olabileceği tanımı yapılmıştır.

Bu anlamda birinci seviye mantıktan daha kapsamlı modellemelerin yapılabildiği ikinci seviye mantık, bazı durumlarda da sorun arz eder.

2.4 Kipler Mantığı

Mantığın bir türü olan şekli mantığında şekiller (modal) bir önermenin doğruluğunu göstermek için kullanılır. Genel olarak şekil mantığında gösterilen 3 tip şekil bulunur:

- olabilirlik (possibility)
- ihtimal (probability)
- gereklilik (necessity)

Doğal dil açısından ve dilbilim gözüyle şekli mantığa bakarsak, aslında yukarıdaki bu liste ve şekil mantığının dayanağı İngilizcede açıkça kullanılan ve kelime zamana ve bakışlarını etkileyen modallara bağlıdır. Yani İngilizcede “*eventually, formerly, can, could, might, may, must, ought, need*” gibi modalların her birisi bu yukarıda sayılan üç tipten birisine girmektedir.

Türkçe için tam bir karşılığı olmayan bu kavramlar, Türkçede kipler ve fiil zamanları kavramları ile karşılanmaktadır.

Mantık bilimi açısından konu incelendiğinde aslında dört tip işlemin (operator) üzerinde tanımlı olduğu bir mantıktan bahsedilmektedir. Bu işlem tipleri aşağıdaki şekilde sıralanabilir:

- Alethic (gerekirlik)
- Deontic (Ahlaksal)
- Axiological (Belitsel)
- Epistemic (Bilgisel)

Yukarıda sıralanan bu şekli mantık tipleri aşağıda açıklanmıştır.

2.4.1 Axiological (Axiomatic, Belitsel) Mantık

Önergeleri, gerekirlik (necessary) ve olasılık (possibility) olarak değerlendirildiği mantıktır (Cantini, 1996). Bu yaklaşıma göre bir önerme nin gerekir veya olası olması aşağıdaki operatörler ile ifade edilir:

$\Box p$ (gerekir (necessarily) p)

$\Diamond p$ (olası (possibly) p)

Yukarıdaki operatörlerin kullanımını aşağıdaki örnek üzerinden anlayabiliriz.

P: İnsanlar yemek yer

Q: İnsanlar halı dokur.

Bu önermeler için aşağıdaki gösterimler doğrudur. Öncelikle ilk önermeyi sıfırıncı düzey mantıkla modelleyelim.

P: İnsan (x) $\wedge$ yemekyer(x)

Ardından bu önermeyi birinci seviye mantığa çıkaralım

P: $\forall x (insan(x) \wedge yemekyer(x))$

Bu mantığa axiomatic gösterim ekleyelim:

P: $\forall x (insan(x) \wedge \Box yemekyer(x))$

İkinci önermeyi de aynı şekilde modellemeye çalışalım:

Q: İnsan (x) $\wedge$ halıdokur(x)

Ardından bu önermeyi birinci seviye mantığa çıkaralım

Q: $\exists x (insan(x) \wedge halıdokur(x))$

Bu mantığa axiomatic gösterim ekleyelim:

Q: $\exists x (insan(x) \wedge \Diamond halıdokur(x))$

Bu iki birinci derece mantık normalde birleştirilseydi aşağıdaki problem oluşurdu

?x (insan(x) $\wedge$ halıdokur(x) $\wedge$ yemekyer(x))

Yukarıdaki ? işareti yerine ne yazılacağı problemidir. Oysaki bu problem kipler mantığında bulunan axiomatic şekil ile çözülebilir ve aşağıdaki sonuca ulaşılır:

$\forall x (insan(x) \wedge \Diamond halıdokur(x) \wedge \Box yemekyer(x))$

Yukarıdaki bu gösterimi şu şekilde okuyabiliriz: Bütün insanların halı dokuma ihtimali bulunur (possibly) ve bütün insanlar yemek yemek zorundadır (necessity).

2.4.2 Alethic (Gerekirlik) şekli

Bu şekli mantık türünde bir önermenin yargılanması gerekirlik açısından yapılır. (Bailhache, 1998)

Örneğin “Çember kare olamaz” cümlesindeki olamamazlık ile “Ali mühendis olamaz” cümlesindeki olamamazlık arasında fark vardır. Birincisi doğası itibariyle imkansız iken (olmaması gerekir iken) ikincisinin olmamasının gerekliliğinden sadece bir yorum veya kanaat olarak bahsedilebilir. Dolayısıyla ilk önerme için alethic (gerekirlik şeklinden doğru) denilirken ikinci önerme için non-alethic (gerekirlik şeklinden yanlış) denilebilir.

Klasik olarak gerekirlik şeklini üç başlıkta toplayabiliriz:

- olabilirlik (possible) : Önermenin yanlış olaması gerekmiyorsa
- gerekirlik (necessary) : Önermenin yanlış olma ihtimali yoksa
- şartlı (contingent) : Önermenin yanlış olması gerekmiyorsa ve aynı anda önermenin doğru olması da gerekmiyorsa

Yukarıdaki bu üç grup için önermelerde gerekirlik aranabilir. Buradaki üçüncü madde biraz karışık gelebilir. Bu maddeye göre bir önermenin aynı anda hem doğru hem de yanlış olmaması durumu basitçe “doğruluğu mümkün ama gerekli değil” şeklinde açıklanabilir.

Gerekirlik şeklini ayrıca gerekirlik dünyası açısından da incelemek mümkündür. Yani bir önerme şayet fiziksel kurallar dahilinde gerekirse bu gerekliliğe fiziksel gerekirlik (Physical alethic) ismi verilebilir. Benzer şekilde, fiziküstü (metafiziksel, tinsel) gerekirlikten de söz edilebilir. Örneğin “Aya yürüyerek gidip geldim” cümlesindeki gerekirlik fiziksel bir gerekirlik şeklidir. Benzer şekilde “Allah ikidir” cümlesi İslami açıdan non-alethic (gerekirlik şeklinden yanlış) bir önermedir.

Alethic mantığı anlamak için şu örnek verilebilir. Bir dünyadaki doğruluktan bahsediliyorsa bu alethic bir doğruluktur, buna karşılık bir bireyin doğru kabul ettiği doğruluk ise epistomolojik doğruluk olarak isimlendirilir.

2.4.3 Deontic (Zorundalık) Mantığı

Bu mantık modelinde bir önermenin zorunluluğu incelenir. Bu mantıkta basit iki operatörden bahsedilebilir (Bailhache, 1998):

O ile gösterilen ilk operatör, İngilizce “ought to” kipinden (modal) veya “obligation” kelimesinden gelmiş olup bir önermenin zorunluluğunu ifade ederken

P ile gösterilen ikinci operatör İngilizcedeki “permitted to” kipinden gelmiş olup bir önermenin izninin olduğunu gösterir.

F ile gösterilen üçüncü bir operatör bulunsa da bu operatörü aslında P operatörünün tersi olarak düşünmek mümkündür. Diğer bir deyişle aşağıdaki eşitlik doğrudur:

$$\forall x (P(x) = \neg F(x)) \text{ veya tam tersi olarak } \forall x (F(x) = \neg P(x))$$

Diğer bir deyişle F operatörü, hiç olası olunmamasını anlatmaktadır.

Yukarıda listelenen bu iki temel operatörü aşağıdaki şekilde örneklendirebiliriz:

Q: Her insan yaşamalıdır

P: Ali yaşamalıdır.

Yukarıdaki bu örneklerden anlaşılan anlam ile aşağıdaki anlamlar farklıdır:

Q: Her insanın yaşamaya hakkı vardır

P: Ali'nin yaşamaya hakkı vardır.

Örneğimizi şu şekilde değiştirebiliriz:

Q: Ali öğrencidir

R: Ali mezun olur

S: Ali yaşamalıdır

Bu önermelerden çıkarılabilecek bir sonuç olarak aşağıdaki önermeyi ele alalım

Ali öğrencidir ve mezun olması için yaşamalıdır

$$Q \wedge S \rightarrow R$$

Elbette yukarıdaki gösterime henüz bu tezin konusu olan zamansal bir yaklaşım eklenmemiştir. Dolayısıyla yukarıdaki gösterimi sadece kendi yapısı içerisinde inceleyerek buradaki Ali'nin yaşaması zorunluluğunun, Ali'nin yaşama hakkında olan zorunluluktan farklı olduğunu söyleyebiliriz.

O halde yukarıdaki model aşağıdaki şekilde yazılabilir:

$$Q \wedge O(S) \rightarrow R$$

Aynı örnekte Alinin öğrenci olması, Ali'nin inisiyatifinde olmasına karşılık, Ali'nin mezun olması için bir gerekliliktir. Dolayısıyla sonucun doğruluğu açısından aşağıdaki gösterim yapılır:

$$O(Q) \wedge O(S) \rightarrow R$$

Bu gösterimde bulunan ve aynı anda iki farklı şeye zorunluluk belirten yapı, aslında iki farklı şeye ayrı ayrı zorunluluk belirtirken bu iki önermeye birden de zorunluluk belirtmektedir.

Bu durumda gösterimimizi aşağıdaki şekilde değiştirebiliriz:

$$O(Q \wedge S) \rightarrow R$$

Son olarak yukarıdaki gösterimde “ise $\rightarrow$ ” bağlacının sol tarafının doğruluğu, sağ tarafta da bir deontic (zorundalık) gerektirir. Bu zorundalık ise aslında izin verme durumudur.

$$O(Q \wedge S) \rightarrow P(R)$$

Yukarıdaki son halimizde şunu söylemiş oluyoruz. Ali öğrenciyse(Q) ve yaşıyorsa(S) mezun olma hakkı elde eder (R).

Bu mantıksal sistemin doğruluğu açısından Q ve S üzerinde kesin olma şartı bulunurken R sonucu bir şartla bağlanmamış bunun yerine elde edilen bir hak bir izin olarak modellenmiştir.

2.4.4 Epistemic (Bilgisel) şekli

Bu şekil bilgiye dayalı olarak önermelerin (önerme) doğruluğunu sınar. Daha basit bir ifadeyle kişinin konu hakkında emin olması veya önermesini kesin olarak bildiği temellere dayandırması sorgulanır (Gochet, 2003; Hintikka, 1998; Meyer, 2001).

Örneğin “Ali borsanın yükseleceğine inanıyor” cümlesi ile ” Ali'nin bütün bilgisine dayanarak borsa yükselebilir” cümlesi arasında fark vardır. İlk cümlede bir kanaat bir düşünce dillendirilirken ikinci cümlede kişinin bilgisine dayanarak kesin bir yargı ifade edilmektedir. Yani ilk cümle örneğinde kişinin tam anlamıyla inanması beklenmez veya kişinin elinde aksini de gösteren durumlar bulunabilir. Ancak ikincisi kişinin bilgisine dayanarak kesinlik arz eder.

Örneğin “borsanın yükseleceğine inanıyorum” cümlesindeki bilgi şekli ile “Davud’un

peygamber olduğuna inanıyorum” cümlesindeki bilgi şekli arasında fark vardır. Birinci bilgi tipinde fiziksel bilgilere dayanarak ulaşılmış bir önerme bulunurken ikinci tip cümlelerde tinsel (metafiziksel) bir inanış (kanaat, yargı, önerme) bulunmaktadır.

Her iki cümle yapısında da ortaya konan yargı geçmiş bilgi ve tecrübeler üzerine inşa edilen bilgisel bir yargıdır. Dolayısıyla yargı kişiden kişiye değişmekle birlikte bütün insanlar aynı yargıda bulunsa bile doğru olduğu sonucuna varılmaz.

Bu ifadeleri şekli mantıkta (modal logic) göstermek için iki önemli işlem (operator) kullanılır.

L harfi ile gösterilen işlemde ” X ...’e inanıyor” şeklinde önermeler ifade edilir.

M harfi ile gösterilen işlemlerde ise “X’in bütün bilgisine dayanarak Y doğru olabilir” şeklinde önermeler ifade edilir.

Bazı kaynaklarda L işlemi için $\square$ sembolü ve M işlemi için de $\Diamond$ sembolü kullanılmaktadır. Bu işlemler kullanılarak önerme denklemleri çözülebilir. Yani örneğimize geri dönecek olursak Ali borsanın yükseleceğine inanıyorsa (L_p) şeklinde gösterilebilir. Benzer şekilde “Ali'nin bütün bilgisine dayanarak borsa yükselecektir” cümlesini de (M_p) olarak gösterebiliriz.

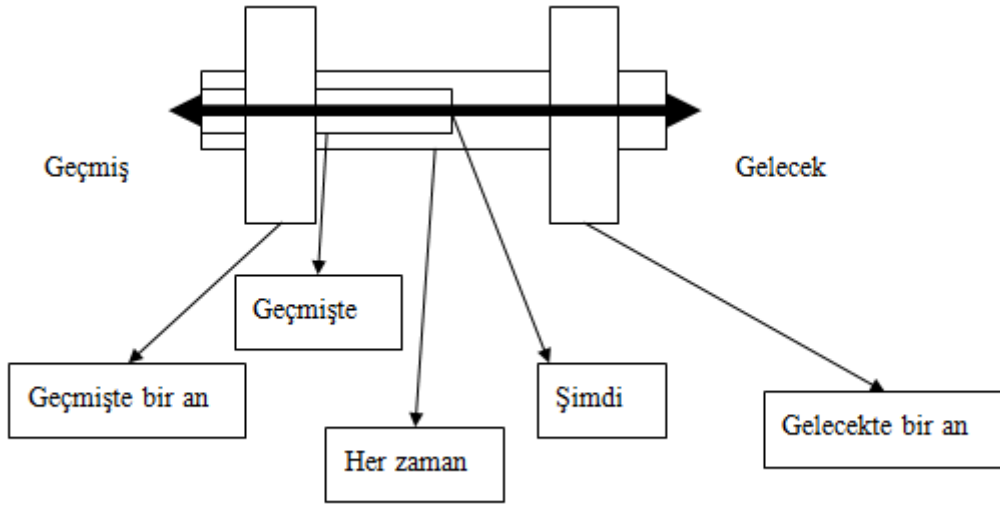
Bu iki cümle arasında kişini bilgisine dayanarak ortaya konan şekil farkı bulunmaktadır.

2.4.5 Zamansal Mantıklar

Zamansal mantıklar, bu tezin ana konusunu oluşturmakta olup daha ileride detaylıca bu mantıklardan ve çeşitlerinden bahsedilecektir. Burada kipler mantığının bir çeşidi olarak zamansal mantıklar ele alınmış ve bir zamansal mantığın, kipler mantığı gözüyle içerdiği operatörler açıklanmıştır.

Temel olarak zamanı doğrusal bir hat olarak düşünürsek, bu hattın 2 ucu ve şimdiyi gösteren bir referans noktasından bahsedilmesi gerekir.

Bu zaman çizgisi üzerinde tanımlı olan zamansal mantık operatörleri Şekil 2.2’de gösterilmiştir:



Şekil 2.2 Zamansal mantık operatörlerinin zaman çizgisi üzerinde gösterimi

Şekil 2.2’deki zaman çizgisi üzerinde gösterilen operatörler aşağıdaki şekilde sıralanabilir:

G: Her zaman

H: Geçmişten şimdiye kadar herhangi bir anda

P: Geçmişte bir an

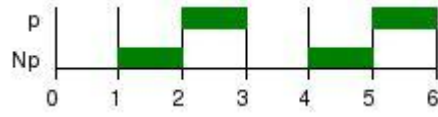
F: Gelecekte bir an

Yukarıda sıralanan bu operatörlerin ikili ilişkilerdeki durumları aşağıda açıklanmıştır:

Ayrıca zamana ait işlemler (operators) ise aşağıdaki tabloda verildikleri gibi sıralanabilir:

- Tekli işlemler (unary operators):
 - Sonra (Next)
 - Gelecek (Future)
 - Genel (Globally)
 - Hep (All)
 - Mevcut (Exists)
- İkili işlemler (Binary Operators):
 - -e Kadar (Until)
 - -den sonra (Release)

Yukarıda listelenen bu işlemleri aşağıdaki şekillerle göstermek ve örneklerle açıklamak mümkündür.

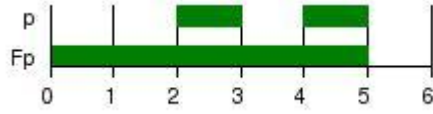


Şekil 2.3 N operatörünün p önermesi üzerindeki etkisi

Şekil 2.3’den, sonra operatörü anlaşılabilir. Buradaki 1-2 ve 4-5 aralığındaki eylemden sonra 2-3 ve 5-6 aralığındaki eylem gelmiştir.

İşlem (operatör) N harfi ile gösterilir (bazı kaynaklarda neXt kelimesindeki X harfi ile de gösterilir). Önerme (önerme, haber, predicate) p harfi ile gösterilmiştir. Dolayısıyla ilk eylem (1-2 aralığındaki eylem) bittikten sonra diğer eylem başlamıştır (2-3 aralığındaki eylem) yani p önermesi, Np’ye göre sonradır.

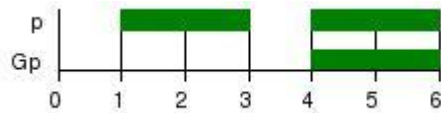
Örneğin “Şöyle bir olaydan sonra p gibi bir olay olacaktır” tipindeki cümleler bu gruba örnektir.



Şekil 2.4 F operatörünün p önermesi üzerindeki etkisi

Şekil 2.4’de, gelecek işlemi (future operator) görülebilir. İşlem F harfi ile gösterilir. Burada p önermesi Fp’ye göre ilerdedir. Bazı kaynaklarda nihayet (finally) olarak da geçmektedir.

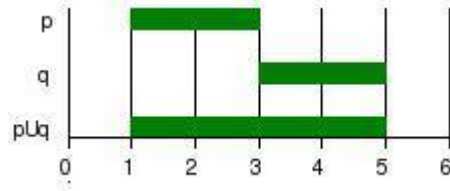
Örneğin “gelecekte bir zamanda p gibi bir şey olacaktır” şeklindeki cümleler bu gruba örnektirler



Şekil 2.5 G operatörünün p önermesi üzerindeki etkisi

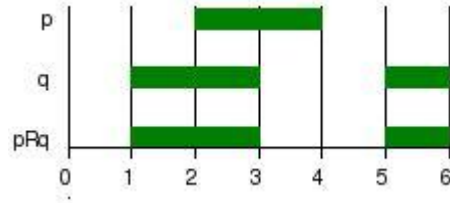
Şekil 2.5’deki, işlem genel zamanı ifade etmektedir. (Global operator) Önerme p sonradan gelen bir yolu ifade etmektedir.

Örneğin “p durumunda hep böyle olacaktır” şeklindeki cümleler bu mantık işlemine örneklerdir.



Şekil 2.6 –e kadar (until) operatörünün p ve q önermeleri üzerindeki etkisi

Yukarıdaki örnek ikili işlemlerden (binary operators) -e kadar (until) ifadesi için çizilmiştir. Yani p ve q önermelerinden p’nin bittiği anda başlayan q ifadesi pUq ile gösterilebilir.



Şekil 2.7 Sonra operatörünün p ve q önermelerine etkisi

Şekil 2.7’de ise -den sonra (release) işlemi gösterilmiştir. Burada p ve q eylemleri için p’den sonra q anlamında işleme tabi tutulması söz konusudur.

2.4.6 Kipler mantıkları arasındaki ilişkiler

Yukarıda anlatılan kipler mantığının alt parçaları olarak sayılabilecek şekiller arasında geometrik ve yapısal bağlar bulunmaktadır. Bu bağları anlamak için öncelikle yukarıda sıralanan mantık şekillerini bir tabloda gösterelim:

Çizelge 2.1 Kipler mantığındaki sembollerin karşılaştırılması

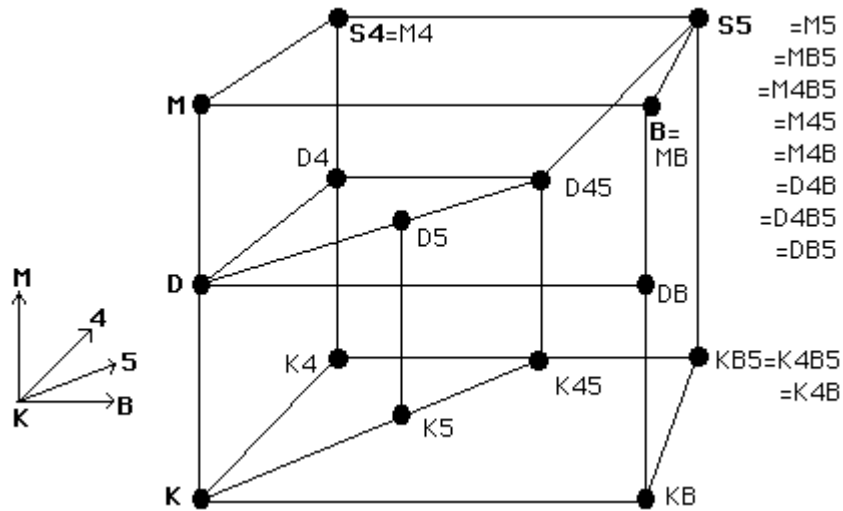
Mantık	Sembol	Anlamı
Axiomatic Mantık	$\Box x$	x gereklidir
	$\Diamond x$	x olasıdır
Zorunluluk (Deontic) Mantığı	Ox	x yapılmakla sorumludur
	Px	x 'in yapılma izni vardır
	Fx	x'in yapılması yasaktır.
Zamansal (Temporal) Mantık	Gx	Her zaman için geçerli x
	Fx	Gelecekte x
	Hx	Geçmişte, şimdiye kadar x
	Px	Geçmişte x
Bilgisel (Epistemic) Mantık	Lx	x'e inanır
	Mx	Bilgiye dayanarak x'e inanıyor

Çizelge 2-1'de gösterilen bu operatörlerin üzerinde tanımlı olan özellikler Çizelge 2-2'de toplanmıştır.

Çizelge 2.2 Zamansal mantık özelliklerinin

Aksiyom Sembolü	Aksiyom	Koşullar	İlişki
(D)	$\Box A \rightarrow \Diamond A$	$\exists u wRu$	Serilik (Serial)
(M)	$\Box A \rightarrow A$	wRw	Yansıma (Reflexive)
(4)	$\Box A \rightarrow \Box \Box A$	$(wRv \& vRu) \Rightarrow wRu$	Geçişlilik (Transitive)
(B)	$A \rightarrow \Box \Diamond A$	$wRv \Rightarrow vRw$	Simetri (Symmetric)
(5)	$\Diamond A \rightarrow \Box \Diamond A$	$(wRv \& wRu) \Rightarrow vRu$	Öklit (Euclidean)
(CD)	$\Diamond A \rightarrow \Box A$	$(wRv \& wRu) \Rightarrow v=u$	Teklik (Unique)
($\Box M$)	$\Box(\Box A \rightarrow A)$	$wRv \Rightarrow vRv$	Kaydırma Yansıma Shift Reflexive
(C4)	$\Box \Box A \rightarrow \Box A$	$wRv \Rightarrow \exists u(wRu \& uRv)$	Yoğunluk (Dense)
(C)	$\Diamond \Box A \rightarrow \Box \Diamond A$	$wRv \& wRx \Rightarrow \exists u(vRu \& xRu)$	Yakınsaklık (Convergent)

Yukarıdaki bu ilişkilerin geometrik olarak 3 boyutlu bir şekilde gösterilmesi de mümkündür.



Şekil 2.8 Zamansal mantık özelliklerinin ilişki görüntülemesi (Bellini,, 2000)

Şekil 2.8’de, ilişkiler üzerinde tanımlı olan geçişler gösterilmiştir.

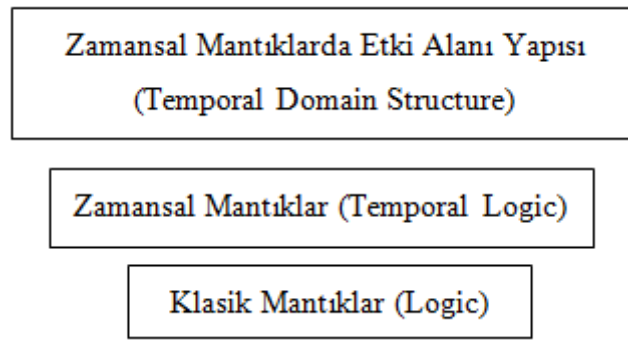
Bu geçişlerin zamansal mantık üzerine uygulanması bir sonraki bölümde örneklendirilecektir.

2.5 Zamansal Mantıklarda Etki Alanı Yapısı (Temporal Domain Structure)

Zamansal mantıkların etki alanı ile ifade edilmek istenen, bir zamansal mantığın yapısal olarak nasıl sınıflandırılacağıdır. Ayrıca bu sınıflandırma sayesinde bütün zamansal mantıklar için geçerli bir dizi özellik tanımı da yapılabilir.

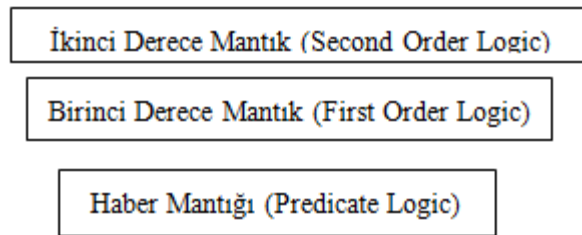
Bu bölümde önce zamansal mantıklar, kipler mantığı (modal logic) altında sınıflandırılacak ve ardından da bu mantıklar üzerinde tanımlı bir dizi özelliğin kullanılması açıklanacaktır.

Zamansal mantıklardaki etki alanı ile kastedilen seviye, Şekil 2.9'da gösterilmiştir.



Şekil 2.9 Klasik mantık ve zamansal mantıklar arasındaki seviye ilişkisi

Şekil 2.9'da , bu seviyelendirme 3 katmanda gösterilmiştir. Buna göre bir zamansal mantığın (temporal logic) seviyesi, üzerine kurulu olduğu mantık seviyesine bağlıdır. Klasik mantıkları seviyelendirirken 3 seviyeden bahsetmek mümkündür:



Şekil 2.10 Mantık seviyeleri

Yukarıdaki bu seviyelendirmede, birinci derece mantık (first order logic) için düşük seviye mantık (lower order logic) veya ikinci derece mantığa (second order logic) yüksek seviye mantık (higher order logic) isimleri de verilmektedir. Sonuçta bir zamansal mantık yukarıda gösterilen 3 seviyeden herhangi birisi üzerine kurulu bir mantık olarak düşünülebilir.

Bu durumda bir zamansal mantığın seviyesinden bahsedilmesi demek, bu zamansal mantığın

üzerine inşa edildiği mantığın seviyesinden bahsedilmesi demektir.

Ayrıca, zamansal mantıkların etki alanı yapısı ise, zamansal mantıklarda ortak olan ve ilişkileri analiz etmeye yarayan bir yaklaşımdır.

Bu yaklaşımda öncelikle temel işlem özelliklerini tanımlayabiliriz.

Örneğin iki zamansal belirleyici (quantifier) arasında tanımlı olan bir ilişkiye (relation) R ismini verecek olursak:

Geçişlilik özelliği (transitivity) : $\forall xyz. x R y \wedge y R z \rightarrow x R z$

Yansımama özelliği (nonreflexivity) : $\forall x. \neg x R x$

Doğrusallık özelliği (linearity) : $\forall xy. x R y \vee x = y \vee y R x$

Sol doğrusallık (left linearity) $\forall xyz. y R x \wedge z R x \rightarrow y R z \vee y = z \vee z R y$

Sağ doğrusallık (right linearity) $\forall xyz. x R y \wedge x R z \rightarrow y R z \vee y = z \vee z R y$

Başlama (begin) $\exists x. \neg \exists y. y R x$

Bitme (end) $\exists x. \neg \exists y. x R y$

Takip (precedence) $\forall x. \exists y. y R x$

Yoğunluluk (density) $\forall xy. x R y \rightarrow \exists z. x R z R y$

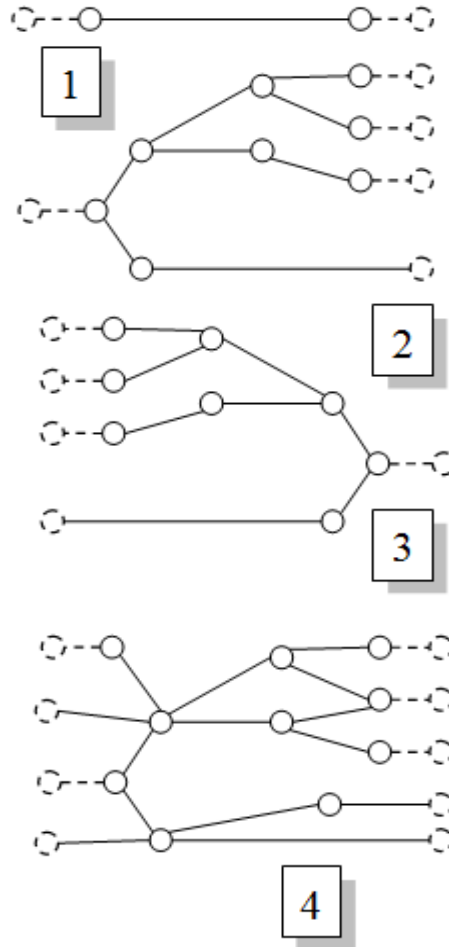
Yukarıda sıralanan özellikler bir zamansal mantığın taşıyabileceği özelliklerdir. Genellikle bir zamansal mantıktaki bütün ilişkiler geçişli, yansımaz ve doğrusaldır. Bu özellikleri açıklamaya çalışalım.

Bir zamansal mantığın başlama / bitme özelliğinin bulunması bu mantığın zamanda bir geçmiş ve bir de gelecek limitinin olması anlamına gelir. Tam tersi olarak takip (precedence / successor) ise zamanda sonsuz olduğu anlamına gelir. Yani bir zamansal mantığın geçmişte ve gelecekte bir sonunun olmaması demektir.

Bir zamansal mantığın yoğun olması, iki ilişki arasına üçüncü bir ilişki girebilmesi anlamına gelmektedir. Öte yandan kesikli olması bunun tam tersi şeklinde üçüncü bir ilişkinin, iki ilişki arasına girememesi anlamına gelir.

Yukarıda geçen doğrusallık ise 3 farklı şekilde anlaşılabilir. Zamanı doğrusal bir çizgi üzerine oturtursak iki yöne gidilen doğrusallık ve taraflardan birine gidilen doğrusallık şeklinde anlayabiliriz.

Örneğin aşağıda bu 3 farklı durum için gösterilen doğrusallık örnekleri bulunmaktadır:

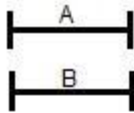
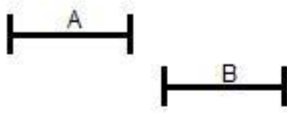
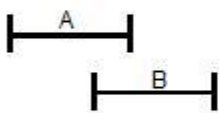
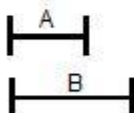
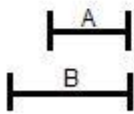
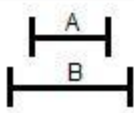


Şekil 2.11 Zamansal mantıklarda doğrusallık ve kesinlik ilişkileri,

Şekil 2.11’de 1 numara ile gösterilen şekilde tam doğrusallıktan yani iki uca doğru giden doğrusallıktan bahsedilebilir. 2 numara ile gösterilen çizimde ise sol doğrusallık gösterilmektedir. Bu gösterimde dikkat edilirse sağ tarafta bir doğrusallık bulunmamakta buna mukabil sola doğru bir doğrusal zaman çizgisi gösterilmektedir.

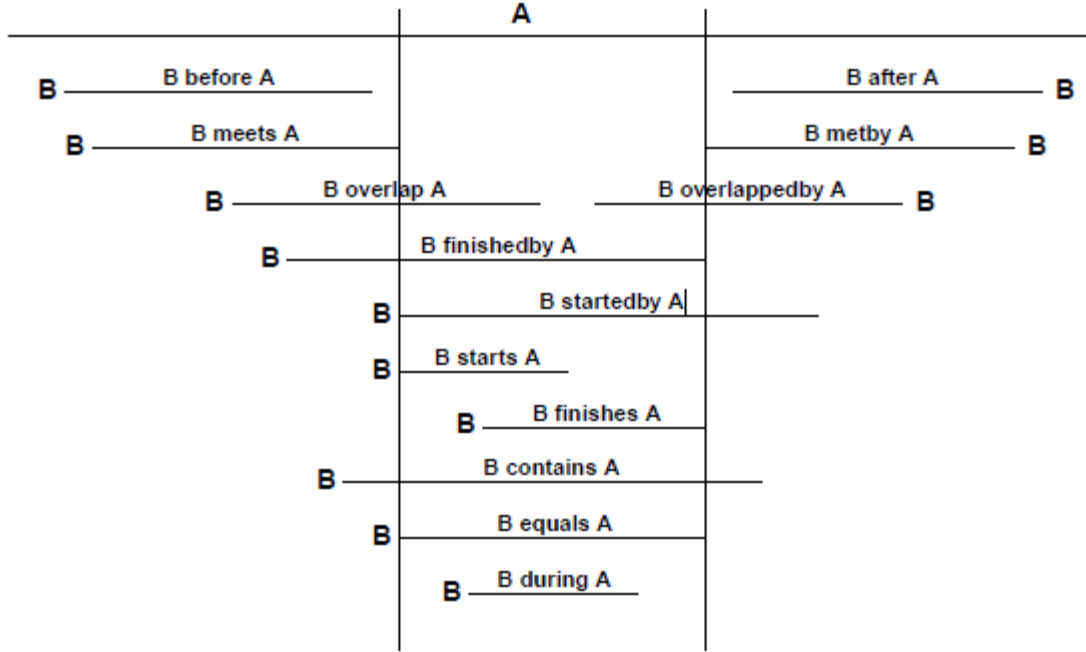
3 numaralı çizim, 2 numaralı çizimin tersine sağda doğrusallığın olduğu ve sol tarafta doğrusallığın olmadığı bir çizimdir. 4 numaralı çizim ise iki tarafta da doğrusallık bulunmayan dolayısıyla doğrusal olmayan çizimdir.

Yukarıda açıklanan zamansal mantık etki alanı yapılarına ilave olarak, zamansal mantıklarda bulunan fasıla işlemleri de ayrıca incelenmelidir. Bir zamansal mantıkta 13 temel fasıla bulunur. Bunlar aşağıdaki şekilde gösterilmiştir:

	A ile B eşit B ile A aynı zamanda
	A, B'den önce B, A'dan sonra
	A'yı B takip eder B, A tarafından izlenir
	A ile B çakışır B, A ile çakışır
	A, B'yi başlatır B, A tarafından başlatılır
	A, B'yi bitirir B, A ile biter
	A esnasında B olur B, A'yı kapsar

Şekil 2.12 Zamansal mantıklardaki olay ilişkileri ve AZM ilişki şekilleri

Şekil 2.12’de görülen bu temel 13 bağlantı hemen hemen bütün zamansal mantıklarda bulunan fasıla durumunu ifade eder.

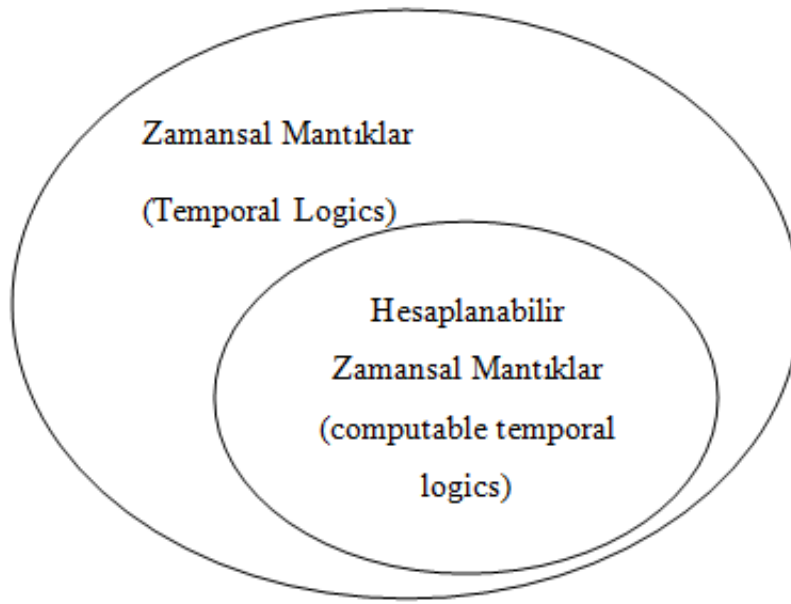


Şekil 2.1 Zamansal mantıklardaki olay ilişkileri ve AZM ilişki şekilleri

Şekil 2.12’de P. Bellini tarafından yayınlanan ve “Temporal Logics for Real-Time System Specification” başlıklı çalışmadan alınan gösterimde de bir önceki şekilde bulunan çizimin biraz daha değiştirilmiş hali görülebilir.

3. ZAMANSAL MANTIKLAR

Bir önceki bölümde, klasik mantık içerisinde zamansal mantıkların yeri açıklanmış ayrıca zamansal mantıkların sınıflandırılması yapılarak ortak özellikleri anlatılmıştır. Bu bölümde tez çalışması kapsamında incelenen bazı zamansal mantıklar bir önceki bölümde yapılan sınıflandırma çerçevesinde incelenecek ve farkları ortaya konulacaktır. Ayrıca bu bölüm için unutulmaması gereken bir nokta, zamansal mantıkların tamamının hesaplanabilir (computable) olmadığıdır. Diğer bir deyişle zamansal mantıkları hesaplanabilir ve hesaplanabilir olmayan (a-computable) olarak iki grupta incelemek gerekir.



Şekil 3.1 Zamansal mantık ve hesaplanabilir mantık ilişkisi

Şekil 3.1’de görüldüğü üzere hesaplanabilir zamansal mantıklar, zamansal mantıkların bir alt kümesi olarak düşünülebilir ve bu iki kümenin farkını hesaplanabilir olmayan zamansal mantıklar oluşturur.

Buradaki fark ise yapısal olarak zamansal mantığın üzerine kurulu olduğu klasik mantıktadır. Yani şayet zamansal mantığın üzerine inşa edildiği mantık matematiksel bir mantık yapısına sahip ise veya hesaplanabilir mantık ise, bu mantığın bilgisayarlar tarafından işlenebilir olduğundan da bahsedebiliriz.

Hesaplanabilir olmayan zamansal mantıklar bu tezin çalışma alanı dışındadır ve bu tez kapsamında bu mantıklardan bahsedilmeyecektir.

Bu tez kapsamında incelenen zamansal mantıkların seçimi sırasında mümkün olduğunca birbirinden farklı ve yeni özellikler arz eden mantıklar tercih edilmiştir.

En sade zamansal mantık olarak düşünülebilecek olan LTL (Linear Temporal Logic, Doğrusal Zaman Mantığı) ile başlanacak ve bu mantığa yapılan ilaveler ile diğer mantıklar anlatılmaya çalışılacaktır.

3.1 LTL (Linear Temporal Logic, Doğrusal Zaman Mantığı)

Bu mantık, zamanı bir çizgi olarak düşünür ve geçmişten geleceğe uzanan bu çizgi üzerinde şimdiyi bir nokta olarak kabul eder. Şayet birden fazla olayın aynı anda gerçekleşmesi söz konusu ise bu durumda bu alternatif olaylar arasında geçen doğrusal bir yolu LTL ile modellemek mümkün olabilir.

Üzerine kurulu olduğu önermeler mantığı ve doğrusal olarak modellediği önermeler itibariyle literatürde önerme zamansal mantığı (prepositional temporal logic, PTL) ismiyle de geçmektedir.

Hemen hemen bütün zamansal mantıklara temel teşkil eden bu yaklaşımda tanımlı olan bazı operatörler aşağıda sıralanmıştır (Pnueli, 1971):

Gp *Globally, p önermesinin her zaman için doğru olduğu anlamındadır.*

Fp *Future p önermesinin gelecekte herhangi bir anda doğru olduğu anlamındadır.*

pUq *Until, p önermesinin q önermesinin doğruluğuna kadar doğru olduğu anlamındadır.*

Diğer bir deyişle, p olayı q olayından öncedir.

Xp *Next p önermesinin bir sonraki önerme olduğunu belirtir.*

Yukarıdaki bu operatörleri örneklemek için aşağıdaki doğal dildeki cümleyi ele alalım:

“Ali uyuyorken Ayşe oyun oynuyordu. Ali uyanıp okula gitti.”

p: “Ali uyur”

q: “Ayşe oyun oynar”

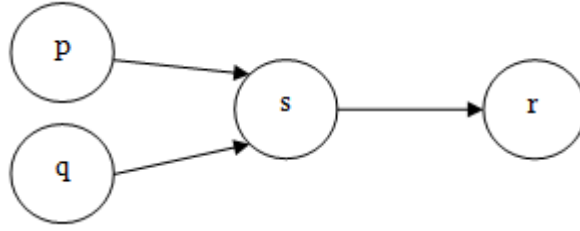
s: “Ali uyanır”

r: “Ali okula gider”

Yukarıdaki bu önermeler arasındaki ilişkilerin LTL ile gösterilmiş hali aşağıdadır:

$$p \text{ } U \text{ } s \wedge q \text{ } U \text{ } s \wedge X r$$

Yukarıdaki örnekte görüldüğü üzere zaman çizgisi üzerinde bir şimdi referansı alınmış ve bu referansa göre olaylar sıralanmıştır. Ayrıca aynı anda olan farklı olayların bir olay üzerinden birleşmesi durumunda bu olaylara giden yollar çizilerek sıralama yapılabilir.

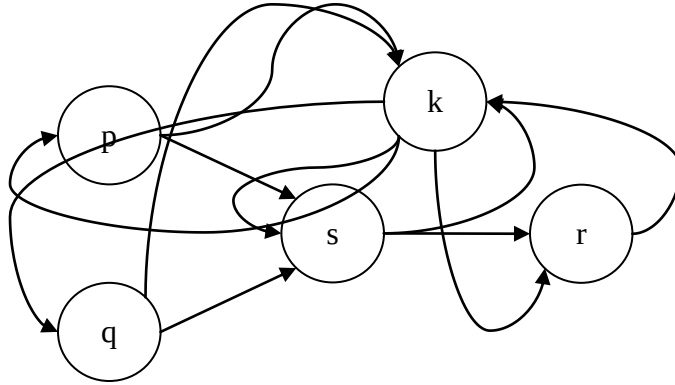


Şekil 3.2 Örnek olay ilişki şeması

Örneğimizde birinci yol $p \rightarrow s \rightarrow r$ şeklinde gidilirken, ikinci yol $q \rightarrow r \rightarrow s$ şeklinde gidilmekte ve bu iki yol aynı LTL modelinde temsil edilmektedir.

Ne yazık ki LTL ile modelleme yapılırken bağlantısız bir bilgi gösterilemez. Örneğin yukarıda verilen örneğe “Ahmet bisiklete biniyor” gibi diğer önermeler ile ilişkisiz bir önerme eklenseydi bu bilgi LTL ile modellenemezdi.

k: Ahmet bisiklete biniyor.



Şekil 3.3 Örnek olay ilişki şemasına ilave bir olay eklenmesi

Bunun sebebi yeni gelen önermenin herhangi bir yol ile mevcut önermelere bağlanamamasıdır. Her olaydan sonra veya önce bu yeni olayın gelmesi mümkündür.

3.2 Kesit Mantığı (Choppy Logic)

LTL üzerine geliştirilmiş ve LTL’de bulunan bütün operatör ve mantığı aynen taşımasına karşılık ilave tek bir kesit operatörü kullanan mantıktır. Buna göre choppy (kesit) ismi verilen ve ξ sembolü ile gösterilen bu ilave operatör olayların içinde aralık tanımlama imkanı sağlar.

Literatürde ilk defa 1986 yılında (Rosner ve Pnueli, 1986) geçen bu operatör ile aslında aralık tanımlama özelliği olan ve günümüzde de oldukça kullanışlı olan çoğu zamansal mantığa temel teşkil eden zaman aralık mantığına (interval temporal logic) giriş yapılmıştır.

Bu yöntemde LTL üzerinde bulunan operatörler ve problemler aynen taşınmakta ancak bazı olaylar alt olaylara bölünebilmektedir.

Örneğin $\sigma \equiv p \xi q$ şeklindeki bir gösterimin anlamı σ olayının iki farklı parçadan oluştuğu ve p ’nin q ’dan önce geldiği veya q ’nun, p ’nin hemen ardından geldiğidir.

3.3 CTL (Computation Tree Logic) Hesaplama Ağacı Mantığı

Bu yöntem, zamanı olaylar üzerinde çatallanmalar olan bir ağaç yapısı kullanarak modeller. Literatürde bu isimle yapılan ilk çalışmalar 1986 ve 1987 yıllarındadır (Clarke ,1986) , (Clarke ve Grumber 1987) ve (Stirling 1987).

LTL mantığından farklı olarak CTL’deki operatörlerde ilave bir A veya E ön eki bulunmaktadır. Buradaki A her zaman, E ise herhangi bir zaman anlamına gelmektedir.

Bu örneklerle birlikte CTL mantığında bulunan operatörler aşağıda gösterilmiştir:

EF, gelecekte bir zamanda

AF, gelecekteki her an için

EP, geçmişteki bir zamanda

AP, geçmişteki her zaman için

EG, herhangi bir zamanda

AG, her zaman için

Yukarıdaki bu basit operatörlerden diğer operatörleri inşa edebiliriz:

EF p $\equiv$ true EU p

$$EG p \equiv \neg (true AU \neg p)$$

$$AF p \equiv true AU p$$

$$AG p \equiv \neg (true EU \neg p)$$

$$AX p \equiv \neg EX \neg p$$

Bir önceki bölümde bulunan LTL örneğimizi CTL için genişletecek olursak aşağıdaki şekilde yeni bir gösterim elde edebiliriz:

“Ali uyuyorken Ayşe oyun oynuyordu. Ali uyanıp okula gitti.”

p: “Ali uyur”

q: “Ayşe oyun oynar”

s: “Ali uyanır”

r: “Ali okula gider”

Yukarıdaki bu önermeler arasındaki ilişkilerin LTL ile gösterilmiş hali aşağıdadır:

$$p AU s \wedge q EU s \wedge EX r$$

Ayrıca bu örneğe LTL için problem olan

k:Ahmet bisiklete biner

Önermesini eklemek artık mümkündür.

$$p AU s \wedge q EU s \wedge EX r \wedge EG k$$

Bu anlamda CTL mantığını birinci derece mantık üzerine kurulmuş (First order predicate calculus, birinci derece yüklem mantığı) kabul edebilir ve mantıkta bulunan operatörler üzerinde bir etki alanı (domain) tanımlama imkanı verdiği söylenebilir.

3.4 BTTL (Branching Time Temporal Logic) Zaman Çatallanmalı Zamansal Mantık

Bu zamansal mantık yaklaşımında, gelecek modelinin çatallanması söz konusudur. (Ben-Ari, 1983)

Bu zamansal mantığın çıkışında önemli rol oynayan problem, zaman kavramının belirli (deterministic) olmayışdır. Yani gelecek bilinemez ve gelecekle ilgili modellemelerde

kesinlik arz eden bir doğrusallık hatalı olur.

BTTL burada devreye girerek belirsiz gelecek modellemelerinde alternatif model imkanı sunar ve gelecekteki olayların olasılıksal olarak ifade edilmesine imkan verir.

Burada BTTL için gelecekte olabilecek olayların gösterildiği ilave operatörler devreye girer. Bu operatörler aşağıda açıklanmıştır:

π zaman alternatiflerinin geçtiği bir yol ve

x bu yol üzerinde bir olay olmak üzere

- $\forall \square$ Bütün π yolları ve bütün $x \in \pi$ olayları için.
- $\exists \square$ En az bir π yolu ve bu yoldaki bütün $x \in \pi$ olayları için
- $\forall \Diamond$ En az bir π yolu ve bu yoldaki bütün $x \in \pi$ olayları için
- $\exists \Diamond$ En az bir π yolu ve bu yoldaki en az bir $x \in \pi$ olayı için

Anlamında kullanılmaktadır. Bu sayede gelecekteki herhangi bir alternatif veya bütün

3.5 ITL (Interval Temporal Logic) Aralıklı Zaman Mantığı

Literatürde ilk olarak 1983-1986 yılları arasında çalışılmış olan ITL temel olarak LTL üzerine kurulmuş ve kesikli, doğrusal, geçmişli belirli, geleceği ise belirsiz bir mantıktır. (Halpem, 1983; Moszkowski ve Manna, 1984; Moszkowski, 1985; Moszkowski, 1986).

Bu mantık, sayısal işaret işleme konusunda kullanılmak üzere bir gösterim ihtiyacından doğmuştur. Dolayısıyla bu mantık zamanlar arasında herhangi bir metrik ölçümden bahsedilemez. Bunun yerine olaylar arasındaki ilişkilere dayalı bir gösterim izlenir.

Ayrıca bu mantıkta kullanılan operatörler, şimdiye kadar değişik zamansal mantıklarda geçen operatörlerden çok farklı değildir. Bu operatörleri aşağıdaki şekilde sıralayabiliriz:

- $\square$ Her zaman (Always)
- $\Diamond$ Bazı zamanlarda (Eventually)
- O Sonraki (Next)
- ξ Kesit operatörü (Chop)

Yukarıdaki operatörler daha önce bahsedilen zamansal mantıklarda da kullanılmıştır. ITL

mantığının getirdiği en büyük fark ise bu mantıkta olaylar arasındaki zamanın olay sayısı cinsinden belirtilebilmesidir. Bunun için mantıkta kullanılan ilave bir fonksiyon olan “Len” kullanılmıştır. Şimdiye kadar bahsedilen mantıklarda olmayan ve Len(n) şeklinde bir tam sayı olarak iki zamansal olay arasında geçen olayların sayısını gösteren fonksiyonun kullanımı aşağıdaki örnekte gösterilmiştir:

“Ali uyandı, giyinip dışarı çıktı ve okula gidene kadar kitap okudu. “

Yukarıdaki örnek cümlede geçen olayları sıralayalım:

P: uyandı

Q: giyindi

R: okula gitti

S: kitap okudu

Yukarıdaki bu olayları örneğin aşağıdaki şekilde gösterirsek ITL mantığına uygun ve doğrulanabilir bir gösterim elde etmiş oluruz.

$P \xi R \rightarrow$ Ali uyandı ve okula gitti

Yukarıdaki gösterimde kesit operatörü kullanılmıştır. Bu operatör, ITL açısından oldukça önemlidir çünkü ITL mantığında daha önce farklı mantıklarda bahsedilen Önce – Sonra anlamında veya –e kadar anlamında olan Until operatörü (örneğin LTL için geçerli bir operatördür) bulunmaz. Bunun yerine iki olayı öncelik – sonralık ilişkisinde modelleyen kesit operatörü kullanılır.

Yukarıdaki gösterimi açarak aşağıdaki sonuç da elde edilebilir ve ITL mantığında doğrudur:

$P \xi Q \xi S \xi R \rightarrow \text{true}$

Yukarıdaki gösterimi ilk haline geri indirirsek aşağıdaki şekilde bir ilave yapmamız mümkündür:

(1) $\text{Len}(2) (P \xi R) \rightarrow \text{true}$

Burada ifade edilen, P ve R olayları arasında iki adet olay olduğudur.

Yine yukarıdaki örnekten aşağıdaki sonuç da çıkarılabilir:

(2) $\text{Len}(1) (P \xi S) \rightarrow \text{true}$

Artık, yukarıdaki iki gösterimden (1) numaralı gösterim ile (2) numaralı gösterim karşılaştırıldığında S olayının R olayından önce olduğu sonucunu çıkarabiliriz.

Gerçekten de örneğimizde, Ali, okula vardığında kitap okuma işlemi bitmiştir.

3.6 Aralık Mantığı (Interval Logic)

Literatürde ilk defa 1982, 1983 (Mellar, 1982; Schwartz, 1983), yıllarında ismi geçen bu mantık, şimdiye kadar anlatılan mantıklardan farklı olarak olay modellemesini zaman aralıkları ile yapmaktadır.

Aşağıda aralık mantığının temel operatörü gösterilmiştir:

$[\vartheta] \alpha$

Yukarıdaki bu gösterimde kullanılan ϑ sembolü bir aralığı belirtirken α sembolü LTL için tanımlı olan herhangi bir önerme ifade etmektedir.

Örneğin aşağıdaki doğal dilde verilen cümleyi ele alalım:

“Ali bütün gün uyudu”

Bu önermenin gösterimi aşağıdaki şekilde yapılabilir:

p: Ali uyudu

t : bütün gün

olmak üzere önermenin IL gösterimi aşağıdaki şekildedir:

$[t] p$

Yukarıda bulunan bu aralık mantığı gösterimize LTL öğeleri eklersek ve aşağıdaki iki ayrı önermeyi birbirinden ayırmak istersek

(1) Ali bütün gün aralıksız uyudu

(2) Ali bugün bir ara uyudu

Gösterimleri aşağıdaki şekilde farklılaştırmak mümkündür:

(1) $[t] \Box p$

(2) $[t] \Diamond p$

Görüldüğü üzere zamansal mantıklara özgü operatörlerin aralık mantığı ile birlikte

kullanılması mümkündür.

Ayrıca tanımlanan aralık ve önerme arasında aşağıdaki ilişkiler olabilir:

- Öncesinde (Before)
- Sonrasında (After)
- Sırasında (In)

Bu ilişkiyi göstermek için $\Rightarrow$ sembolü kullanılır. Sembol kullanılmaması *sırasında* anlamındadır.

Yukarıdaki örneğimizi bu eklentiler doğrultusunda ilerletecek olursak,

p: Ali uyudu

t: bugün

olmak üzere

Ali bugün uyudu $[t]p$

Ali bugünden sonra uyudu $[t\Rightarrow] p$

Ali bugünden önce uyudu $[\Rightarrow t]p$

Şeklinde üç farklı aralık/önerme ilişkisi gösterilebilir.

3.7 Genişletilmiş Aralık Mantığı (Extended Interval Logic)

Bu mantık sistemi, bir önceki bölümde anlatılan Aralık Mantığının (IL, Interval Logic) yetersiz kaldığı özellikle gerçek zamanlı sistemler (real time systems) için geliştirilmiştir (Melliar-Smith 1987). Gerçek zamanlı sistemlerde karşılaşılan bir problem, bir olayın başlangıç ve bitiş zamanlarının kesin olmasını gerektirir.

Bu durumda aralık mantığı yetersiz olur, çünkü bir olayın bir aralıktan önce veya sonra olması kesinlik arz etmez.

Örneğin “Ali bugünden önce uyudu” önermesi için “Ali dün uyudu” önermesi de doğrudur. “Ali geçen hafta uyudu” önermesi de doğrudur.

Dolayısıyla gerek aralık, gerekse bu aralıkla önerme arasında olan öncelik ve sonralık ilişkisinde metrik bir değer bulunmamaktadır.

EIL eklentisi ise tam bu noktada devreye girerek aralık / önerme ilişkisini kesin zaman metrikleri ile bağlar.

Örneğin aşağıdaki gösterimi ele alalım:

$$\Box [t \Rightarrow \text{bitiş}] (p \wedge \text{başlangıç})$$

Yukarıda eklenen yeni “bitiş” ve “başlangıç” değerleri, bu modelde bulunan olayın doğruluğu için gereken başlangıç zaman koşulunu ve aralığın geçerli olduğu bitiş zamanını göstermektedir.

Kısacası bir zaman aralığı ile bir önerme ilişkilendirildiğinde önermenin doğruluğu için bir zamansal koşulun gerçekleşmesi şartı öne sürülebilir.

Olayın bitişinde ise bu zamansal şart zaten gerçekleşmiş olmak zorundadır.

3.8 Rtl (Real Time Interval Logic) Gerçek Zamanlı Aralık Mantığı

Aralık Mantığının Rozik ve Gorlik tarafından 1989 yılında genişletilmiş halidir. (Rozik ve Gorlic 1989)

Gerçek zamanlı sistemlerde karşılaşılan bir problem, bir olayın başlangıç ve bitiş zamanının, diğer olayların başlangıç ve bitiş zamanlarını etkilemesidir. Bu durumda bir olayın başladığı ve bittiği zamanın, zamansal mantık üzerinde kesin olarak ifade edilmesi beklenir.

RTIL bu problemi ilave bir operatör kullanarak çözmektedir.

$$\Box [\odot p \Rightarrow t] (\odot \text{başlangıç} \Rightarrow \odot \text{bitiş})$$

Yukarıdaki notasyonda p önermesinin t zamanında olduğu, “başlangıç” ve “bitiş” zamanları arasında gerçekleştiği anlatılmaktadır.

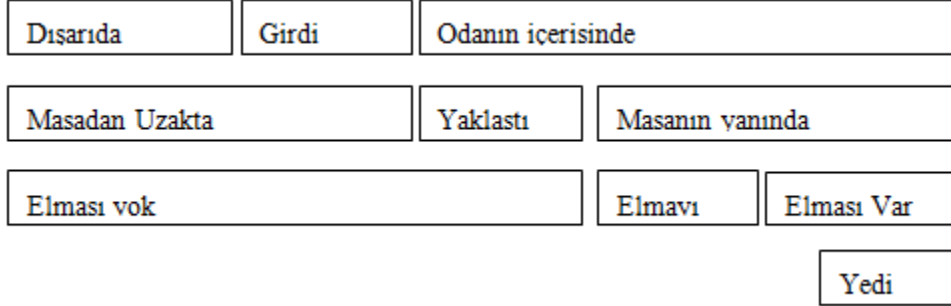
3.9 Allen Zamansal Mantığı (Allen Interval Logic)

Allen Zamansal Mantığı (AZM), olaylar arasındaki bağlantıları ve olayların sıraları üzerinde çalışır. Örneğin “olay A, olay B’den öncedir” veya “olay A ile olay B aynı zamandadır” gibi işlemleri kullanarak olaylar arasındaki ilişkileri modellemeye çalışır.

AZM üzerindeki en temel işlemler, iki parametresi bulunan ve iki olay arasındaki ilişkiyi betimlemeye yarayan işlemlerdir.

Bu durum aşağıdaki cümle üzerinden daha rahat anlaşılabilir:

“Ali, odaya girdikten sonra, masanın üzerindeki elmayı yedi”



Şekil 3.4 Örnek olay durum ilişkisi

Yukarıdaki bu cümledeki olaylar sıralanacak olursa şekil 3.4’deki gibi bir durum ortaya çıkacaktır.

Şayet cümleler üzerinde daha detaylı bir işleme yapılırsa, Ali’nin ilk başta odanın dışında olduğunu biliyoruz, ayrıca odaya girdikten sonra masanın üzerindeki elmayı alması için, masaya yaklaşması gerektiği de çıkarılabilir. Son olarak Ali’nin masadan elmayı alana kadar, aslında elinde elma olmadığı da doğrudur. Bu durumlar modele eklenecek olursa, Şekil 2’de gösterilen tablo ortaya çıkar.

Yukarıdaki cümleden örneğin Ali’nin odanın dışındayken elması olmadığı veya Ali’nin masanın yanındayken bir elması olduğu veya Ali’nin elmayı yerken odada olduğu ve masanın yanında olduğu sonuçları çıkarılabilir.

Yukarıdaki olaylar zincirini modellemek için AZM üzerinde bazı değişkenleri tanımlayalım. Örneğin girme (G) eyleminin, odanın dışında (OD) olma durumunu gerektirdiğini ve odaya girme eyleminden sonra odanın içinde (OI) olma durumuna geçildiğini söyleyebiliriz. Benzer şekilde masaya yaklaşma (MY) eylemi, masadan uzakta olma (MU) durumunu masanın yakınında olma (MYO) durumuna çevirmiştir. Ayrıca elmayı alma (EA) eylemi de elması yok (EY) durumunu elması var (EV) durumuna çevirmiştir. Bütün bu eylemler aslında yemek (Y) eyleminin ön koşullarıdır.

Yukarıdaki bu durum ve şartları aşağıdaki şekilde bağlamak mümkündür.

$Meets(OD,G) \wedge Meets(G,O\dot{I}) \wedge During (G,MU) \wedge During (MU,O\dot{I}) \wedge Meets(MU,MY) \wedge$
 $Meets(MY,MYO) \wedge During(MU,EY) \wedge During(EA,O\dot{I}) \wedge During (EA,MYO) \wedge Meets(EY,EA) \wedge$
 $Meets(EA,EV) \wedge During(Y,EV) \wedge During (Y,MYO) \wedge During(Y,O\dot{I}) \wedge Meets(EA,Y)$

Yukarıdaki model, çıkarılabilecek bazı zamansal ihtimalleri modellemektedir. Bunun yanında bir okuyucu, örnek cümlemizden, Ali'nin henüz elmayı almadan, aslında aç olduğu, sonucu da çıkarabilir. Bu durumda modelimiz aşağıdaki şekilde olmalıdır:

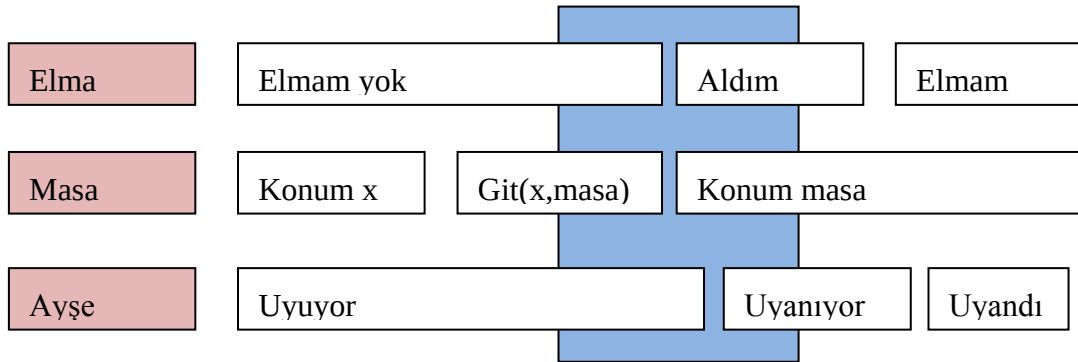
$Occurs (aç, EY) \wedge Holds (aç, EA) \wedge Meets (aç, Y)$

Yukarıdaki modele göre, Ali'nin henüz elması yokken, Ali açtı, ve Ali elmayı aldığında da açtı ve Ali'nin aç olma durumu, Ali'nin elmayı yemesine kadar sürdü; anlamı çıkarılabilir.

Yukarıda anlatılan AZM için aşağıdaki örnekler görselleştirilmiştir:

Masanın yanına gidip elmayı alırken Ayşe uyandı. Ayşe uyanırken masanın yanına gidip elmayı almıştım.

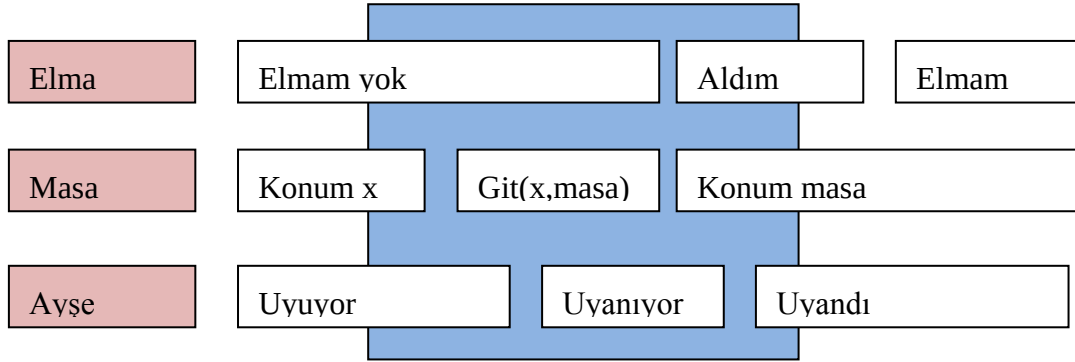
“Elmayı almam Ayşe 'yi uyandırdı.”



Şekil 3.5 Örnek olay durum ilişkisi

Elmayı almak için masanın yanına giderken ayşe uyandı. Ayşe uyanırken elmayı almak için masanın yanına gidiyordum.

“Masanın yanına gitmem Ayşe 'yi uyandırdı”



Şekil 3.6 Örnek olay durum ilişkisi

Yukarıdaki bu işlemleri de kapsayan AZM işlemleri aşağıdaki şekilde sıralanabilir:

- Before (x,y) or After (y,x) {önce veya sonra}
- Overlaps (x,y) or Overlapped (y,x) {üst üste binmek}
- Meets(x,y) or MetBy(y,x) {bir olayın bitmesi ile başkasının başlaması}
- Contains(x,y) or During (x,y) {bir olayın bir diğerini zamansal olarak kapsaması}
- Starts(x,y) or StartedBy(y,x) {bir olayın bir diğerini başlatması}
- Ends(x,y) or EndedBy(y,x) {bir olayın bir diğerini bitirmesi}
- Equals(x,y) {iki olayın tam olarak aynı zamanda olması}

3.10 Reichenbach Zamansal Mantığı

Reichenbach Zamansal Mantığı, bu tez çalışması kapsamında incelenen ve bu raporda anlatılan tek doğal dil işleme kökenli mantıktır (REICHENBACH, 1947). Bu mantığın kurulması sırasında üç ana unsur ele alınmıştır:

1. olayın geçtiği zaman (E)
2. olayın yazıldığı veya anlatıldığı zaman (S)
3. referans zamanı (R)

Örneğin aşağıdaki cümleyi ele alalım:

“Ali gelip gerçekleri anlattığında çoktan talimat vermiştim.”

Yukarıdaki cümlede geçen fiilleri ayıklayacak olursak:

(“(Ali gelip gerçekleri anlattığında)R (çoktan talimat vermiştim)E.”)S

Yukarıda gösterilmek istenen bütün cümlelerin şu anda anlatıldığı veya okunduğudur. Ayrıca talimat verme eylemi tam olarak olayın geçtiği zamanda gerçekleşmiştir. Son olarak Alinin gelip gerçekleri anlatması eylemi olayın zamanına referans alınan bir olaydır. Yani bu cümlede bu iki olay arasında bir bağlantı kurulmuştur.

Dolayısıyla Reichenbach analizine göre yukarıdaki cümlede 3 farklı zamanın sıralaması aşağıdaki şekilde olacaktır:

$$E < R < S$$

Yani olayın geçtiği zaman, referans zamanından çok önce olmuştur, benzer şekilde olayın referans zamanı ise olayın anlatıldığı şimdiki zamandan öncedir.

Yukarıdaki örnekten yola çıkarak Reichenbach 13 farklı zaman dizilimi alternatifi getirmiştir. Bu dizilimler Çizelge 3-1'deki gibi bir tabloda toparlanabilir.

Çizelge 3.1 Reichenbach zaman ilişkileri

İlişki	Reichenbach Zaman İsmi	İngilizce Zaman İsmi	Türkçe Karşılığı
$E < R < S$	Anterior past	Past perfect	Hikaye Geçmiş Zaman
$E = R < S$	Simple past	Simple past	Geçmiş Zaman
$R < E < S$			
$R < S = E$	Posterior past		Gelecek Zamanın Mazisi
$R < S < E$			
$E < S = R$	Anterior present	Present perfect	Rivayet Geçmiş Zaman
$S = R = E$	Simple present	Simple present	Geniş Zaman
$S = R < E$	Posterior present	Simple future	YOK
$S < E < R$			YOK
$S = E < R$	Anterior future	Future perfect	YOK
$E < S < R$			
$S < R = E$	Simple future	Simple future	Gelecek Zaman
$S < R < E$	Posterior future		

Çizelge 3-1'de 13 farklı zaman gösterilmektedir. Bu zamanların hepsinin İngilizce veya Türkçe için birebir karşılıkları yoktur ancak konuşmacı (yada yazar) farklı şekillerde bu zamanları ifade etmektedir.

4. İŞLENEBİLİR ZAMANSAL DİLLER

Bu bölümün amacı, literatürde geçen ve bilgisayarlar tarafından işlenebilir dillerin açıklanmasıdır. Bu doktora tezinin de konusu olan bilgisayarlar tarafından işlenebilir zamansal diller, bir önceki bölümde bulunan zamansal mantıkların bir veya birkaçının aynı anda kullanılmasıyla üretilmiş ve iyi şekillendirilmiş dillerdir.

4.1 STTL

STTL -> A Standard Timetabling Language (Standart bir Zaman Çizelgeleme Dili)

Bu dil ilk olarak Jeff Kingston tarafından zaman çizelgeleme problemlerine bir girdi standardı elde etmek için önerilmiştir. (Kingston, 1999) Dilin çıkışında kullanılan problem bir lisede haftalık ders programlarının düzgün yerleştirilmesini hedefliyordu. Nesne yönelimli (Object oriented) bir dil olarak önerilen STTL dilinde varlıklar arası miras ilişkisi bulunmaktadır. Örneğin:

```
class TEACHER inherit RESOURCE
max_class_load:INTEGER
unavailable_times:SET[TIME]
cats:SET[TEACHER_CATEGORY]
functions
defects:SEQ[DEFECT]=
(
defect_check(TeacherClash, clashes) +
defect_check(TeacherOverload, overload) +
defect_check(TeacherDayOverload, day_overload)
)
end
```

Yukarıdaki STTL alıntısında Öğretmen (TEACHER) ile kaynak (RESOURCE) arasında miras ilişkisi kurulmuştur. Yukarıdaki tanımlamadan sonra aşağıdaki varlığın dilde tanımlanması mümkündür:

```
TEACHER("Knott", 22, {}, {English})
```

Buna göre Knott isminde bir öğretmen haftalık azami 22 saatlik ders yükü ve herhangi bir ders engeli bulunmadan English kategorisinin altında tanımlanmıştır.

STTL dilinin diğer özelliklerinden birisi de kaynak tanımlamasıdır. Örneğin dersin yapılacağı sınıfın, ders için kullanılacak malzemelerin birer kaynak olarak tanımlanması ve dolayısıyla zamanlamaya dahil edilmesi mümkündür.

```
r01= ROOM("r01", {DramaRoom})
r02 = ROOM("r02", {ScienceLab})
```

```
Yr7C = STUDENT_GROUP("Yr7C")
Yr7K = STUDENT_GROUP("Yr7K")
```

Örneğin yukarıda sınıf ve öğrenci grupları birer kaynak olarak tanımlanmıştır. Zaman modellemesi içinde alternatifler sunan STTL dilinde örneğin aşağıdaki şekilde bir zaman tanımlaması yapmak mümkündür:

```
TIME({2}, INTERVAL(1, 14), {Mon}, {INTERVAL(t(9, 30), t(12, 30))})
```

Yukarıdaki tanımlamada, 2. dönem için 1'den 14. haftalara kadar Pazartesi günleri saat 9.30 ile 12.30 arası kastedilmiştir. Dolayısıyla tekrarlı bir olayı ifade etmektedir.

Sonuç olarak zaman çizelgelemesine göre hazırlanmış olan bu dilde kesin tarihlerin belirlenmesi zaman aralıklarının ifade edilmesi mümkün olmakla beraber bağıl zaman ifadeleri ve zamanın belirsiz olması durumları düşünülmemiştir. Örneğin A ile B olayları olsun ve tek bilinenin A olayının B olayından sonra olması durumu olduğunda bunu göstermek bu dil ile mümkün değildir.

4.2 Html Time

HTML -> Hyper Text Markup Language (Hiper Metin İşaretleme Dili)

TIME -> Timed Interactive Multimedia Extensions (Zaman etkileşimli çoklu ortam uzantıları)

Microsoft, Compac/DEC ve Macromedia firmaları tarafından W3C'a gönderilen bir dil önerisidir. Dil, XML üzerine kurulu SMIL (Synchronized Multimedia Integration Language , Eş güdümlü çoklu ortam uyarlama dili) üzerine kuruludur ve bu dildeki amaç ses, görüntü

veya film gibi çoklu ortam uygulamalarının web üzerinde bir standart olan HTML dilinin üzerine zamanlama ile birlikte gömülmesidir. Buna göre örneğin Internet Gezini 5.5 sonrasında microsoft basit bir ses kaydının sayfada istenildiği zamanda çalınmasına imkan verebilmektedir.

```
<HTML>
<HEAD>
<STYLE>
.time {behavior: url(#default#time2);}

</STYLE>
</HEAD>
<BODY>
<H1 CLASS="time" BEGIN="0" DUR="11" TIMEACTION="style"
STYLE="Color:Red;">timeAction</H1>
<P>Bu satır derhal görüntülenir ve altına yeni satırlar
eklenir</P>

<P CLASS="time" BEGIN="2" DUR="5" TIMEACTION="display">2
saniye sonra görüntülenir.</P>
<P CLASS="time" BEGIN="4" DUR="5" TIMEACTION="display">4
saniye sonra görüntülenir.</P>
<P CLASS="time" BEGIN="6" DUR="5" TIMEACTION="display">6
saniye sonra görüntülenir.</P>
<P>Son satır.</P>
</BODY>
</HTML>
```

Yukarıdaki kodda, bulunan 5 satırdan ilk ve son satırlar sayfanın yüklenmesinde görüntülenirken, 2. 3. ve 4. satırlar 2şer saniye ara ile görüntülenir. Bunu sağlayan P etiketinin içindeki sınıf stillerine (class style) ilgili zaman kodlarının yüklenmiş olmasıdır. İlginç olan diğer bir özellik ise bu yazıların sadece 5şer saniye ekranda görüntülenmeleridir.

4.3 Owl Time

Gelişen zamanlama ihtiyaçları ile birlikte zamanın gösterimi ve formüllemesi de bir ihtiyaç haline gelmiştir. Örneğin yapılan her siparişte, siparişin zamanının tutulması, basit bir kiralama işleminde veya bilet satış işleminde yapılan işlemin hangi tarih ve saatler için yapıldığının tutulması artık sıradan birer gereksinim haline gelmiştir. Bu amaçla doğmuş olan OWL Time, Web Ontology Language (Baş harflerinin sırası okunmasını kolaylaştırmak için değiştirilmiş ve OWL olarak kısaltılmıştır) altında zaman göstermek amacıyla XML üzerine kurulu bir dil olarak tasarlanmıştır. (Jerry, 2006) Projenin daha önceki ismi DAML-Time olarak geçmekteydi.

Dilin yapısında varlıklar TemporalEntity (zaman varlığı veya fânî varlık olarak isimlendirilmektedir) ve her TemporalEntity altında Instant (kesin, ânî) veya Interval (aralık, zaman aralığı) olarak iki farklı varlık bulunur.

```
:Instant
    a      owl:Class ;
    rdfs:subClassOf :TemporalEntity .

:Interval
    a      owl:Class ;
    rdfs:subClassOf :TemporalEntity .

:TemporalEntity
    a      owl:Class ;
    rdfs:subClassOf :TemporalThing ;
    owl:equivalentClass
        [ a      owl:Class ;
          owl:unionOf (:Instant :Interval)
        ] .
```

Bu durum yukarıdaki RDF/XML gösterimi ile temsil edilmiştir. Dikkat edilirse Instant ve Interval sınıfları (Class), TemporalEntity sınıfının (Class) birer alt sınıfıdır (subclass).

4.3.1 Sürelerin İfade edilmesi

OWL-Time kullanılarak sürelerin ifade edilmesi de ayrı bir problemidir. Örneğin bir eylemin süresi 1 gün ve 2 saat veya 26 saat veya 1560 dakika olabilir ve bütün bu zamanlar aynı süreye işaret etmektedir. Dolayısıyla sağlıklı bir zaman gösteriminde Yıl, Ay, Hafta, Gün, Saat, Dakika, Saniye olmak üzere 7 farklı zaman gösterimi gerekmektedir. Ayrıca bu zaman gösterimini belirli eden ilave bir gösterici ile toplam 8lik gösterici kullanılmaktadır. Bu durum aşağıdaki sınıf yapısında gösterilmiştir:

```
:DurationDescription
```

```

    a          owl:Class ;
    rdfs:subClassOf
        [ a          owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :seconds
        ] ;
    rdfs:subClassOf
        [ a          owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :minutes
        ] ;
    rdfs:subClassOf
        [ a          owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :hours
        ] ;
    rdfs:subClassOf
        [ a          owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :days

```

```

    ] ;
    rdfs:subClassOf
        [ a          owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :weeks
        ] ;
    rdfs:subClassOf
        [ a          owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :months
        ] ;
    rdfs:subClassOf
        [ a          owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :years
        ] .

```

Yukarıda gösterilen sınıf `DurationDescription` yani Süre Tanımlayıcı sınıfının yapısıdır. Bu sınıftan üretilen bir örnek aşağıda verilmiştir:

```

duration
    a          :DurationDescription ;
    :seconds 20 ;
    :hours 5 ;
    :days 15 .

```

Yukarıdaki örnekte 15 gün 5 saat ve 20 saniyelik bir zaman dilimi temsil edilmiştir.

Yukarıda Süre Tanımlayıcısı (`DurationDescription`) sınıfının tanımına dikkat edilirse her zaman birimi için ayrıca bir de `maxCardinality` (azami sayısalılık) tanımı yapılmış ve bütün zaman birimleri için bu değer 1 atanmıştır. Bunun OWL dili açısından anlamı bu değerlerin sayısal olarak istedikleri gibi atanabilecekleridir. Ancak OWL dili bilindiği üzere

varlıkbilimsel bir dildir ve her zamanın bütün bu birimleri ifade etmesi beklenemez. Örneğin üniversite eğitiminin saniyeler dakikalar cinsinden ifadesi beklenmedik bir durumdur, genelde üniversite eğitimi 2, 3, 4, 5, 6 gibi yıllar ile ifade edilir. Veya bir yemek süresi dakika ve saatler mertebesinde bu sürenin yıllar ile ifade edilmesi beklenmedik bir durumdur. İşte OWL-Time bu gibi durumlarda diğer zaman birimlerinin kapatılmasına imkan verir ve yapılması gereken basitçe bu zaman birimlerinin maxCardinality bilgisini 0 yapmaktır:

```
:Year
```

```

a          owl:Class ;
rdfs:subClassOf :DurationDescription ;
rdfs:subClassOf
    [ a          owl:Restriction ;
      owl:cardinality 1 ;
      owl:onProperty :years
    ] ;
rdfs:subClassOf
    [ a          owl:Restriction ;
      owl:cardinality 0 ;
      owl:onProperty :months
    ] ;
...

rdfs:subClassOf
    [ a          owl:Restriction ;
      owl:cardinality 0 ;
      owl:onProperty :seconds
    ] .

```

Örneğin yukarıda bir yıl sınıfı tanımı yapılmıştır ve yıl dışındaki bütün zaman birimlerinin cardinality (sayısallık) bilgisi 0 yapılmıştır.

4.3.2 Tarih ve Saat kavramı

OWL-Time kullanılarak belirli bir tarihin veya saatin ifade edilmesi de gerekmektedir. Örneğin 10:30 bir saat ifade etmektedir buna karşılık 26 Mart 2008 ise bir tarih ifade etmektedir. Dolayısıyla herhangi bir zamanı ifade etmek için öncelikle hangi birimden ifadenin yapılacağına karar verilmelidir. Yani ifade edilecek olan zaman bir dakikayı mı bir günü mü yada bir saati mi ifade etmek amacındadır belirlenmelidir (“saat 10” önermesi bir saat ifade ederken “26 Mayıs” bir günü ifade etmektedir). Bu belirlenen birime unitType (birim tipi) denilirse geriye zaman diliminin ifade edilmesi sorunu kalmaktadır. Yani istenilen saat acaba hangi zaman dilimine göre (örneğin GMT, EST, PST) belirtilmektedir. Son olarak tarih ve saati ifade eden zaman birimleri aşağıdaki şekilde tanımlanabilir:

```
:DateTimeDescription
    a          owl:Class ;
    rdfs:subClassOf
        [ a          owl:Restriction ;
          owl:cardinality 1 ;
          owl:onProperty :unitType
        ] ;
    rdfs:subClassOf
        [ a          owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :second
        ] ;
    rdfs:subClassOf
        [ a          owl:Restriction ;
          owl:maxCardinality 1 ;
          owl:onProperty :minute
        ] ;
    rdfs:subClassOf
        [ a          owl:Restriction ;
```

```

        owl:maxCardinality 1 ;
        owl:onProperty :hour
    ] ;
rdfs:subClassOf
    [ a      owl:Restriction ;
      owl:maxCardinality 1 ;
      owl:onProperty :day
    ] ;
rdfs:subClassOf
    [ a      owl:Restriction ;
      owl:maxCardinality 1 ;
      owl:onProperty :dayOfWeek
    ] ;
rdfs:subClassOf
    [ a      owl:Restriction ;
      owl:maxCardinality 1 ;
      owl:onProperty :dayOfYear
    ] ;
rdfs:subClassOf
    [ a      owl:Restriction ;
      owl:maxCardinality 1 ;
      owl:onProperty :week
    ] ;
rdfs:subClassOf
    [ a      owl:Restriction ;
      owl:maxCardinality 1 ;
      owl:onProperty :month
    ] ;

```

```

rdfs:subClassOf
    [ a      owl:Restriction ;
      owl:maxCardinality 1 ;
      owl:onProperty :year
    ] ;
rdfs:subClassOf
    [ a      owl:Restriction ;
      owl:maxCardinality 1 ;
      owl:onProperty :timeZone
    ] .

```

```

:hasDateTimeDescription
    a      owl:ObjectProperty ;
    rdfs:domain :DateTimeInterval ;
    rdfs:range :DateTimeDescription .

```

Görüldüğü üzere yukarıdaki gösterimde haftanın günleri veya yılın günü gibi kavramlar da ilave edilerek bir tarih/saat ifadesine yer verilmiştir. İlk eklenen alt sınıf (subclass) bir zaman birimini gösterirken son alt sınıf ise zaman dilimini (timeZone) göstermektedir.

Bu bilgiler paralelinde aşağıdaki toplantı varlığını inceleyelim:

```

:meetingStart
    a      :Instant ;
    :inDateTime
        :meetingStartDescription ;
    :inXSDDateTime
        2006-01-01T10:30:00-5:00 .

```

```

:meetingStartDescription
    a      :DateTimeDescription ;

```

```

:unitType :unitMinute ;
:minute 30 ;
:hour 10 ;
:day 1 ;
:dayOfWeek :Sunday ;
:dayOfYear 1 ;
:week 1 ;
:month 1 ;
:timeZone tz-us:EST ;
:year 2006 .

```

Yukarıda aynı toplantı varlığı için iki farklı gösterim bulunmaktadır. İlk gösterimde XSD zamanı denilen XML Scheme Definition kelimelerinin kısaltılmışı olan ve XML üzerinde tanımlama yapılan dilin biçimine uygun olarak yazılmış gösterime yer verilmiştir. Aynı tarih değeri altta bizim tanımladığımız tarih saat yapısına uygun olarak tekrar gösterilmiştir. Buna göre toplantı dakika cinsinden bir zamana işaret etmekte olup zaman birimi EST'dir ve toplantının zamanı 1/1/2006 saat 10:30'dur.

4.3.3 OWL Time kullanarak zaman aralığı gösterimi

Bilindiği üzere zaman gösterimlerinde her zaman kesin ve belirli zamanları ifade etmek mümkün değildir. Örneğin bir kargo firmasının teslim süresi olarak 2-3 gün arasında demesi, teslimin en az 2 en çok 3 günde gerçekleşeceğini ifade etmektedir. Bu durum OWL Time kullanılarak aşağıdaki şekilde ifade edilebilir:

```

:DeliveryDuration
    a          owl:Class ;
    rdfs:subClassOf
        [ a          owl:Restriction ;
          owl:cardinality 1 ;
          owl:onProperty :maxDeliveryDuration
        ] ;

```

```

rdfs:subClassOf
    [ a          owl:Restriction ;
      owl:cardinality 1 ;
      owl:onProperty :minDeliveryDuration
    ] .

```

```

:maxDeliveryDuration
    a          rdf:Property ;
    rdfs:domain :DeliveryDuration ;
    rdfs:range  time:Interval .

```

```

:minDeliveryDuration
    a          rdf:Property ;
    rdfs:domain :DeliveryDuration ;
    rdfs:range  time:Interval .

```

Yukarıdaki gösterimde bir teslim tarihi için iki alt bilgi girilmiş bunlardan birisi minDelivery (asgari teslim) diğeri ise maxDelivery (azami teslim) süresi olarak belirlenmiştir. Bu belirlemenin yanında 2 günlük ve 3 günlük sürelerde aşağıdaki şekilde tanımlanabilir:

```

:Interval2Days
    a          owl:Class ;
    rdfs:subClassOf time:Interval ;
    owl:subClassOf
        [ a          owl:Restriction ;
          owl:hasValue P2D ;
          owl:onProperty
time:durationDescriptionDataType
        ] .

```

```

:Interval3Days
    a          owl:Class ;
    rdfs:subClassOf time:Interval ;
    owl:subClassOf
        [ a          owl:Restriction ;
          owl:hasValue P3D ;
          owl:onProperty
time:durationDescriptionDataType
        ] .

```

Artık bu tanımlamalardan sonra kargo firmasının teslim süresi tanımlanabilir:

```

:Cargo2-3dayDuration
    a          owl:Class ;
    rdfs:subClassOf :DeliveryDuration ;
    rdfs:subClassOf
        [ a          owl:Restriction ;
          owl:allValuesFrom :Interval3Days ;
          owl:onProperty :maxDeliveryDuration
        ] ;
    rdfs:subClassOf
        [ a          owl:Restriction ;
          owl:allValuesFrom :Interval2Days ;
          owl:onProperty :minDeliveryDuration
        ] .

```

yukarıda görüldüğü üzere kargo firmasının teslim süresi 2 günlük zaman diliminde asgari ve 3 günlük zaman diliminde azami süreler olarak tanımlanmıştır.

OWL-Time kullanarak ne yazık ki belirsiz iki eylemin göreceli zamanlarını tanımlamak mümkün değildir. Örneğin A olayı B olayından sonra olmuştur önermesi OWL-Time kullanımıyla gösterilemez.

4.4 TTML

Zaman çizelge işaretleme dili (Timetabling Markup Language (TTML)), XML üzerine kurulmuştur. MathML üzerine kurulu zaman çizelgeleme problemlerinin çözümünde kullanılmaktadır.

TTML üzerinde kullanılan formülleme dili küme teorisine dayandırılabilir. Örneğin MathML üzerinde kullanıcılara tamamen yeni semboller ile bu sembollerin ifade ettiği yeni fonksiyon ve formülleri tanımlama imkanı sağlanır. Bu durum MathML kullanan TTML dilinin de küme teorisini desteklemesini sağlamaktadır. Bu sayede kullanıcı esnek bir şekilde kendi kıstaslarını belirleyebilmekte ve zaman çizelgelemesini özelleştirebilmektedir.

Bilindiği üzere zaman çizelgelemesi NP Complete (NP Tam) problem ailesindendir ve zaman çizelgelemesindeki amaç uygun zaman aralıklarına uygun olaylar kümesini yerleştirmektir. Gelişmekte olan şartlara göre zaman çizelgelemede aranan özellikler ve kurallar kümesi de gelişmekte dolayısıyla problemin karmaşıklığı artmaktadır. Bu konuda bir standart bulunmaması ayrıca araştırmacıların karşılaştırma imkanını kaldırmakta ve mukayeseli sonuçlar çıkarılmasını zorlaştırmaktadır. Bu konuda ilk çalışma Andrew Cumming tarafından ICPTAT'95 'de yapılmış ve çalışmanın sonucu olan dile SSTL ismi verilmiştir. Beklenildiği üzere SSTL dili de bir standartlaşmayı sağlayamamıştır çünkü elde bulunan problemlerin bu dile çevrilmesi tam anlamıyla mümkün olmamıştır. Günümüzde neredeyse her araştırmacı kendi gösterimini kullanmakta ve bu konuda bir standartlaşma henüz sağlanamamaktadır.

5. İŞLENEBİLİR ZAMANSAL MANTIKLARIN KULLANIM ALANLARI

Bu tezin de hedeflediği üzere bir zamansal mantığın işlenebilir olması, o zamansal mantığın bilgisayarlar tarafından kullanılabilir olmasını doğurur. Bunun sebebi bilgisayarların ilgili zamansal mantık modeli üzerinde işlem yaparak çeşitli neticelere varabilmesidir.

İşlenebilir zamansal diller, aşağıdakilerle sınırlı olmamakla birlikte genel olarak üç ana maddede toplanabilir:

- Soru cevaplama
- Kronoloji çıkarımı (veya olay sıralama)
- Metin işleme

5.1 Soru Cevaplama

Doğal dildeki bir metin içerisinde yine doğal dildeki bir sorunun cevaplanması anlamına gelen soru cevaplama çalışmalarının bir kısmı da zamansal sorulardan oluşmaktadır.

Örneğin “Ali okula geldiğinde Ayşe kitap okuyordu ve ders çoktan başlamıştı” cümlesi için sorulabilecek zamansal sorular aşağıda verilmiştir:

- Ali okula ne zaman geldi?
- Ayşe derste kitap okuyor muydu?
- Ders ne zaman başlamıştı?

Yukarıda sıralanan bu soruların ve benzerlerinin amacı olayların zamansal olarak bir referansa dayandırılmasıdır. Bu sorulara beklenen cevaplar ise yine farklı bir olay şeklinden veya bir zamansal ifade olarak verilir.

5.2 Kronoloji Çıkarımı

Bu grup çalışmalar bir doğal dildeki metin üzerinde çalışarak bu metinde bulunan olayları sıralamayı hedefler. Genelde bir zaman çizgisi üzerinde olaylar sıralı olarak gösterilir ve sonuçta çıkarılan bilgi görsel veya iyi şekillendirilmiş bir dokümandır.

Kronoloji çıkarımı sırasında amaç olayların hangi sıra ile gerçekleştiği ve hangi olayın hangi diğer olayların etkisinde kaldığını bulmaktır.

Kronoloji çıkarımı doğal dil işleme çalışmaları yanında planlama ve zamansal gösterim

gereken yerlerde de kullanılır.

Bu anlamda akış diyagramları, UML Sıra Diyagramları (UML Sequence Diagram) ve Gantt Çizgeleri (Gantt Chart) gibi gösterim yöntemlerini sıralayabiliriz. Bu yöntemlerin hepsinde zamansal birer mantık kullanılmakta ve kullanılan bu zamansal mantığa dayalı olarak olaylar sıralanmaktadır.

5.3 Metin İşleme Çalışmaları

Metin işleme çalışmalarına örnek olarak aşağıdaki konular gösterilebilir:

- Metin özetleme (Mani, 2001)
- Ontoloji (Udo, 2002)
- Arama motorları (Penev, 2006)

Yukarıdaki sıralanan ve metin işleme çalışmalarına örnek olan konular metin içerisinde bulunan zaman kavramı ve bu zaman kavramı etrafında şekillenen olayları hedef almaktadır.

Örneğin, metin özetleme çalışmaları sırasında bir metinde bulunan olayların bir kısmının eksiltilmesi istendiğinde, bu eksiltilecek olayların, mevcut olay zincirini bozmaması gerekir.

“Ali, pencereden baktığında güneşin üzerine ışıdığı, pırıl pırıl bir bahar havası gördü ve Ayşe’ye dönerek buradan gitmek istediğini söyledi”

Yukarıdaki cümle özetlenerek aşağıdaki şekle getirilebilir:

“Ali, Ayşe’ye dönmüş ve buradan gitmek istediğini söyledi”

Yukarıdaki ilk cümlede bulunan bazı kelimeler ikinci cümlede çıkarılmıştır.

Bu eksiltme işlemi sırasında olaylar arasında kurulan bağlantı sonucunda, dönmek fiilinin, söylemek fiilinden önce olduğu çıkarılmıştır. Bu çıkarım sonucunda metinden bazı olaylar çıkarılarak metin değiştirilmiştir.

Ontoloji çalışmalarında da benzer şekilde metin analizi yapılır ve olaylar zamanlarına göre sınıflandırılır.

Örneğin bir orduya aşağıdaki şekilde bir emir verilmesi durumunda ontolojik hata bu sayede tespit edilebilir.

“Ordu Ankaraya intikalinden on dakika sonra Trabzondaki çatışmalara müdahale etsin”

Yukarıdaki bu cümlede ontolojik bir problem bulunmaktadır. Ordu seviyesindeki bir askeri gücün 10 dakika içerisinde Ankara'dan Trabzon'a gidemeyeceği bilinmektedir. Bu aradaki süre ve ordunun kabiliyeti karşılaştırılırken sürenin çıkarımı, olayların hangi sırayla yapılmak istediği ve aradaki vakit yine işlenebilir zamansal mantıklar ile yapılabilir.

İşlenebilir zamansal mantıkların bir diğer kullanım alanı ise arama motorlarıdır. Günümüzde kullanılan arama motorlarının büyük kısmı artık aranan metin üzerinde doğal dil işleme çalışmaları yapmaktadır.

Henüz Türkçe için arama motorları destek sunmamakla birlikte, İngilizce için, bu tezin yazıldığı sırada en çok arama yapılan arama motoru olan Google üzerinde aşağıdaki soru sorulduğunda:

“When Micheal Jackson died?” (Micheal Jackson ne zaman öldü?)

Google ilk sırada sonuç olarak aşağıdaki sonucu bulup döndürebilmektedir.

“On *June 25, 2009*, American singer *Michael Jackson* died after he suffered cardiac arrest at his home in the Holmby Hills neighborhood in Los Angeles...”

(25 haziran 2009'da Amerikan şarkıcısı Michael Jackson, Los Angeles'a komşu olan Holmby Hills'daki evinde kalp rahatsızlığı yüzünden öldü.

Görüldüğü üzere sonuç bulma işlemi sırasında aranan metindeki zaman bilgileri üzerinde doğal dil işleme yapılmakta ve sonuç üretimi buna göre yapılmaktadır. Ayrıca indekslenen web siteleri üzerinde de benzer işlemler yapılarak sonuç üretim başarısı arttırılmaktadır.

6. TIMEML

Bir önceki bölümde açıklanan ve bilgisayarlar tarafından işlenebilen dillerin dışında özellikle doğal dil işleme çalışmalarında kullanışlı olan TimeML dili daha detaylı olarak bu bölümde anlatılacaktır. Bu bölüm aynı zamanda Türkçe olarak TimeML dilinin kullanımı ve şekillendirilmesi konusunda ilk yapılan çalışmadır. Bu anlamda bu tezin bu bölümü tek başına bir yardımcı döküman olarak da kullanılabilir.

TimeML diline göre metnin işaretlenmesine yönelik anlatma kurallarını tarif etmektedir. TERQAS anlatma çalışma grubunun bugüne kadarki tartışmalarının ve deneme anlatmalarının yanı sıra TANGO atölyesi esnasında eklenen değişiklikleri sunmaktadır. (TERQAS, 2002)

Bu bölüm, TimeML etiketlerinin (XML öğeleri) neler olduğunu ve bunların nasıl anlatılacağını açıklamaktadır. Aynı zamanda her bir etiket için anlamlarının neler olduğunu belirtmekte ve etiketle anlamları için BNF tanımlaması sağlamaktadır.

6.1 Timeml Etiketlerinin Ve Anlamlarının Açıklaması

Bu bölüm kapsamında etiketler açıklanacak ve her etiket için örnekler verilecektir.

6.1.1 <EVENT> etiketi

“Olaylar”, gerçekleşen veya ortaya çıkan durumları kapsayıcı bir terim olarak görülebilir. Olaylar anlık (bkz. Örnekler 1-2) olabilir yada belirli bir süre (bkz. Örnekler 3-4) devam edebilir. Aynı zamanda olaylar, bir şeylerin elde edildiği yada gerçek olduğu (bkz. Örnek 5) durumları yada halleri tarif eden yüklemeler olarak görülebilir.

Ancak, belirtici yüklemelerin tümü işaretli değildir. ‘STATE’ sınıfına ait olan olaylar için alt-bölüm’e bakınız.

1. Ferdinand Magellan, Portekizli bir kaşif, baharat arayışı içerisinde adalara ilk ulaşan oldu.
2. Orada Cumartesi günü taptaze bir lav, gaz ve enkaz akıntısı ortaya çıktı.
3. Yerel Aeta aborjinleri dahil 11.024 kişi, 18 felaket kurtarma merkezine boşaltıldı.
4. Bugün erken saatlerdeki bir telefon görüşmesinde “Ciddi bir patlama bekliyoruz” dedi.
5. İsrail, birkaç yüz bin gaz maskelik kılığın ardından yurtdışından daha fazla maske satın almak için mücadele ediyor.

6.1.1.1 EVENT'lerin nasıl anlatılacağı

Olaylar, zamanlı yada zamansız fiiller (bkz. Örnek 1 ve 2), adlandırmalar (bkz. Örnek 3), sıfatlar (bkz. Örnek 4), yüklem cümlecikleri (bkz. Örnek 5) yada önerme cümleleri (bkz. Örnek 6) yoluyla ifade edilebilir:

1. Cumartesi günü orada taze bir lav, gaz ve enkaz akıntısı ortaya çıktı.
2. Başbakan, ülkesinin katkıda bulunduğu binlerce gaz maskesi için kendisine teşekkür etmek üzere Hollanda Başbakanı'nı çağırdı.
3. İsrail, Yahudi devleti muhtemel bir Irak saldırısına karşı tamamen hazır oluncaya kadar Birleşik Devletler'den Irak'a yönelik askeri bir saldırıyı ertelemesini isteyecek.
4. Altı yüzyıldır uykuda duran bir Filipinler yanardağı, kızgın gazlarla, kalın küllerle ve ölümcül enkazlarla patlamaya başladı.
5. Mordechai, Yediot Ahronot'a her gün "Hazırlıklı olmamamız için hiçbir gerekçe yok" dedi.
6. Aeroflot Airbus'ın içindeki 75 kişinin tamamı öldü.

Yukarıdaki cümlelerde tüm "işartlenebilir"ler, etiketli değildir. İlk örnekte, örneğin, ne akma ne de Cumartesi işaretlenmemiştir.

Resmen basit olayların anlatılması (yukarıdaki örnekler 1, 3, 4 ve 6) doğrudandır. Ancak, resmen karmaşık olaylar birtakım bağlamlarda dizinsel olarak süreksiz olabilir.

1. Hazırlıklı olmamamız için hiçbir gerekçe yok.
Tam hazırlıklı olmamız için hiçbir gerekçe yok.
2. Hazır olmamızı kesinlikle göz önüne alacaklar.
Bunu kesinlikle göz önüne alacaklar.

Süreksiz dizinleri etiketlemekten kaynaklanan sorunlardan kaçınmak için aşağıdaki stratejileri uygulayınız:

Eğer olay, fiilli bir cümleyle ifade edilirse (mücadele ediyor, satın almak, bildirildi), OLAY etiketi sadece aşağıdaki örneklerde kalın harflerle işaretlenen başlığa uygulanacaktır:

Çin, yurtdışından daha fazla maske satın almak için mücadele ediyor.

Hiçbir yaralanma bildirilmedi.

Özel sektör, özel bir kurum kurabilir.

Eğer bize saldırılmış olsaydı, ...

Eğer ana fiil, yukarıdaki üçüncü örnekte olduğu gibi yardımcı kipin yada aşağıda olduğu gibi olumsuz takının kapsamı dahiline girerse,

OLAY etiketi sadece ana fiile uygulanır. Kip ve/veya olumsuzlama, sözkonusu olayla bağlantılı MAKEINSTANCE (ÖRNEK YAP) etiketinde temsil edilecektir.

- Eğer ana yüklem, cümleli bir fiilse, sadece fiilli olan bölümü (ve takısı) işaretlenecektir:
- Önümüzdeki hafta ilave dağıtım merkezleri kurulacaktır.
- Eğer olay, durumsal fiil ve ana fiil içeren fiili bir kümeyle temsil ediliyorsa her iki fiil de bağımsız olaylar olarak etiketlenecektir. Durumsal fiillerin örnekleri şunlardır: başlamak, durmak, sona ermek, korumak, v.s.
- Eğer durumsal fiilin öncesinde (aşağıdaki son iki örnekte olduğu gibi) yardımcı biçimler geliyorsa, yukarıdaki ilk kriter de uygulanacaktır. Aşağıdaki örneklerde, altını çizmek fiilli kümenin sınırını kaldırırken köşeli parantezler her bir OLAY etiketinin kapsamının sinyalini vermektedir:

Özel sektör, özel bir kurum [kurmaya] [başladı].

Eğer ABD diğer ülkelerin politikalarına [müdahale etmeyi] uzun süre önce [bırakmış] olsaydı,...

Başkan Reagan'ın 600 gemi planına muhtemelen yeni bir yapımla [cevap vermeye] [başlamış] olurlardı.

- Eğer olay, diğer öğelerle [belirtenler, tamamlayanlar, değiştirenler] birlikte NP'nin başı olarak görünen bir adlandırmaysa sadece baş öge işaretlenecektir:

Genç endüstrinin hızlı büyümesi de birçok yönüne polislik yapmak isteyen düzenleyicileri çekiyor.

- Eğer olay, (aşağıdaki altı çizili kısımlarla gösterildiği gibi) birtakım hafif yüklemelerin eşlik ettiği bir adlandırmaysa, her iki ögeyi de işaretleyeceğiz.

Geçen hafta içinde birkaç Irak-taraftarı gösteri gerçekleşti.

Bizim hazır olmamızı kesinlikle göz önüne alacaklar.

Her iki öge de etiketlidir çünkü hem fiiller hem de adlı başlıklar, farklı türde olay bilgisiyle ilişkilidir. Etiketli iki olay, TLINK etiketindeki relType anlamıyla ÖZDEŞ (IDENTICAL) olarak ilişkilendirilecektir.

- Eğer olay, yüklemli bir maddeyse, sadece yüklemli öge (aşağıdaki örneklerde sıfat yada nominal) etiketlenecektir. Bu, neyin işaretleneceğine karar vermek için başlıklılık testinin uygulanmasıyla yapılır:

1. Eğer yüklemli öge ENDOCENTRIC (MERKEZ İÇİ) ise (yani, eğer başlığı varsa), sadece etikete başlık atacağız. Aşağıdaki örneklerde, yüklemli ögenin altı çizilmişken başlığı kalın karakterle işaretlenmiştir:

[Tam olarak hazırlıklı] olmamız için hiçbir neden yok.

Eğer, her şeye rağmen [hazır] olmazsak, Birleşik Devletler'den operasyonu geciktirmesini isteyeceğiz.

James Pustejovsky birkaç yıllığına [Lingo Motors'un CTO'su] idi.

2. Eğer yüklemli öge EXOCENTRIC (DIŞ MERKEZLİ) ise (yani, eğer tek bir başlığı yoksa) TimeML etiketi içindeki bütün ifadeyi işaretleyeceğiz.

75 kişinin tümü, sabah 9:00'da uçaktaydı.

3. Eğer olay, önermeli cümleyle ifade ediliyorsa, biz de önermeli cümlemin başlıklılığına dayalı strateji kullanacağız. Bu nedenle, PP'nin dış merkezli olduğu aşağıdaki örnek, gösterildiği gibi etiketlenecektir:

Aeroflot Airbus uçağındaki 75 kişinin tümü öldü.

- Nedensel yüklem, nedenin işaretini verdiği olayın nedenin mantıksal konusu olabilecek olaydan ciddi biçimde farklı olup olmadığına dair konuları öne çıkarabilir. Örneğin, aşağıdaki

Yağmurlar, sele yol açtı.

yağmur ,neden olaydır ve bunlar da bu şekilde mi etiketlenmelidir?

Üç durum ayrıştırılabilir:

Örnek 1 OLAY, OLAYA neden olur.

[Yağmurlar], [sele] [yol açtı].

Örnek 2 KİMLİK OLAYA neden olur.

Ali [yangına] [neden oldu].

Örnek 3 OLAY. Söz işaretçisi OLAY

Topa [vurdu] ve havaya [yükseldi].

Şimdilik, TimeML'deki nedensel yüklemeler için aşağıdaki çözümü kabul ediyoruz.

1. Eğer anlatıcı, cümledeki tezler arasındaki nedensel ilişkileri tasvir eden bir yüklemle karşılaşır, ya
 - (a) Eğer (mantıksal) özne, ifadeyi gösteren bir olaya (örn., savaş, sel, yağmur, toplantı, v.s.), bunu bir olay olarak işaretleyiniz (YUKARIDA ÖRNEK 1)
 - Ya da,
 - (b) Eğer (mantıksal) özne tek bir kimlik ise (örn., Ali, kadın, şirket, v.s.) buna hiçbir şey yapmayınız. (YUKARIDA ÖRNEK 2)
2. Nesneyi olay (sel, yangın) olarak işaretleyiniz.
3. Nedensel ilişkiyi gösteren yüklemi (neden olunan) bir olay olarak işaretleyiniz.
4. Eğer Örnek 1'de olduğu gibi öznenin anlattığı bir olay varsa, özne olayla olan nedensel ilişkiyi belirleyen IDENTITY (KİMLİK)'in relType'ıyla TLINK başlatınız.
5. BEFORE (ÖNCEDEN) relType ile nedensel yüklem olayıyla nesne konumuyla bağlantılı olay arasında TLINK başlatınız.

Bu çözüm, nedensel anlamlarında aşağıdakiler gibi fiiller için kabul edilmelidir: neden olmak, kaynaklanmak, yol açmak, üretmek, tehlikeye sokmak, hazırlamak, sevk etmek, vesile olmak, üretmek, ortaya çıkarmak, çıkarmak, güvence altına almak.

Yukarıdaki ÖRNEK 3'te anlatıcı, relType BEFORE yada değil TLINK için ifade

işaretçisini ve sinyal olarak belirleme seçeneğine sahiptir. Anlatıcı, olaylar arasındaki TLINK’i belirleYEBİLİR, fakat buna GEREK YOKTUR.

Yukarıdaki örneklerin tam anlatımlı versiyonları, bölüm 4.3’te sunulmaktadır.

6.1.1.2 Neyin EtiketlenMEyeceği

- Jenerikler, bunları dondurmak soru cevaplamada yararlı olabilse de en azından şimdilik etiketlenmeyecektir.

Örneğin, aşağıdaki cümlelerde olay kelimeleri etiketlenmeyecektir:

Siyasi yolculuk için kurum jet uçaklarının kullanılması meşrudur.

İşletmeler, devlet düzenleyicileri dahil hiç kimse kendilerini izleyemeyecek şekilde İnternette çok hızlı doğmaktadır.

Yahudilerin birbirlerini öldürmesi yasaklanmıştır.

Burada her bir olaylar sınıfına bir özellik atfedilmiştir yada bir olaylar sınıfıyla kimlikler dizisi mensupları arasında bir ilişki vurgulanmıştır, fakat hiçbir tek olay örneği yada olay örnekleri dizisi, bu sınıfta zamanın içine yada diğer geçici yerleştirilmiş olaylara yerleştirilmemiştir.

Zaman içerisinde açıkça yerleştirilmiş ve dolayısıyla açıkça etiketlenebilir olan aşağıdaki gibi tek bir olayı ifade eden önceki saf jenerik ve diğerleri gibi örnekler arasında durmak,

7 Haziran’da, Bay Sununu, Fiber Materials Inc. tarafından temin edilen bir jet uçağını kullandı.

aşağıdaki gibi daha az net örneklerdir

Bay Sununu, siyasi yolculuk için kurumsal uçakların düzenli kullanımına başvurdu.

Bu gibi cümleler, tipik faaliyet düzenlerini ifade etmektedir, fakat belirli olaylara açıkça atıfta bulunmaz. Şimdilik, bu gibi örnekler etiketlenmeyecektir.

- Açıkça zamansal olarak yerleştirilen fakat tamamlayıcıları jenerik olan bağımlı fiillerin kendisi etiketlenmemelidir. Örneğin, bir jeneriği niteleyen zamanlı bir raporlama fiili etiketlenmemelidir:

Rabbi, Yahudilerin birbirlerini öldürmekten menedildiğini söyledi.

Bu durumda, bağımlı önerme etiketlenmediğinden bahsi geçen fiil de etiketlenmeyecektir.

- Bağlı olduğu fiilin temin ettiğinin ötesinde hiçbir bilgi sağlamayan bir olay adlandırmasının etiketlenmesine gerek yoktur. Örneğin:

Gazete raporları diyordu ki ...

Bu bağlamda *diyordu* bir olay olarak etiketlenir ve *raporlar* etiketlenmez.

6.1.1.3 OLAY etiketi için BNF

attributes (nitelikler) ::= eid sınıfı

eid ::= e<integer>

class (sınıf) ::= 'REPORTING' | 'PERCEPTION' | 'ASPECTUAL' | 'I_ACTION' | 'I_STATE' | 'STATE' | 'OCCURRENCE'

stem (kök) ::= CDATA

6.1.1.4 OLAY Nitelikleri

a. **Olay Kimlik numarası (eid)** Seçmeli-olmayan nitelik. Her bir olay özel bir kimlik numarasıyla belirlenmelidir. Birtakım dizilere bir OLAY etiketinin her tahsis edildiğinde anlatma aracıyla bu, otomatik olarak tahsis edilecektir.

b. **Sınıf** Seçmeli-olmayan nitelik. Her bir olay, aşağıdaki sınıflardan birine aittir. Örnekler, belirli bir türde fiilleri gösterirken, sözkonusu fiilin her bir meydana gelişinin aynı türde bir olayı ifade etmesine gerek olmadığını not ediniz. Yani, olay sınıfı itibarıyla fiiller kapalıdır.

- **BİLDİRME:**

Olayları rapor etmek, bir şeyler beyan eden, bir olayı anlatan, bir olay hakkında bilgi veren, v.s. bir kişinin yada örgütün eylemini tarif eder. Bazı örnekler: söylemek, bildirmek, anlatmak, açıklamak, belirtmek:

Punongbayan, 4.795-foot yüksekliğindeki yanardağın 1,800 dereceye kadar gaz kızdırdığını söyledi.

Hafta sonu boyunca hiçbir yaralanma bildirilmedi.

- KAVRAMA:

Bu sınıf, başka bir olayın fiziksel kavrayışını içinde bulunduran olayları içerir. Bu gibi olaylar, tipik olarak şu gibi fiillerle ifade edilir: görmek, seyretmek, bakış atmak, gözlemlemek, görmek, duymak, dinlemek, kulak misafiri olmak.

Şahitler, Birmingham polisine bir adamın koştuğunu gördüklerini söylerler. Bir şahit “Orada binlerce ufak patlama duyabilirsiniz” dedi.

- DURUMSAL

İngilizce ve Fransızca gibi dillerde durumsal yüklem, olay tarihinin farklı yönlerine odaklanan dilbilimsel bir aracı vardır:

1. Başlama: başlamak, start vermek, başlangıç vermek, yola koyulmak, ortaya çıkmak, başa geçmek, kaynaklanmak, girişmek.
2. Yeniden başlama: yeniden başlamak, yeniden başlatmak, yeniden alevlendirmek (mecazi anlamda)
3. Sona erdirmeye: durdurmak, iptal etmek (I_ACTION sınıfına da bakınız), son bulmak, durdurmak, sona ermek, devam etmemek, kesintiye uğramak, bırakmak, terk etmek, engellemek, kırmak, koyuvermek, iptal etmek, tasfiye etmek.
4. Tırmanma: son bulmak, tamamlamak.
5. Devam etme: devam etmek, korumak, süregitmek, ilerlemek, boyunca gitmek, devam ettirmek, sürdürmek, taşımak, direnmek, azim etmek.

Birkaç örnek:

Yanardağ, Nisan’da 600 yıldır ilk defa etkinlik işaretleri göstermeye başladı,...

Elzem-olmayan tüm personel, yayılan tabanı boşaltmaya başlamalı.

Burada, kapalı yüklem sınıfının bir mensubu da bir tez olarak fiilli yada adlı tamamlayıcı seçebilir ve bu olayı, yukarıdaki yönlerden biriyle bağlantılı işlevle

(tanımlama) işaretleyebilir. İki olay arasındaki ilişkinin anlatılması için bölüm 3'e bakınız.

- **I_ACTION:**

I_ACTION Kasıtlı bir Eylemdir. I_ACTION, kendisinden I_ACTION ile ilişkili bir şeyler çıkarabileceğimiz bir eylemi yada durumu tarif eden (metin açıklığında olması gereken) bir olay tezini öne sürer. Örneğin, (1)'de I_ACTIONS tezleri olarak öne sürülen olayların, I_ACTION gerçekleşirken gerçekleşmesine gerek yoktur. (Aşağıda satırlar 5-9'da olduğu gibi) Açık uygulamalı yüklemeler de buraya dahil edilmiştir. I_ACTION sınıfının durumları kapsamadığını not ediniz (aşağıda I_STATES tarifine bakınız).

I_ACTION'ların koyu karakterli olduğu ve sundukları olayların altının çizili olduğu aşağıdaki liste, bu sınıfa dahil edilen olay türlerini temsilendir (kapsamlı değil):

1. **Teşebbüs etmek, denemek, mücadele etmek:**

Microsoft yada worldcom MCI gibi şirketler, İnternet erişimini monopolleştirmeye **çalışıyor**.

İsrail, yurtdışından daha fazla maske satınalmak için **mücadele ediyor**.

2. **Soruşturmak, sorgulama, bakmak, araştırmak:**

Afrika Birliği Örgütü, 500.000'den fazla Tutsi azınlığın Hutular tarafından örgütlenmiş soykırımını **soruşturacak**.

Yeni bir Essex İlçesi görev kuvveti, Perşembe günü 14 siyahi kadının katledilmesini **araştırmaya** başladı.

3. **Gecikme, erteleme, öteleme, önleme, geri bırakmak:**

İsrail, Birleşik Devletler'den Irak'a karşı askeri bir saldırıyı **geciktirmesini** isteyecek.

4. **Kaçınmak, önlemek, iptal etmek:**

Filistin polisi, Filistin Çalışanları Sendikası'nın planlı bir Irak-yanlısı gösterisini **önledi**.

5. **İstemek, emretmek, ikna etmek, talep etmek, yalvarmak, komuta etmek, zorlamak, yetkilendirmek:**

Irak askeri yetkilileri, Kuveyt'teki tüm Amerikalıların ve Britanyalıların bir otelde toplanmasını **emretti**.

Kendilerinden beraberlerinde önemli belgeleri de getirmeleri **istendi**.

6. Söz vermek, teklif etmek, güvence vermek, teklif etmek, kabul etmek, karar vermek:

Almanya, İsrail'e kimyasal ve biyolojik silahlara karşı 180,000 koruyucu takım borç vermeyi **kabul etti** ve İsviçre, İsrail'e 25,000 maske daha vermeyi **teklif etti**.

7. Yemin etmek, ahdetmek.

8. Adlandırmak, aday göstermek, tayin etmek, beyan etmek, açıklamak.

9. İddia etmek, savunmak, önermek.

• **I_STATE:**

I_STATE olayları önceki sınıfa benzerdir. Bu sınıf, (aşağıdaki örneklerde köşeli parantezle sınırları kaldırılan), bağlı cümleciklerle (1-2), adlandırmalarla (3) yada zamansız fiil kelmie gruplarıyla(VP) (4) sunulabilen alternatif yada muhtemel dünyalara atıfta bulunan durumları içerir:

1. Rusya halen [ABD'nin en azından BM genel sekreteri Kofi Annan Bağdat'ı ziyaret edinceye kadar beklemesi gerektiğini] hissediyor.
2. Bir kere söndürdüklerinde, [oturanların kendi evlerine dönmeye başlayacaklarını] ümit ediyorlar.
3. "Bir [saldırı] için hazırlıklı olmamamız için hiçbir gerekçe yok".
4. Kuruluşlar, [ajanları ve dolandırıcıları dinlemek üzere kodları kır]amayabileceklerinden korkuyorlar.

Yukarıda olduğu gibi, I_STATE'lerin aşağıdaki listesi, sadece temsil amaçlıdır, kapsayıcı değildir:

1. İnanmak, düşünmek, şüphe etmek, hayal etmek, şüphelenmek, hissetmek, anlaşılır olmak, emin olmak:

"[Sözlerinin dünyanın dikkatini Irak'taki saldırganlıktan geri çeviremeyeceğine] inanıyoruz".

Analistler [tedarikçilerin kendi başarılarına kurban olduklarından] da şüphe ediyor.

Rusya, halen [ABD'nin en azından BM genel sekreteri Kofi Annan Bağdat'ı ziyaret edinceye kadar beklemesi gerektiğini] hissediyor.

[Daha büyük bir patlamanın birkaç saat içerisinde gerçekleşmesi] anlaşılabilir.

[Daha büyük bir patlamanın gerçekleşeceğinden] emin olduğunu söyledi.

2. İstemek, aşık olmak, sevmek, arzu etmek, dayanamamak, arzulamak:

Bir A.B.D mürettebat şefi [bizimle oynamak] istemiyorlar” dedi.

[Değişiklik] için arzu uyandıracak dernekleşmeler başlatıyoruz.

İsrail’in [sona erdirmeyi] çok arzu ettiği, fakat ... bir işgal.

Ali kuzenini seviyor’da olduğu gibi sevmek I_STATE olarak kabul edilmez. Benzer bir biçimde, I_STATE başka bir olayın üstünde olmalıdır.

3. Ümit etmek, beklemek, bekleyiş içinde olmak, planlamak:

Bugün erken saatlerde bir televizyon mülakatında “[Ciddi bir patlama] bekliyoruz” dedi.

Bir kere söndürdüklerinde, [oturanların kendi evlerine dönmeye başlayacaklarını] ümit ediyorlar.

[Değişiklik için arzu uyandıracak oluşumlar başlatmayı] amaçlıyoruz.

4. Korkma, nefret etmek, çok korkmak, endişe etmek, korku içinde olmak:

Kuruluşlar [ajanları ve dolandırıcıları dinleme kodlarını kıramayacaklarından] korkuyorlar.

[Kalmaktan] korkuyorlardı.

5. Gereksinim duymak, gerek olmak, talep etmek

6. Hazır olmak, istekli olmak, hazırlıklı olmak

Genç endüstrinin hızlı büyümesi de [birçok yönüne polislik yapmak] isteyen düzenleyicileri kendine çekiyor.

“[Bir saldırı] için hazırlıklı olmamamızın hiçbir gerekçesi yok.

7. Yapabilmek, yapamamak

Kuruluşlar, [ajanlar ve dolandırıcıları dinlemek için kod kır]amayacaklarından korkuyorlar.

Tüm I_STATE’ler tutarlı olsunlar yada olmasınlar, işaretlenmeleri süresince anlatılacaktır (sonraki sınıfa bakınız).

- STATE:

Durumlar, bir şeyin gerçek olduğu yada bulunduğu hali tarif eder. Ancak, biz sadece anlatacağız:

1. Belgenin işaretlenmesi süreci içerisinde belirlenebilir biçimde değişmiş olan durumlar.

Örneğin, aşağıdaki ilk örnekte, Aeroflot Airbus ifadesinde, Airbus'ın Aeroflot tarafından yönetildiğini ve işletildiğini belirten ilişki, istenen anlamda STATE (DURUM) değildir. Bundan çok, belgenin olay hattı boyunca tutarlı olduğu için ortadan kaldırıyoruz ve böylece işaretlenmiyor. Diğer yandan, bir maddede temsil edilen / bildirilen olaylar esnasında değiştiği bilinen özellikler STATE olarak işaretlenecektir.

Bu ve sonraki örneklerde, işaretlenebilir durum koyu karakterdedir.

Aeroflot **uçacağındaki** 75 kişinin tamamı öldü.

İsrail, birkaç yüz bin gaz maskesi **eksikliğinin** keşfedilmesinden sonra ufak çocuklar için olanlar da dahil yurtdışından daha fazla maske satınalmak için mücadele ediyor.

Hafta sonu boyunca hiçbir **yaralanma** bildirilmedi.

2. Zamansal ifadeyle doğrudan ilişkili durumlar.

Bu kriter, TLINK vasıtasıyla TIMEX3 işaretlenebilirle bağlantılı tüm durumları içermektedir. Burada, durumun koyu karakterde olduğu ve kendisiyle bağlantılı zamansal ifadenin altının çizildiği iki örnek verilmiştir.

James Pustejovsky, birkaç yıllığına CTO idi.

1 2/2 yıl boyunca B.M.'in yönettiği mülteci kamplarında yaşadılar.

3. I_ACTION, I_STATE yada REPORTING olayıyla başlatılan durumlar.

Durumlar, koyu karakterdedir, başlatıcı olayın altı çizilmiştir.

Krizde aracılık etti.

Saddam Hüseyin, başka bir cephede **barış** arayışı içinde oldu.

Har-Shefi, polise Rabin'in **hain** olduğunu söyledi.

4. Yükleme, belge oluşturma süresine bağlı olanın geçerliliğini belirtir

Herhangi TIMEX3 ifadesiyle açıkça ilişkili olmamasına rağmen, aşağıdaki örneklerde İngilizcedeki yardımcı fiil “are” ile ortaya konulan durumlar, geçerliliklerinin vurgulandıkları zaman içerisindeki noktayla göreceli olmasıyla etiketlenecektir.

“A total of about 3,000 Americans, 3,000 Britons and more than 450 Japanese are in Iraq and Kuwait.

Overall, more than 2 million foreigners are in both countries.”

Toplam 3.000 Amerikalı, 3.000 Britanyalı ve 450’den fazla Japon, Irak’ta ve Kuveyt’te bulunuyor.

Genel olarak her iki ülkede 2 milyondan fazla yabancı bulunuyor.

Bu aynı zamanda finansal gazetelerde yer alanlar gibi birtakım niceliksel bildirimleri de içerecektir:

Gaz fiyatları, varil başına yirmi-bir dolardan bugünkü ondört dolara düştü.

Bu yine de sadece yüklem durumlarında geçerli olacaktır. Ayırıştırma durumları (Devlet Başkanı, CTO, v.s. gibi) işaretlenmeyecektir.

Mevcut sınıfın, STATE’in I_STATE olarak etiketlenmiş durumları içermeyeceğini not ediniz.

- OCCURRENCE:

Bu sınıf, dünyada olan yada gerçekleşen bir şeyleri tarif eden olayların diğer tüm türlerini içerir. Birtakım örnekler, resmetme olarak verilmektedir:

Savunma Bakanlığı, şimdiye kadar biyolojik ve kimyasal savaş araçlarına karşı koruyucu teçhizatlı 16 uçağın geldiğini söyledi.

Mordechai, yurtdışından gelen tüm gaz maskelerinin kısa zamanda geleceğini ve halka dağıtılacağını söyleyerek ilave dağıtım merkezlerinin önümüzdeki hafta kurulacağını ekledi.

Pazar öğleden sonra 3’ten kısa süre sonra iki hafif şiddetli patlama, daha büyük bir patlamanın sinyalini verdi.

Finansal dergilerden birtakım örnekler:

RMS, bir önceki yılın \$29,956’lık yada hisse başına iki sentlik zararıyla karşılaştırıldığında üçüncü çeyrekte \$158,666 yada hisse başına 10 sent zararının olduğunu söyledi.

Ralston, yeniden yapılanma maliyetlerinin N.C., Greenville’deki batarya tesisinin aşama aşama bitirilmesini içerdığını söyledi.

Tahıl bölümü, hacim artışlarında daha yüksek işletim karı gerçekleştirdi, fakat aynı zamanda promosyon için daha fazla para harcadı.

c. Kök Seçmeli Nitelik. Olayın başının kökü burada anlatılabilir.

6.2 <TIMEX3> Etiketi

TimeML'in zamansal ifadeleri anlatmak için kullandığı etiket kümelerinin ayrıntıları, ayrıntılarda STAG'daki TIMEX etiketinden ve TIDES'deki TIMEX2 etiketinden farklılık gösterdiği için burada, zamansal ifadeler için TIMEX3 etiket adını kullanıyoruz.

6.2.1 TIMEX3'lerin nasıl anlatılacağı

TIDES TIMEX2 anlatımıyla mümkün olduğu kadar uygunluk içerisinde olmak için, TIMEX3 etiketi, büyük ölçüde, TIMEX2 işaretlenebilir ifadelerine uygulanacaktır. Açıklama niyetinde olduğumuz özel ifadeler. (TIDES, 2002)

Ancak, TimeML aşağıdaki şekillerde TIDES'ten farklılık gösterecektir (aşağıda verilen örnekler, TIDES(02)'den uyarlanmıştır):

1. Kullanılacak olan TIMEX2 nitelikleri:

TimeML, sadece TIDES nitelikleri olan VAL ve MOD'u TIMEX3 öğeleri için uygun olarak alacaktır (TIDES(02), bölümler 3.2 ile 3.4 arası). Aşağıdaki özelliklerden hiçbiri, doğrudan ele alınmayacaktır: SET (KÜME), PERIODICITY (DÖNEMSELLİK), GRANULARITY (PARÇACIKLIK), NON_SPECIFIC (ÖZEL OLMAYAN) (TIDES(02), bölümler 3.5 ve 3.6'da açıklanmıştır). İlk üç nitelik üzerinden TIDES'in taşıdığı bilgiler, burada TANGO atölyesi esnasında TIMEX3'e eklenen özellikler kullanılarak ifade edilecektir.

2. Zaman ifadesinin etikete genişletilmesi (konu, TIDES(02), bölüm 4'te ele alınmıştır). TIMEX3 için etiket kapsamı, TIMEX2'de önerilene çok benzer olacaktır. Farklılığı açıklığa kavuşturmak için TIDES(02), bölüm 4 şemasını takip ediyoruz.

(a) Apozitifler (açıklayıcı ikinci kelime)

Ele alınması, tamamen TIDES (bölüm 4.2.1)'de olduğu gibi olacaktır.

(b) Kapsam İfadeleri:

Yine, TIDES, bölüm 4.2.2'de olanla aynı işlem.

(c) Birleştirilmiş İfadeler:

Bizim buradaki ele alışımlar, birleştirmenin (yada ayrıştırmanın) iki farklı TIMEX3 ifadesinin sınırını kaldırmasıyla TIDES'te teklif edilene benzer olacaktır. Ancak, en uzun zaman ifadesi dizinlerinin daha küçük TIMEX3 ifadelerine bölünebilmesinde TIDES'ten farklılık gösterecektir.

Örneğin, *Saddam şu andan altı ay yada bir yıl sonrasında bütün oyunu yeniden oynayabilir* cümlesinde *altı ay* zaman ifadesinin doğru olarak değerlendirilmesi için *şu andan (itibaren)*e atıfta bulunması gerekir. Bu nedenle, *şu andan, bir yıl* zaman ifadesinin parçası olarak ele alınamaz.

(d) Yerleşik İfadeler:

TIDES, yerleşik ifadelerin farklı işlemlerini belirlemektedir (TIDES(02), bölüm 4.2.4). Bazı durumlarda, tek bir etiket oluşturulur. Diğerlerinde, TIMEX2 etiketlerini yerleştirme ihtimali bulunuyor. Ancak bu, zamansal çıpa genel konsepti altında TimeML'de gerekli değildir. TimeML'deki yerleşik ifadelerin ele alınması, aşağıda gösterildiği gibidir:

- i. İfadelere TIDES(02), bölüm 4.2.4.1'de (*Ne Zaman Tek Etiket Oluşturulacağı*) olduğu gibi **ONE (BİR) tek TIMEX3 etiketi** tahsis edilecektir. TimeML anlatıcısı, bu gibi ifadelerin ayrıntıları için bu bölüme başvurmalıdır.

Örneğin, *gece yarısı saat oniki yada sabah onbir* TIMEX3 etiketinin kapsamı, zamansal ifadenin tam kapsamını içine almaktadır.

Diğer bazı örnekler şunlardır:

Cuma akşamı

18 Salı

3 Ocak 1984, saat oniki

Aralık'ın ikisi

1963'ün Ekimi

1964 yazı

- ii. Birbirinin içinde hiçbir etiket yuvası olmayan TWO (İKİ) TIMEX3 etiketleri, TIDES(02) bölüm 4.2.4.2'de atıfta bulunulan ifadelerin çoğuna tahsis edilecektir.

- A. Birbirine nispi olarak düzene konulan iki zamansal ifadenin dizinleri. Genellikle *den*, *önce*, *sonra*, *ardından*, *öncesinde*, v.s. gibi zamansal önermelerin ve bağlaçların kullanımını içerir. Bu gibi kelimeler, sinyaller olarak anlatılacaktır ve iki zamansal ifadenin nispi düzenini ifade etmek için bağlantılar kullanılacaktır – sırasıyla bölümler 2.3 ve 3’e bakınız. Buna ilaveten, bu ifade türleri çıpalı süreler olarak kabul edilebilir ve TimeML’in en yeni versiyonunda bu şekilde anlatılabilir – alt bölüm 2.2.3’e bakınız

Önümüzdeki Perşembe’den itibaren iki haftalığına tatile çıkıyorum.

<TIMEX3 tid="t1">

iki hafta

</TIMEX3

den

<TIMEX3 tid="t2">

önümüzdeki Perşembe

</TIMEX3>

Ali, dünden 2 gün önce ayrıldı..

<TIMEX3 tid="t1">

2 gün

</TIMEX3>

önce

<TIMEX3 tid="t2">

dün.

</TIMEX3>

Üç yıl önce bugün, Los Angeles’i ciddi bir deprem sarstı..

<TIMEX3 tid="t1">

üç yıl

</TIMEX3>

önce

<TIMEX3 tid="t2">

bugün

</TIMEX3>

Anlatıcı, yukarıda bölüm 4'teki örneklerin tam anlatılmasını kontrol etmeye güçlü biçimde teşvik edilmektedir.

- B. TIDES(02)'nin bu bölümüne iyelik oluşumları eklenmişken, bunlar çıpalı süreler değildir. Böylece, hem iyelik cümlesi hem de değiştirdiği cümle zamansal ifadelerse, bunlar ONE (BİR) TIMEX3 etiketiyle anlatılabilir.

Bu yılki zirve, çok nadir biçimde sıcaktı.

<TIMEX3 tid="t1">

Bu yılki zirve

</TIMEX3>

- iii. TLINK etiketiyle ilişkilendirilen TWO (İKİ) TIMEX3 etiketi, yukarıda açıklanandan başka durumları ele almak üzere uygulanacaktır. Aşağıdaki örneklerde, her bir TIMEX3 etiketinin kapsamı, köşeli parantezle işaret edilmiştir.

[1998]'de bazı [Perşembe günleri] bir İngiliz öğrencisine ders verdim.

Konser, [Cuma] günü [öğleden sonra saat 8.00]'da.

Konser [Cuma günü] [öğleden sonra saat 8.00]'da.

Üçüncü örnekte (Cuma öğleden sonra saat 8.00'de) ve yukarıda 2(d)i'deki diğer bazılarında olduğu gibi, (Cuma akşamı gibi) tek bir TIMEX3 etiketiyle ele alınan zamansal ifadeler arasındaki farklılık, önceki örnekte **de** sinyalinin bulunmasıdır. Yukarıda 2(d)ii'de verilen durumlar haricinde, SIGNAL ile işaretlenebilir öğeler, TIMEX3 etiketlerinin sınır kaldırıcıları olarak kullanılacaktır.

(e) Son değiştiricileri içeren zamansal ifadeler

TimeML aynı zamanda, ilgili bir olayı ortaya koyan son değiştiricileri (önerme cümlecikleri ve bağımlı cümlecikler) içeren zamansal ifadelerle ilgili olarak TIDES(02)'den farklılık gösterir. Bu nedenle, TIDES(02)'nin aksine, aşağıdaki örneklerde parantez içerisindeki son değiştiriciler, zamansal ifadenin parçası değildir. Bu örneklerin, zaman ifadesiyle olay arasındaki bağlantıları içeren tam

bir anlatması bölüm 4’te verilmiştir.

geri gelmesinden sonra beş gün

<TIMEX3 tid="t1">

beş gün

</TIMEX3>

sonrasında

<EVENT eid="e1" class="OCCURRENCE">

geldi

</EVENT>

geri

neredeyse kırk yıllık tecrübe

<TIMEX3 tid="t1">

neredeyse kırk yıl

</TIMEX3>

lık

<EVENT eid="e2" class="STATE">

tecrübe

</EVENT>

aylarca süren tekrar eden düşmanlık

<TIMEX3 tid="t1">

aylarca süren

</TIMEX3>

tekrar eden

<EVENT eid="e2" class="STATE">

düşmanlık

</EVENT>

İnsanlarımızın geleceği

<TIMEX3 tid="t1">

gelecek

</TIMEX3>

insanlarımızın.

Buradaki durumlarla, değiştirici sonrasının olay değil zamansal bir ifade ortaya koyduğu yukarıdakiler arasındaki farkı fark etmeye değer.

Zarfa ait son değiştiriciler, TIMEX3 işaretlenebilir ifadesinin parçası olarak ele alınacaktır:

en iyi ikinci devre

<TIMEX3 tid="t1">

en iyi ikinci devre

</TIMEX3>

üç yıl önce

<TIMEX3 tid="t1">

üç yıl önce

</TIMEX3>

3. **Boş İçerik TIMEX3'leri.** TimeML'in yeni özelliklerinden biri, metin harcamayan fakat bunun yerine metinde yerleşik olan zamansal bir ifadeyi temsil eden TIMEX3 etiketlerine imkan tanır. Bunlar en çok, metin önerdiği bir zamanda fakat içinde açık olmayan bir olay gerçekleştiğinde olur. Çıpalı sürelerin bulunması, sıklıkla boş içerik TIMEX3 gereksinimini harekete geçirir. Eğer belge garanti ediyorsa anlatıcı, bir dizi zamansal ifadede ima edilen bir süreyi temsil etmek yada bir sürenin başlangıç yada sona eriş noktasını açık hale getirmek üzere üzere boş bir TIMEX3 oluşturabilir.

6.2.2 TIMEX3 etiketi için BNF

attributes (nitelikler) ::= tid type
 [functionInDocument][beginPoint][endPoint][quant][freq] (Belgedeki işlev – başlangıç noktası – sona erme noktası – miktar- sıklık)
 [temporalFunction] (value | valueFromFunction) [mod][anchorTimeID] (geçici İşlev – değer, İşlevden değer – biçim – çıpa süresi zaman kimliği=
 tid ::= ID
 {tid ::= TimeID
 TimeID ::= t<integer>}
 type ::= 'DATE' | 'TIME' | 'DURATION' | 'SET' (tip: tarih – saat – süre – küme)
 beginPoint ::= IDREF (başlangıç noktası)
 {beginPoint ::= TimeID} (başlangıç noktası)
 endPoint ::= IDREF (sona erme noktası)
 {endPoint ::= TimeID} (sona erme noktası)
 quant ::= CDATA (miktar)
 freq ::= CDATA (sıklık)
 functionInDocument ::= 'CREATION_TIME' | 'EXPIRATION_TIME' |
 'MODIFICATION_TIME' | 'PUBLICATION_TIME' |
 'RELEASE_TIME' | 'RECEPTION_TIME' | 'NONE' (Belgedeki işlev – oluşturma zamanı – sona erme zamanı * değiştirme zamanı – yayınlama zamanı * açıklama zamanı – alma zamanı – hiçbirisi)
 {default, if absent, is 'NONE'} (kurulu ayar, eğer yoksa, hiçbirisi)
 temporalFunction ::= 'true' | 'false' {default, if absent, is 'false'} (zamansal İşlev – doğru, yanlış, kurulu ayar, eğer yoksa, yanlış)
 {temporalFunction ::= boolean} (zamansal işlev)
 value ::= CDATA (değer)
 {value ::= duration | dateTime | time | date | gYearMonth | (değer, süre, tarih, saat, tarih, g Yılı Ay)
 gYear | gMonthDay | gDay | gMonth} (g Yılı, g Günü Ay, g Günü, g Ayı)
 valueFromFunction ::= IDREF (işlevden değer)
 {valueFromFunction ::= TemporalFunctionID (İşlevden değer) (Geçici İşlev Kimliği)
 TemporalFunctionID ::= tf<integer>} (Geçici İşlev Kimliği)

$\text{mod} ::= \text{'BEFORE'} \mid \text{'AFTER'} \mid \text{'ON_OR_BEFORE'} \mid \text{'ON_OR_AFTER'} \mid$
 $\text{'LESS_THAN'} \mid \text{'MORE_THAN'} \mid \text{'EQUAL_OR_LESS'} \mid$
 $\text{'EQUAL_OR_MORE'} \mid \text{'START'} \mid \text{'MID'} \mid \text{'END'} \mid \text{'APPROX'}$ (biçim: önce, sonra,
 gününde yada öncesinde, gününde yada sonrasında, daha az, daha fazla, eşit yada az,
 eşit yada fazla, başlat, orta, son, yaklaşık)
 $\text{anchorTimeID} ::= \text{IDREF}$ (çıpa süresi kimliği)
 $\{\text{anchorTimeID} ::= \text{TimeID}\}$ (çıpa süresi kimliği)

6.2.3 TIMEX3 için nitelikler

a. Timex Kimlik numarası (tid) Seçmeli-olmayan nitelik. Her bir TIMEX3 ifadesi, özel bir kimlik numarasıyla belirlenmelidir. Bu, anlatma aracıyla otomatik olarak tahsis edilir.

b. Tür Seçmeli-olmayan nitelik. Her bir TIMEX3, aşağıdaki türlerden birine tahsis edilir: DATE (tarih), TIME (saat), DURATION (süre) yada SET (küme). Nitelik değerinin biçimi, TIMEX3'ün türüyle belirlenir. Örneğin, bir DURATION (süre)nin 'P' harfiyle başlayan bir değeri vardır, zira süreler bir zaman süresini temsil eder. Bu, aşağıda değer kısmında değerlendirilecektir. Buna ilaveten, birtakım seçmeli nitelikler, belli türde zamansal ifadelerle birlikte özellikle kullanılır.

Aşağıdaki örnekler, her bir muhtemel tür değerini sergilemektedir. Bunlarda, TIMEX3 işaretlenebilir ifadesi, koyu karakterdedir:

- TARİH: İfade, bir takvim süresini ifade etmektedir.

Ali, 1 Ekim 1999, Cuma günü ayrıldı.

Aralık'ın ikinci günü

dün

1963 Ekim'inde

1964 yazında

18 Salı'da

Kasım 1943'te

bu yıl yazın

önümüzdeki Perşembe'den iki hafta sonra

geçen hafta

DATE (tarih) de, bir aralık içeren iki TIMEX3 işaretlenebilir ifadesinin her birinin tür niteliğinin değeri olabilir. Örn.,

Ali, Pazartesi ile Çarşamba arasında ayrıldı.

Ali arasında ayrıldı

<TIMEX3 tid="t1" type="DATE">

Pazartesi

</TIMEX3>

ile

<TIMEX3 tid="t2" type="DATE">

Çarşamba

</TIMEX3>

Aynı örneğin tam anlatılmış bir versiyonu için bölüm 4'e bakınız.

- TIME (zaman): İfade, (aşağıdaki son iki örnekte olduğu gibi) belirsiz bir yolla da olsa günün bir zamanına atıfta bulunmaktadır:

Ali, üçe on dakika kala ayrıldı

sekize beş dakika kala

onikiyi yirmi geçe

öğleyin oniki buçukta

sabah saat onbirde

1 Ekim 1999, Cuma günü sabah saat 9'da

31 Ocak sabahı

dün gece geç vakitte

dün gece

Daha önce olduğu gibi, TIME aynı zamanda birlikte zamansal bir aralığa (örn., Bay Smith sabah saat 8 ile 10 arası ayrıldı) atıfta bulunan iki TIMEX3 işaretlenebilir ifadesinin her biri için tür değeri de olabilir.

- DURATION (SÜRE): İfade, bir süreyi tarif eder. Bu değer, sadece aşağıdakiler gibi açık sürelerle tahsis edilir:

Ali, Boston’da 2 ay kaldı

48 saat

üç hafta

dün tüm gece

Temmuz’da 20 gün

geçen Pazartesi 3 saat

Kural olarak, eğer zamansal ifadede herhangi belirli takvim bilgisi tedarik edilmişse TIMEX3’ün türü ya DATE (tarih) yada TIME (zaman) olmalıdır. Eğer metin bir olayın o yıl boyunca devam ettiğini ortaya koyarsa birtakım anlatıcılar, “1985” gibi bir şeyi DURATION (süre) yapmaya yöneltilebilir. Ancak, bir DURATION (süre)nin nitelik değerinde özel bir biçiminin olması gerektiğinde böyle bir zamansal ifade, burada tarif edilen gibi DATE (tarih) olmalıdır.

- SET (küme): İfade, bir dizi saati tarif etmektedir. Bu değer, TIDES(02) bölüm 3.5’te olan gibi ifadelere tahsis edilir. Örneğin:

Ali haftada iki defa yüzer

her 2 günde bir

- Değer. Değer niteliği (TIMEX2’de VAL’e eşittir), tamamen TIDES(02) bölümler 3.2 ve 3.3’te belirtildiği şekilde anlatılacaktır. Ancak, bu bölümlerin aynı zamanda TimeML’de kullanılmayan iki ek niteliğin, ANCHOR_VAL ve ANCHOR_DIR kullanımını da içerdiğini not ediniz.

Anlatım aracının yardımıyla anlatıcı, hem zaman ifadesinde hem de metin bağlamından mevcut olan kadar çok bilgi sunacaktır. Önceki bölümlerden olan aşağıdaki örnekler, günün saatleri, tarihler, süreler ve kümeler için değer niteliğinin kullanımını kısmen sergilemektedir:

- Günün saatleri ve tarihler:

```
<TIMEX3 tid="t1" type="TIME" value="T24:00">
```

gece yarısı oniki

```
</TIMEX3>
```

Metnin, zamanın çıpalandığı belirli bir tarihe birtakım atıflarda bulunması halinde (örn., bugünün 12 Temmuz 2002 Cuma olduğu varsayılırsa *Geçen Cuma güünkü parti, gece yarısı onikiye kadar başlamadı*), değer niteliği tarihi içermelidir ve şöyle olmalıdır: value="2002-07-05T24:00"

```
<TIMEX3 tid="t2" type="TIME" value="T11">
```

sabah saat onbir

```
</TIMEX3>
```

Aynı yorum, yukarıda olduğu gibi burada da geçerlidir.

```
<TIMEX3 tid="t3" type="TIME" value="2001-01-12TEV">
```

Cuma akşamı

```
</TIMEX3>
```

Yukarıdaki örnekteki değer, belge oluşturma zamanının 8 Ocak 2001 Salı olduğunu bildiğimizi varsaymaktadır.

```
<TIMEX3 tid="t4" type="TIME" value="1984-01-03T12:00">
```

3 Ocak 1984 saat oniki

</TIMEX3>

Öğle vaktinin kastedildiğini varsayıyoruz.

<TIMEX3 tid="t5" type="DATE" value="XXXX-12-02">

Aralık'ın ikinci günü

</TIMEX3>

Metnin mevcut zamansal ifadeyi çıpaladığı yılı bilmedikçe, 'değer' niteliği, yer tutucular vasıtasıyla yılı az belirtilmiş şekilde bırakacaktır: XXXX

<TIMEX3 tid="t6" type="DATE" value="1964-SU">

1964 yazı

</TIMEX3>

arasında ayrıldı

<TIMEX3 tid="t1" type="TIME" value="T18:00">

öğleden sonra saat 6:00

</TIMEX3>

ile

<TIMEX3 tid="t1" type="TIME" value="T20:00">

8:00

</TIMEX3>

Önceki ve sonraki örneklerde aralığın süresi, olayı hem başlama saati yada tarihi hem de sona erme saati yada tarihiyle bağlayan bir çift TLINK ile taşınacaktır. Bazı durumlarda, aralıkla ifade edilen süreyi açıkça temsil etmek üzere boş bir etiket de oluşturulabilir.

‘dan itibaren izinde olacağız

<TIMEX3 tid="t1" type="DATE" value="XXXX-07-01">

1 Temmuz

</TIMEX3>

‘a kadar

<TIMEX3 tid="t1" type="DATE" value="XXXX-08-15">

15 Ağustos

</TIMEX3>

- Süreler:

<TIMEX3 tid="t1" type="DURATION" value="P4M">

4 ay

</TIMEX3>

için

<TIMEX3 tid="t1" type="DURATION" value="PT20M">

20 dakika

</TIMEX3>

<TIMEX3 tid="t2" type="DATE" value="XXXX-07-08">

geçen Pazartesi günü

</TIMEX3>

boyunca

<TIMEX3 tid="t1" type="DURATION" value="P2D">

iki tüm gün

</TIMEX3>

‘da

<TIMEX3 tid="t2" type="DATE" value="1999-SU">

1999 yazı

</TIMEX3>

- Kümeler:

Kümeleri tam olarak anlatmak için, TIMEX3 aynı zamanda eğer her ikisi olmazsa quant yada freq niteliklerini de içermelidir. Aşağıdaki örnekler, değer niteliğiyle ilgili olduğu kadarıyla TIMEX3 kümesinin anlatımını başlatır:

```
<TIMEX3 tid="t1" type="SET" value="P1W">
```

haftada iki defa

```
</TIMEX3>
```

```
<TIMEX3 tid="t1" type="SET" value="P2D">
```

her 2 günde bir

```
</TIMEX3>
```

```
<TIMEX3 tid="t1" type="SET" value="P1W">
```

her hafta 3 gün

```
</TIMEX3>
```

```
<TIMEX3 tid="t1" type="SET" value="XXXX-10">
```

her Ekim ayı

```
</TIMEX>
```

Son örnekte, değer in daha çok DURATION (süre)den daha çok DATE (tarih) gibi görüldüğüne dikkat ediniz. Bu böyledir, dolayısıyla zamansal ifadede mevcut bulunan takvim bilgilerini alabiliriz. Birtakım anlatıcılar, ne zaman bir DATE (tarih)-benzeri biçimin tercih edileceğini kafa karıştırıcı bulabilir. Genelde, eğer belirtilen hiçbir takvim tarihi (örneğin, “Ekim” yada “Salı”) yoksa SET (küme) değeri DURATION (süre)ninki gibi olacaktır.

d. Mod Doğrudan TIMEX2 MOD niteliğinden alınan seçmeli nitelik. Değeri, TIDES(02), bölüm 3.4’te belirtildiği gibidir.

e. Zamansal İşlev zamansal ifadenin değerinin zamansal işlevin değerlendirilmesi yoluyla belirlenmesi gerektiğini ifade eden çiftli nitelik. Zamansal işlevler, T3PO (TIMEX3 İnceleme Nesneleri) modülü dahilinde süreç sonrası olarak uygulanacaktır.

Bu niteliğin değeri, tarih değerinde daha yüksek düzenli (sol el) konumları doldurmak üzere gerekli tüm bilgileri içermeyen durumlarda pozitif olacaktır. Bazı örnekler şunlardır:

sabah saat onbir

31 Ocak

dün gece geç vakitte

geçen hafta

Metin gerekli bilgileri sağladığı için değer niteliği tamamıyla doldurulmuş olsa dahi temporalFunction (zamansal İşlev) niteliğinin, T3PO'daki sürekli zamansal işlev, insan anlatıcının tahsis ettiği değeri yeniden üretip üretmeyeceğini test edilebilecek şekilde yine de pozitif bir değer alması gerektiğini not ediniz.

Diğer yandan, değer yüksek sıralı konumunun doldurulduğu durumlarda, temporalFunction (zamansal işlev) negatif değerde tahsis edilmelidir. Bu gibi durumlar şunları içerir:

3 Ocak 1984, saat oniki

1964 yazı

1 Ekim 1999, Cuma

1 Ekim 1999, sabah saat 9

31 Ocak 1999 sabahı

Süreler, eksik belirtme durumunda temporalFunction (zamansal İşlev) niteliğinin değerini doğru olarak alacaktır. Bu nitelik için doğru olarak belirtilecek ifade örnekleri, TIMEX3 etiketinin kapsamının koyu karakterde olduğu şekilde aşağıdaki gibidir:

geçen aylarda

<TIMEX3 tid="t1" type="DURATION" value="PXM"
temporalFunction="true"> (geçici İşlev = doğru)

önümüzdeki saatlerde

<TIMEX3 tid="t1" type="DURATION" value="PXH"

temporalFunction="true"> (geçici İşlev = doğru)

Diğer yandan, temporalFunction (zamansal İşlev) niteliği değeri olarak tahsis edilecek olan süre ifadeleri şunlardır:

iki ay süreyle

48 saat

Kurulu ayarlarla, hemen yukarıdakiler gibi belirli ifadelerin esasta nadir görünmesiyle temporalFunction (zamansal İşlev)de sunulan değer pozitif olacaktır.

f. anchorTimeID (çıpa Saat kimliği) Seçmeli nitelik. TIMEX3 işaretlenebilirinin geçici olarak çıpalandığı zaman ifadesinin kimliğini sunar. O zaman değeri, her zaman timeID'dir. *Salı günü ayrıldı* cümlesinde *ayrıldı* olayıyla *Salı* zamanı arasındaki ilişki, TLINK vasıtasıyla anlatılacaktır (bölüm 3'e bakınız).

Anlatılabilir olan zamansal çıpalar, TIMEX3 etiketinin kapsamının dışında olanlardır. Dolayısıyla, *önümüzdeki Salı'dan itibaren iki hafta* dizininde, *önümüzdeki Salı*, *iki haftanın* zamansal çıpası olarak anlatılmayacaktır. Bunun yerine, bütün TIMEX3-işaretli ifadenin zamansal çıpası, *önümüzdeki Salı* tarihinin değerlendirilebileceğiyle nispi olacaktır. Bunun, timeID t0 ile atıfta bulunulan tarihe karşılık geldiğini varsayınız:

Mary önümüzdeki Salı'dan itibaren iki hafta içerisinde gelecek.

<TIMEX3 tid="t1" type="TIME" value="2002-08-06"

temporalFunction="true" anchorTimeID="t0">

önümüzdeki Salı'dan itibaren iki hafta

</TIMEX3>

Başka bir örnek:

Mary, Salı günü ayrıldı ve Ali sonraki gün vardı.

Mary ayrıldı
gününde

<TIMEX3 tid="t1" type="DATE"
value="1998-WXX-4" temporalFunction="true" (değer, zamansal İşlev, doğru)
anchorTimeID="t0"> (çıpa zamanı)

Perşembe

</TIMEX3>

ve Ali vardı

<TIMEX3 tid="t2" type="DATE"
value="1998-WXX-5" temporalFunction="true" (değer, zamansal İşlev, doğru)
anchorTimeID="t1"> (çıpa süresi)

gün

</TIMEX3>

sonraki

Önceki durumda, her iki TIMEX3 ifadesi de zamansal işlevin uygulanmasını gerektirir: (sonraki) günün Salı ile nispi olarak ve Perşembe'nin mevcut örnekte açık olmayan başka

bir zaman referansına nispi olarak değerlendirilmesi gerekir.

TERQAS Atölyesi, 15 Temmuz Pazartesi günü yeniden başlayacaktır. Oturum, sabah saat 9'da başlayacaktır.

TERQAS Atölyesi yeniden başlayacak.

<TIMEX3 tid="t1" type="DATE" value="2002-07-15"> (tür, tarih, değer)

15 Temmuz, Pazartesi

</TIMEX3>

Oturum –‘da başlayacak

<TIMEX3 tid="t2" type="TIME" value="T9:00"

temporalFunction="true" anchorTimeID="t1"> (zamansal İşlev, doğru, çıpa Zamanı)

sabah saat 9:00

</TIMEX3>

anchorTimeID niteliğinin bulunmasının temporalAnchor='true' ile görüldüğünü not ediniz. Ancak, durum her zaman böyle değildir.

g. valueFromFunction (değerden İşlevi) Bu nitelik, manüel anlatma amacıyla ilgili değildir, fakat sadece T3PO süreç sonrası içindir. İnsani anlatıcı bunu göz ardı etmelidir.

h. functionInDocument (iç işlev Belgesi) Bu nitelik, bir TIMEX3'ün belgedeki diğer zamansal ifadeler için zamansal çıpa sağlamadaki işlevini belirtir. Yeni bir kaynağın ömründeki başlıca kilometre taşlarını işaret edildiği birkaç defa oldu:

- Hikayenin oluşturulduğu zaman
- Hikayenin değiştirildiği zaman
- Hikayenin yayınlandığı zaman
- Açıklanabileceği zaman (eğer derhal değilse)
- Bir müşteri tarafından alındığı zaman
- Hikayenin sona erdiği zaman (eğer varsa)

Bu niteliğin muhtemel değerleri o zaman şunlardır: 'CREATION TIME', (oluşturma zamanı) 'MODIFICATION TIME' (değiştirme zamanı), 'PUBLICATION TIME' (yayınlama zamanı), 'RELEASE TIME' (açıklama zamanı), 'RECEPTION TIME' (alım zamanı), 'EXPIRATION TIME' (sona erme zamanı), 'NONE' (hiçbiri)

Eğer bu nitelik açıkça tedarik edilmemişse, kurulu değer "NONE" (hiçbiri)dir.

```
<TIMEX3 tid="t1" type="DATE" functionInDocument="CREATION_TIME"
value="1996-0327">
```

03-27-96

```
</TIMEX3>
```

Aşağıdaki nitelikler, TimeML'deki sürelerin ve kümelerin anlatılmasını güçlendirmek için kullanılır. Her biri seçmelidir ve anlatıcının takdirinde kullanılabilir. Ancak, beginPoint (başlama noktası) ve endPoint (sona erme noktası) nın sadece ifade türünün süre olması halinde kullanılması gerektiğini ve quant ve freq'in sadece ifade ayarlandığında kullanılması gerektiğini not etmek önemlidir.

i. beginPoint ve endPoint Bu nitelikler, başka bir zaman ifadesi tarafından bir süre çıpalandığında kullanılır. Eğer bu noktalardan sadece biri metinde verilmişse anlatıcı, eksik noktayı temsil etmek üzere boş bir TIMEX3 oluşturabilir. Bazı yollarda, beginPoint ve endPoint nitelikleri, anchorTimeID ile benzerdir. Bu niteliklerde saklanan değerler, eksik noktayı hesaplamak ve kendisi için bir etiket oluşturmak üzere zamansal işlevler tarafından kullanılabilir. Aşağıdaki örnek, bunun nasıl olabileceğini göstermektedir:

Ali ders vermeye başlar

```
<TIMEX3 tid="t1" type="DURATION" value="P1W" beginPoint="t2"
endPoint="t3">
```

bir hafta

```
</TIMEX3>
```

'den itibaren

```
<TIMEX3 tid="t2" type="DATE" value="XXXX-9-15">
```

15 Eylül

```
</TIMEX3>
```

```
<TIMEX3 tid="t3" type="DATE" value="XXXX-9-22" temporalFunction="TRUE"
anchorTimeID="t1" />
```

Boş TIMEX3 tarafından oluşturulan yeni tid, ders verme olayını doğrudan gerçekleştiği zamana bağlamak için kullanılabilir. Bu konuda daha fazla bilgi için, bölüm 3'e bakınız.

j. quant ve freq Bu nitelikler, zamansal bir ifade, SET türüdeyken kullanılır. Belirleyici (Quant), ifadeyi niteleyen metinden genelde birebirdir. freq, zamansal ifadenin düzenli olarak yeniden oluştuğu sıklığı temsil eden bir tam sayı değeri ve zaman parçacığı içerir. Bu nitelikler, sadece değerlerinin zamansal ifadeyle (yada zamansal çıpayla) temin edilmesi halinde kullanılır. Her ne kadar bu niteliklerin değerlerinin çıkarılabilmesi elverişli görünse de, elle anlatım amacıyla böyle olmayacaktır. Her ne kadar, belirli bir belirleyici olmasa da, kümelerin evrensel olarak nicelikli olduğu hayal edilir. Aşağıdaki örnekler, bu bölümde daha önce sayılan kümelerin anlatımlarını tamamlamaktadır:

```
<TIMEX3 tid="t1" type="SET" value="P1W" freq="2X">
```

haftada iki defa

```
</TIMEX3>
```

```
<TIMEX3 tid="t1" type="SET" value="P2D" quant="EVERY">
```

her 2 günde bir

```
</TIMEX3>
```

```
<TIMEX3 tid="t1" type="SET" value="P1W" quant="EACH" freq="3d">
```

her hafta 3 gün

```
</TIMEX3>
```

```
<TIMEX3 tid="t1" type="SET" value="XXXX-10" quant="EVERY">
```

her Ekim ayı

```
</TIMEX>
```

6.3 <SIGNAL> Etiketi

Sinyal, iki kiři arasındaki iliřkiyi (timex ve olay, timex ve timex, yada olayla olay) açık eden metinsel bir ögedir. Sinyaller genellikle řöyledir:

- Zamansal önermeler: üstünde, içinde, -de, -den, -ye, önce, sonra, esnasında, v.s.
- Zamansal bağlaçlar: önce, sonra, esnasında, zamanında, v.s.
- Değişmeye işaret eden edatlar: -e
- Özel karakterler: aralıkları (4-6 Eylül, Nisan 1999 / Temmuz 1999, v.s.) gösteren zamansal ifadelerde “-“, ve “/”.

6.3.1 SIGNAL’lerin nasıl anlatılacağı

Zamansal Edatlar, Bağlaçlar ve Değişiriciler Bu işlevsel kelimeler, aşağıdaki basit örneklerde gösterildiğı şekilde etiketlenecektir:

Ali </SIGNAL> Pazartesi günü ders verdi <SIGNAL sid="s1">

Uçak dağa çarptığında </SIGNAL> tüm yolcular öldü <SIGNAL sid="s1">.

İki ayrı sinyal yan yana görüldüğünde, eğer yukarıda sayıldığı gibi farklı sinyal sınıflarına aitse bunlar ayrı olarak anlatılabilir. Ancak, bazı durumlar, tek SIGNAL olarak anlatılmalarını gerektirir. Örneğin:

<SIGNAL sid="s1"> Soykırımdan önce, esnasında ve sonrasında </SIGNAL> ABD’nin rolünü araştıracaklar.

Bu durumda, soykırımla rol olayları arasındaki TLINK’in IS_INCLUDED ilişkisini uygun biçimde telafi etmek için üç zamansal edatın tek bir SIGNAL içine çökertilmesi gerekir (bölüm 3’e bakınız).

6.3.2 SIGNAL etiketi için BNF

attributes ::= sid

sid ::= s<integer>

6.3.3 SIGNAL için nitelikler

SIGNAL'in tek bir, seçmeli-olmayan niteliği bulunmaktadır: sid, sinyalin özel kimliği. SIGNAL'in her işaretlendiğinde anlatım aracı tarafından tahsis edilir.

6.4 <MAKEINSTANCE> Etiketi

Olay işaretleriyle olay örnekleri yada gerçekleştirimleri arasında ayrım yapıyoruz – MAKEINSTANCE, bir olayın fiili gerçekleşmesini oluşturur. Hareket noktası, bir fiilin iki olayı temsil ettiği *Ali Pazartesi ve Salı günü ders verdi* gibi örneklerdir. Bu gibi durumları anlatabilmek için, iki farklı olayı temsil eden iki tane *ders verdi* olayı oluşturmak gereklidir. Belirli bir olay örneğinin zamanı, yönü, konuşma bölümü, kutupsallığı ve kipliliği de bu etikette temsil edilir. Böylece, *Ali bugün ders verdi fakat yarın vermeyebilir* gibi bir cümle yeterince temsil edilebilir. MAKEINSTANCE'lar (olay remzini işaretleyen) olay anlatmasına ek olarak oluşturulur.

Çoğu durumda, sadece bir MAKEINSTANCE'a gereksinim vardır ve anlatım aracıyla otomatik olarak oluşturulacaktır, bu nedenle anlatıcının ikinci (ve eğer gerekirse sonraki) örnek yada olay gerçekleşmesi için sadece additional MAKEINSTANCE yerleştirmesi gerekir.¹ Aynı zamanda metnin harekete geçirdiği kadarıyla anlatıcının MAKEINSTANCE'lar yada kardinal değer taşıyan MAKEINSTANCE oluşturabileceği durumlar da vardır. Örnekler, şunlardır:

¹ Anlatım aracının bunu yapamadığı durumlarda anlatıcı için kutupsallık ve kiplilik niteliklerine değerler atamak gerekli olabilir. Bu değerler, büyük ölçüde TimeML özelliğinin önceki versiyonlarında otomatik olarak oluşturulan SIGNAL'lerle ve SLINK'lerle temsil edilir.

1. *Ali, Pazartesi günü iki defa ders verdi.*
2. *Ali, geçen yıl 150 defa ders verdi.*

İlk örnekte anlatıcı, iki kardinallikle ya iki MAKEINSTANCE yada tek bir MAKEINSTANCE oluşturabilir. İkinci örnekte, tek pratik çözüm 150 kardinallikle bir MAKEINSTANCE oluşturmaktır.

6.5 MAKEINSTANCE'ların Nasıl Anlatılacağı

MAKEINSTANCE'ların nasıl anlatılacağını göstermek için örnekler kullanacağız. Bu örneklerde, olayların, zamanların ve sinyallerin ayrıntılı bir anlatımını vermeyeceğiz, dolayısıyla bunların nasıl anlatılacağına dair talimatlar için lütfen uygun bölümlere başvurunuz. Aynı zamanda bu kişilikler için örneklerle ilgili işaretlemeyi de göstermemiz gerekir.

Her bir örnek yada olay gerçekleşmesi için bir MAKEINSTANCE oluşturulmalıdır – zira çoğu, metinden harekete geçmektedir. Alternatif olarak, kardinallik niteliği kullanılabilir, örn. *her* ve büyük numaralar için. Kardinallik yeterince küçük olduğunda (örn. iki defa) anlatıcının uygun kardinallikle MAKEINSTANCE oluşturma yada uygun olduğu şekilde olabildiğinde çok MAKEINSTANCE oluşturma seçeneği bulunmaktadır. Olaylara metinde daha sonra atıfta bulunulmadıkça sadece tek bir MAKEINSTANCE kullanılmasını tavsiye ediyoruz. Anlatım aracı tarafından her bir olay için tek bir MAKEINSTANCE'ın otomatik olarak oluşturulacağını not ediniz, bu nedenle metinde bir olay örneğinden fazlasına atıfta bulunulduğunda sadece anlatıcının ilave MAKEINSTANCE oluşturmaları gerekir. Bu, gerektiği kadar çok sayıda olayı anlatmakla yada 'darbe vurmakla' yapılır. Her bir 'darbe' otomatik olarak yeni bir örnek kimliği oluşturacaktır.

Aşağıdaki örneklerin her birinde, *ders verdi* olayı aşağıdaki gibi işaretlenir.

```
<EVENT eid="e1"> taught </EVENT>
```

1. Ali, Pazartesi günü ders verdi.

Bu, otomatik olarak oluşturulan MAKEINSTANCE'e yol açar:

```
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PAST" aspect="NONE" pos="VERB"/>
```

2. *Ali, Pazartesi ve Salı günü ders verdi*

Bu örnekte ders verme olayının iki farklı zaman değeri olduğu için, iki MAKEINSTANCE gereklidir. İlki, otomatik olarak oluşturulacaktır, fakat ikincisi anlatıcı tarafından oluşturulmalıdır:

```
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PAST" aspect="NONE"
pos="VERB"/>
```

```
<MAKEINSTANCE eiid="ei2" eventID="e1" tense="PAST" aspect="NONE"
pos="VERB"/>
```

3. Ali Pazartesi günü iki defa ders verdi

Metinde sadece tek bir zaman ifadesi bulunduğu için anlatıcının ya ikinci bir MAKEINSTANCE oluşturma yada kardinallik niteliğini kullanma seçeneği bulunmaktadır. Eğer iki MAKEINSTANCE oluşturulmuşsa (*iki defa*'nın signalID="s1" ile sinyal olduğu) bunlar aşağıdaki gibi görünecektir:

```
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PAST" aspect="NONE"
pos="VERB" signalID="s1"/>
```

```
<MAKEINSTANCE eiid="ei2" eventID="e1" tense="PAST" aspect="NONE"
pos="VERB" signalID="s1"/>
```

Eğer bunun yerine bir MAKEINSTANCE oluşturulmuşsa aşağıdaki gibi görünecektir:

```
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PAST" aspect="NONE"
pos="VERB" signalID="s1" cardinality="2"/>
```

4. Ali'un ders vermiş olması gerekirdi

Bu, kiplilik niteliğini içeren tek bir MAKEINSTANCE'e yol açar:

```
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PAST"
aspect="PERFECTIVE"
pos="VERB" modality="SHOULD"/>
```


5. *Ali ders vermedi*

Bu, kutupsallık niteliğiyle tek bir MAKEINSTANCE'a yol açar:

```
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PRESENT" aspect="NONE"
pos="VERB" polarity="NEG"/>
```

6.6 MAKEINSTANCE Etiketİ İçin BNF

attributes ::= eiid eventID tense aspect pos [polarity] [modality] (nitelikler, kutupsallık, kiplilik)

[signalID] [cardinality] (kardinallık)

eiid ::= ID

{eiid ::= EventInstanceID

EventInstanceID ::= ei<integer>}

eventID ::= IDREF

{eventID ::= EventID}

tense ::= 'PAST' | 'PRESENT' | 'FUTURE' | 'NONE' | 'INFINITIVE' |

'PRESPART' | 'PASTPART' (zaman – geçmiş, şimdiki, gelecek, hiçbir, mastar, şimdiki ortaç, geçmiş ortaç)

aspect ::= 'PROGRESSIVE' | 'PERFECTIVE' | 'PERFECTIVE_PROGRESSIVE' |

'NONE' (yön – ilerleyici, ortaç, ortaç_ilerleyici, hiçbir)

pos ::= 'ADJECTIVE' | 'NOUN' | 'VERB' | 'PREPOSITION' | 'OTHER'

polarity ::= 'NEG' | 'POS' {default, if absent, is 'POS'} (kutupsallık, kurulu ayar, eğer yoksa POS)

modality ::= CDATA (kiplilik)

signalID ::= IDREF (sinyal kimliği)

{signalID ::= SignalID} (sinyal kimliği)

cardinality ::= CDATA (kardinallık)

6.6.1 MAKEINSTANCE Nitelikleri

- a. **Olay Örnek Kimlik numarası (eiid)** Seçmeli-olmayan nitelik. Her bir olay örneği, tek bir kimlik numarası belirlenmelidir. Her bir MAKEINSTANCE etiketi oluşturulduğunda bu, anlatım aracıyla otomatik olarak tahsis edilecektir.
- b. **Olay kimlik numarası (olay kimliği)** Seçmeli olmayan nitelik. Her bir olay örneği, metinde bulunan özel bir OLAY ile ilişkilidir. Her MAKEINSTANCE etiketi oluşturulduğunda bu, anlatım aracı tarafından otomatik olarak tahsis edilecektir.
- c. **Zaman, Yön, Konuşma Bölümü (zaman, yön, pos)** Seçmeli-olmayan nitelikler. Bu nitelikler, işlem öncesinde sunulacaktır, fakat insan anlatıcının bunları düzeltmesi ve gerektiğinde eklemesi gerekecektir. Aşağıda, zaman, yön ve pos seçmek için birtakım kurallar bulunmaktadır. Zaman ve yön nitelikleri, fiili cümleciklerin gramer kategorileri arasındaki standart ayrımları yakalar. Zaman niteliğinin PRESENT (şimdiki), PAST (geçmiş), FUTURE (gelecek), INFINITIVE (matar), PRESPART (şimdiki ortaç), PASTPART (geçmiş ortaç) yada NONE (hiçbiri) değerleri taşıyabilir. Yön niteliği PROGRESSIVE (ilerleyici), PERFECTIVE (ortaç), PERFECTIVE_PROGRESSIVE (ortaç_ilerleyici) yada NONE (hiçbiri) değerleri alabilir. Pos niteliği, olaylar olarak işaretlenen cümleciklerin sentaks kategorileri arasındaki ayrımları yakalar. Bu nitelik, şu değerleri taşıyabilir: ADJECTIVE (sıfat), NOUN (isim), VERB (fiil), PREP. (edat).

- **Sonlu Fiiller:**

Tüm sonlu fiillerde pos = “FİİL” olacaktır. Zaman ve yön nitelikleri, aşağıdaki örneklerde belirtildiği şekilde belirlenecektir:

- Etken Hal

Zaman – “ŞİMDİKİ”

Fiil grubu	yön
ders verir	HİÇBİRİ
ders veriyor	İLERLEYİCİ
ders vermiş oldu	ORTAÇ
ders vermiş oluyor	ORTAÇ_İLERLEYİCİ

zaman=“GEÇMİŞ”

Fiil grubu	yön
------------	-----

ders verdi	HİÇBİRİ
ders veriyordu	İLERLEYİCİ
ders vermişti	ORTAÇ
ders veriyordu	ORTAÇ_İLERLEYİCİ

zaman = “GELECEK”

Fiil grubu	yön
ders verecek	HİÇBİRİ
ders veriyor olacak	HİÇBİRİ
ders veriyor olacak	İLERLEYİCİ
ders veriyor olacak	İLERLEYİCİ
ders vermiş olacak	ORTAÇ
ders veriyor olacak	ORTAÇ_İLERLEYİCİ

- Edilgen Hal

Zaman = “ŞİMDİKİ”

Fiil grubu	yön
ders verildi	HİÇBİRİ
ders veriliyor	İLERLEYİCİ
ders verilmiş oldu	ORTAÇ

zaman = “GEÇMİŞ”

Fiil grubu	yön =
ders verildi	HİÇBİRİ
ders veriliyordu	İLERLEYİCİ
ders verilmiş oldu	ORTAÇ

zaman = “GELECEK”

Fiil grubu	yön =
------------	-------

ders verilecek	HİÇBİRİ
ders verilecek	HİÇBİRİ
ders verilmiş olacak	ORTAÇ

- Yardımcı kip öncüllü fiiller

Burada iki farklı durum şart koşulmaktadır:

- Yardımcı kipin öncülü olduğu fiiller: -malı

Pos, zaman ve yönün nitelenmesi, sonlu fiillerle çok benzer olacaktır: pos="FİİL". Zaman ve yön, aşağıdaki örnekleri izleyecektir:

Zaman="ŞİMDİKİ"

Fiil grubu	yön=
ders vermeli	HİÇBİRİ
ders veriyor olmalı	İLERLEYİCİ
ders vermiş olmalı	ORTAÇ
ders vermiş olmalıdır	ORTAÇ_İLERLEYİCİ

zaman="GEÇMİŞ"

Fiil grubu	yön
ders vermeli	HİÇBİRİ
ders veriyor olmalı	İLERLEYİCİ

zaman="GELECEK"

Fiil grubu	yön=
ders vermesi gerekecek	HİÇBİRİ
ders veriyor olması gerekecek	İLERLEYİCİ

- Başka herhangi yardımcı kipin öncülü olduğu fiiller

Bu gruptaki yardımcı kipler şunları içerir: -malı, -ebilir, -ebilir, -ebilir, -ebilir, -malı ve -cak. Bu durumda, MAKEINSTANCE'daki nitelikler zaman = "HİÇBİRİ" ve pos="FİİL" olacaktır. Yön değeri, aşağıdaki örneklere göre tahsis edilecektir:

Zaman = “HİÇBİRİ”

Fiil grubu

yön=

ders verebilir

HİÇBİRİ

ders veriyor olabilir

İLERLEYİCİ

ders vermiş olabilirdi

ORTAÇ

ders veriyor olabilirdi

ORTAÇ_İLERLEYİCİ

- **Sonlu-olmayan Fiiller**

İngilizce’de birtakım bağlı cümlecikler, sonlu-olmayan fiiller içerir. Bunlar, mastar ve ortaç cümleciklerini içerir. Aşağıda bazı örnekler verilmiştir.

Şimdiki Ortaç: (“PRESPART”)

Kral Hüseyin, yeni bir girişim için destek arayışıyla dün Washington’a vardı.

Geçmiş Ortaç: (“PASPART”)

Sahnedede bulunan çelik levhalar, uzmanlar tarafından inceleniyor.

Mastar: (“INFINITIVE”)

Callahan, bilgileri bugün açıklamayı planlıyor.

Her bir durumda, uygun zaman değeri kullanılır: PRESPART, PASTPART ve INFINITIVE.

Eğer zaman INFINITIVE (mastar)a ayarlanmışsa yön niteliği, muhtemel değerlerinden herhangi birini alabilir:

Zaman=’INFINITIVE”

Fiil grubu

yön=

açık(lamak)

HİÇBİRİ

açıklıyor (olmak)	İLERLEYİCİ
açıklamış (olmak)	ORTAÇ
açıklıyor (olmak)	ORTAÇ_İLERLEYİCİ

Diğer yandan, eğer zaman PRESPART yada PASTPART'a ayarlanmışsa, yön niteliği aspect = "NONE" olarak ayarlanacaktır.

Zaman= "PRESPART"

Fiil grubu	yön=
arıyor	HİÇBİRİ

zaman = "PASTPART"

Fiil grubu	yön=
Buldu	HİÇBİRİ

Geçmiş ortaç biçimlerindeki fiillerin de edilgen yada ortaç yapıların parçası olarak da kullanılabileceğini not ediniz. The'nın da sıfat olarak işaretlenmeleri gereken (aşağıda tarif edilen) kullanımları vardır. PASTPART işareti, ortacın bağlı bir cümlecikte olduğu bu durumlar içindir, fakat aynı zamanda diğer kategorilerin hiçbirinde açıkça belirlenemediğinde de kullanılmalıdır.

Şimdiki ortaçtaki fiiller için de, (yukarıda tarif edilen) ilerleyici yapılarda olan fakat aynı zamanda (aşağıda tarif edilen) hem isim hem de sıfat kullanımları olan benzeri bir kural vardır. PRESPART işareti, ortacın bağlı bir cümlecikte olduğu durumlar için de amaçlanmıştır, fakat aynı zamanda ilerleyici, sıfat yada isim olduğunun açıkça belirlenemediğinde de kullanılmalıdır.

- **Sıfatlar**

Bir belirtme sıfatı, bir olayla işaretlendiğinde (bir sıfatın işaretlenebilir olup olmadığını belirlemeye dair kurallar için OLAYlar hakkındaki bölüme bakınız), ilgili MAKEINSTANCE'nının pos niteliği, pos = "ADJECTIVE" (sıfat) olarak ayarlanmış değere sahip olmalıdır.

Sıfatların zamanı ve yönü, bağlantılı fiil *to be*'nin yüklem tamamlayıcısının başlığı olmak haricinde NONE (hiçbiri) değerini taşıyacaktır. Bu durumda, zaman ve yön nitelikleri, bağlantılı fiilin zamanına ve yönüne karşılık gelen değerleri sunacaktır.² Aşağıdaki örnekler, ayrımı ortaya koymaktadır:

Zaman = “HİÇBİRİ”

Örnek	yön= “HİÇBİRİ”
Şüpheli, konuşmak istemiyor...	

Zaman = “ŞİMDİKİ”

Örnekler	
Eş, konuşmak istemiyor	yön =
Eş, konuşmak istemiyor halde	İLERLEYİCİ
Eş, konuşmak istemez halde	ORTAÇLI

Zaman = “GEÇMİŞ”

Örnekler	yön =
Eş, konuşmak istemiyordu.	HİÇBİRİ
Eş, konuşmak istemiyordu.	İLERLEYİCİ
Eş, konuşmak istemiyor durumdaydı.	ORTAÇLI

Zaman = “GELECEK”

Örnekler	yön =
Eş, konuşmaktan memnuniyet duyacak	HİÇBİRİ
Eş, konuşmaktan memnun olacak	ORTAÇLI

ADJECTIVE karşısında PASTPART Sıfatları, sonlu-olmayan geçmiş zaman

² Önerme fiillerinin, TimeML’de olaylar olarak etiketlenmediğini, fakat yüklem tamamlayıcılarının bu şekilde etiketlendiğini hatırlayınız.

ortaçlarından yada edilgenlerden ayırmak zor olabilir. Önemli bir ipucu, eğer bir cümleciği bir ipucu takip ederse, bu bir sıfat değildir. Aşağıdaki testler, bir biçimin sıfat olmasının kanıtıdır. Bunları, soru sorulabilir biçimin olduğu cümleye uygulayı:

- Derecelendirilebilirlik: Biçimin öncesinde derece zarfı gelebilir mi?
Çok şaşırmıştı.
Ondan daha çok şaşırmıştı.
- -un: -un biçimleri olmayan fiiller için, karşılık gelen bir –un sıfatı var mıdır?
Risk hakkında endişe duyamamaya devam ediyor.
SIFAT OLMAYAN: Kapı, muhafız tarafından kilitlenmişti.
- Yüklemlili konum: Biçim, aşağıdaki fiillerden biriyle yapı içerisinde mi oluyor, ve fiil diğerlerinden biriyle değiştirilebilir mi?
Olmak
Hissetmek
Bakmak
Kalmak
Görünmek
Ses vermek

ADJECTIVE karşısında PRESPART Sıfatları sonlu-olmayan şimdiki zaman ortaçlarından yada ilerleyicilerinden ayırmak da zordur. Yukarıda sayılan sıfatlık testleri, burada da geçerlidir. Buna ilaveten, isimlerden önce gelen şimdiki zaman ortaçlarının çoğu, sıfat olarak işaretlenmelidir:

çekici bir yüz
bekleyen limuzin
mevcut kurallar

Bu kuralın bir istisnası, yapının semantiğinin şimdiki zaman ortacının isim olarak kullanılmasını önerir:

bekleme odası

Bu durumda, şimdiki zaman ortacı, soyut bir faaliyete – beklemeye atıfta bulunur ve dolayısıyla olay olarak işaretlenmemelidir. Bu, beklemenin limuzinin yüklemi olduğu yukarıdaki örnekle zıtlık teşkil eder. –ing’nin isimden önce gelmesinin diğer örnekleri, soyut isimdir ve dolayısıyla olay olarak işaretlenmemelidir:

dikiş iğnesi

atış maçı

alkol içme sorunu

- **İsim türünden olanlar**

Bir isim türü, olay olarak işaretlendiğinde (bir isim türünün işaretlenebilir olup olmadığını belirlemek için yukarıdaki OLAYlara dair bölüme bakınız), ilgili MAKEINSTANCE’nin pos niteliği, pos = “NOUN” (isim) olarak ayarlanmalıdır.

Sıfatlarda olduğu gibi, *olmak* fiilinin yüklem tamamlayıcısının başlığı olma haricinde isimlerin zamanının ve yönünün NONE (hiçbiri) değeri olacaktır. Bu durumda, zaman ve yön nitelikleri, bağlantılı fiilinin zamanına ve yönüne karşılık gelen değerleri sunacaktır.

Zaman = “NONE” (hiçbiri)

Örnek

yön = “NONE” (hiçbiri)

Tehdit, ciddi dikkate yol açtı.

Zaman = “ŞİMDİKİ ZAMAN”

Örnekler

yön =

Görüşmeleri başlatma teşebbüsü var

HİÇBİRİ

Rahatsızlık veriyor

İLERLEYİCİ

Birçok aydır endişe oluşturmaya devam ediyor.

ORTAÇLI

Zaman = “GEÇMİŞ”

Örnekler

Görüşmeleri açma teşebbüsüydü.

Rahatsızlık oluyordu.

Birçok aydır bir endişe kaynağıydı.

yön=

HİÇBİRİ

İLERLEYİCİ

ORTAÇLI

Zaman = “GELECEK”

Örnekler

Bu, görüşmeleri açma teşebbüsü olacak.

Görüşmeleri açma teşebbüsü olurdu.

yön=

HİÇBİRİ

ORTAÇLI

İSİM (NOUN) karşısında PRESPART. Fiilli isimler, İSİM olarak işaretlenecektir. Aşağıdaki testler, bir biçimin İSİM olmasının kanıtıdır. Yine, hangi isim türlerinin fiilen işaretlenebilir olduğunu belirlemek için OLAYlara dair bölüme bakınız. Bu testler, biçimin olduğu bütün cümleye uygulanmalıdır:

- Çoğullar: Biçim, çoğul olabilir mi?
Binlercesi, etnik Arnavutların öldürülmesini protesto etmek için yürüyüşe geçti...
- Sıfat değiştirme: Biçim, bir sıfatla değiştirilebilir mi?
Dikkatli araba sürüşü, kazayı önledi.
- -in’de doğrudan nesne: Fiilin nesnesi, -in cümlecğinde ifade ediliyor mu?
Ali’nin (tekrar eden) kapı çarpması, Ayşey’yi kızdırdı.
- Öncesinde isim gelmesi: Biçimin öncesinde isim geliyor mu?
Fabrika kapanması

Aşağıdaki testler, bir biçimin şimdiki zaman ortacı olmasının kanıtıdır:

- Edat değişikliği: Biçim, bir edatla değiştirilebilir mi?
Ali, dikkatlice araba sürerek kazayı önledi.

- NP'deki doğrudan nesne: Fiilin nesnesi, basit bir NP'de ifade edilmiş midir?
Ali'nin (tekrar eden) kapı kapaması, Ayşe'yi kızdırdı.

- **Edatlı Cümlecikler**

Edatlı bir cümle olay olarak işaretlendiğinde, ilgili MAKEINSTANCE'ın pos niteliği, pos = "PREP" olarak ayarlanmalıdır.

Sıfatlara ve isimlere benzer biçimde, edat cümleciklerinin zaman ve yön nitelikleri, PP'nin bağlantılı fiilinin yüklemli tamamlayıcısı olması haricinde NONE (hiçbiri) olarak ayarlanacaktır. Bu durumda, zaman ve yön nitelikleri, bağlantı fiilinin zaman ve yönüne karşılık gelen değerleri taşıyacaktır:

Zaman = "HİÇBİRİ"

Örnek

yön = "HİÇBİRİ"

Thomson, turizm liderleriyle konuşmak için Hindistan'da...

Zaman = "ŞİMDİKİ"

Örnekler

yön =

Proje, sorun içinde.

HİÇBİRİ

Proje, sorun içine giriyor.

İLERLEYİCİ

Proje, birçok aydır sorun içindeydi.

ORTAÇLI

Zaman = "GEÇMİŞ"

Örnekler

yön =

Proje, sorun içindeydi.

HİÇBİRİ

Proje, sorun içinde bulunuyordu.

İLERLEYİCİ

Proje, birçok aydır sorun içindeydi.

ORTAÇLI

Zaman = "GELECEK"

Örnekler

yön =

Proje, sorun içinde olacak.	HİÇBİRİ
Proje, sorun içine girmiş olacak.	ORTAÇLI

- d. Kutupsallık ve Kiplilik (Polarity, Modality)** Bir olay örneğinin kutupsallığı, boolean niteliği, polarity ile temsil edilen gerekli bir nitelikler. Eğer ‘NEG’ olarak ayarlanmışsa, olay örneği dışarıda tutulur. Eğer ‘POS’ olarak ayarlanmışsa yada anlatımda mevcut değilse, olay örneği olumsuzlanmamıştır. Kiplilik niteliği, sadece örneği değiştiren bir kipli kelime bulunması halinde mevcuttur. Ön-işlem, bu niteliklerin tahsis edilmesini göz önünde bulundurmalı, fakat insan anlatıcısı bunları düzeltme yada ekleme gereksinimi duyabilir.
- e. Sayısalılık ve Sinayl Kimliği (Cardinality, Signal ID)** Seçmeli nitelikler. Kardinallik, sadece değerini sağlayan bir sinyal bulunması halinde mevcuttur. Sinyal, bir olayın çoklu örneklerinin insan anlatıcısı tarafından oluşturulması halinde de mevcuttur. TimeML’in en son özelliklerine kümelerin eklenmesiyle, bu nitelikler daha az sıklıkta kullanılacaktır.

6.7 Bağlantı Etiketleri: <TLINK>, <SLINK> Ve <ALINK>

Üç türde bağlantı etiketi bulunmaktadır. Bağlantıların nasıl anlatıldığını ayrıntılı olarak açıklamaya geçmeden önce her birinin işlevi, burada sunulacaktır. TimeML OLAYlarının hiçbir zaman bağlantıya katılmadığını hatırlayınız. Bunun yerine, karşılık olay örneği kimlikleri kullanılır. Aşağıdaki açıklamada, ne var ki, “olaylar ve “olay örnekleri” kelimeleri birbirleriyle değiştirilebilir biçimde kullanılır.

6.8 <TLINK> Etiketi

TLINK yada Zamansal Bağlantı, olaylar, zamanlar arasındaki yada bir olayla zaman arasındaki zamansal ilişkileri temsil eder. Mukabil olay örneği kimlikleri vasıtasıyla OLAYların TLINK’e katıldığını hatırlayınız. Mevcut açıklamada, ne var ki “olaylar” ve “olay örnekleri” kelimeleri, birbirleriyle değiştirilebilir şekilde kullanılır. Aynı gözlem, sırasıyla bölümler 3.2 ve 3.3’te sunulan SLINK’ler ve ALINK’ler için de geçerlidir.

TLINK, aşağıdaki gibi olmaları halinde açık hale getirerek sözkonusu taraflar arasında bağlantı oluşturur:

1. Eşzamanlı

İki olay örneği hakkında, eğer aynı anda olmuşsa yada bağlamı yönünden geçici olarak ayırt edilemezse, yani zamanlarının daha da ayırt edilmesi metnin zamansal yorumlanmasında hiçbir fark yapmayacak şekilde aynı zamana yeterince yakın oluyorsa eşzamanlı olarak hüküm verilir.

2. Biri diğerinden önce:

Öldürme ve yakalanma olayları arasındaki aşağıdaki örnekte olduğu gibi:

Polis, 14 kadının öldürülmesini araştırdı. Bu örneklerin altısında, şüpheliler zaten yakalandı.

3. Biri diğerinden sonra:

Bu, önceki ilişkinin sadece tersidir. Bu nedenle, eğer doğrusallık değiştirilmişse önceki örneğin iki olayı, alternatif olarak sonraki ilişki halinde anlatılabilir.

4. Biri diğerinin hemen öncesinde:

Çarptı ile öldü arasında olduğu gibi aşağıdaki cümledeki gibi:

Uçak dağa çarptığında tüm yolcular öldü.

5. Biri diğerinin hemen ardından:

Bu, önceki ilişkinin tersidir.

6. Biri diğerini içerir şekilde:

Aşağıdaki örnekte zamansal ifadeyle ve olay arasında olduğu gibi:

Ali, Boston'a geçen Perşembe vardı.

7. Biri diğerine dahil edilerek:

Öncekine ters ilişki.

8. Diğer esnasında biri olur:

Bir süre boyunca devam eden durumlar yada olaylar için özellikle uygulanabilir, örneğin:

James, iki yıl süreyle CTO idi.

Ali, Pazartesi günü 20 dakikalığına ders verdi.

9. Diğer esnasında biri anlatılıyor:

Öncekine göre ters ilişki.

10. Birinin, diğerinin başlangıcı olması:

Aşağıdaki örnekte zamansal ifadelerin ilkiyle olay arasında geçerli olduğu gibi:

Ali, öğleden sonra saat 6:00 ile 7:00 arasında spor salonundaydı.

11. Birinin diğeri tarafından başlatılması:

Biraz önce sunulanla ilişkili olarak ters ilişki.

12. Birinin diğerinin sonu olması:

Ali, öğleden sonra saat 6:00 ile 7:00 arasında spor salonundaydı.

13. Birinin diğeriyle sona erdirilmesi:

Az önce sunulmuş olanla ters ilişki.

Buna ilaveten, TLINK'ler aşağıdaki durumlarda da kullanılır.

14. Olay kimliği

Olay kimliği, TLINK yoluyla da anlatılır. Olay kimliğinin anlatımın kapanış bölümünde ele alınan çok önemli bir ilişki olduğunu not ediniz. Bunun için, lütfen tüm kimlik bağlantılarının anlatıldığından emin olunuz. Örn.:

Ali, Boston'a kadar arabayla gitti. Arabayla gidişi esnasında poğaça yedi.

15. Metinde 'küme / alt-küme ilişkisi' gerçekleştiğinde.

Bir örneği şudur:

Polis, 14 kadının öldürülmesini inceledi. Olayların altısında, şüpheliler zaten yakalandı.

TimeML'in şimdiki versiyonunda, bu 'küme / alt-küme' ilişkisini yakalayamıyoruz. Şimdilik yapılacak olan, 'küme'lerin her biri için MAKEINSTANCE, kardinal çoğullu (yada 14?) öldürmeler için bir MAKEINSTANCE ve kardinal çoğullu (yada 6?) durumlar için bir MAKEINSTANCE oluşturmak ve daha sonra bu iki örneği, TLINK yoluyla IS_INCLUDED yada INCLUDES bulunan TLINK üzerinden bağlamaktır.

6.8.1 TLINK'lerin nasıl anlatılacağı

Burada ve SLINK'lerle ALINK'lerin aşağıdaki bölümlerinde, bağlantı türlerinin her

birinin nasıl oluşturulacağını ortaya koymak için örnekler kullanıyoruz. Bu örneklerde, olayların, zamanların ve sinyallerin ayrıntılı anlatımını vermiyoruz – bunların anlatımına dair talimatlar için lütfen uygun bölüme başvurunuz. Aynı zamanda, örneklerle ilgili bu taraflar için sadece işaretlemeyi gösteriyoruz.

Olaylar arasında var olan zamansal bir ilişkinin yada bir olayın ve zamanın anlatılmasının her gerektiğinde bir TLINK oluşturulmalıdır. Bu, olay kimliğinin önemli ilişkisini içerir.³ Örnekler:

1. Ali, Pazartesi günü ders verdi

Olayla zamansal ifade arasında geçerli bulunan zamansal ilişki, -de sinyalinde belirtildiği şekilde, aşağıdaki TLINK'in sunulmasıyla işaretlenir:

```
<TLINK eventInstanceID="ei1" relatedToTime="t1" signalID="s1"
relType="IS_INCLUDED"/>
```

2. Ali, Pazartesi ve Salı günü ders verdi

Yukarıda bölüm 2.4'te açıklandığı gibi, iki MAKEINSTANCE *ders verdi* örneklerini temsil eder:

```
<MAKEINSTANCE eiid="ei1" eventID="e5"/>
<MAKEINSTANCE eiid="ei2" eventID="e5"/>
```

Bu iki olay ve iki zamansal ifade arasında var olan zamansal ilişki, aşağıdaki iki TLINK'in sunulmasıyla anlatılır:

```
<TLINK eventInstanceID="ei1" relatedToTime="t1" signalID="s1"
relType="IS_INCLUDED"/>
```

```
<TLINK eventInstanceID="ei1" relatedToTime="t2" signalID="s1"
relType="IS_INCLUDED"/>
```

³ Olay kimliğinin bağlantıda hiçbir zaman kullanılmadığını hatırlayınız. Bunun yerine, olay örneği kimliği, tüm durumlarda kullanılır (yani, MAKEINSTANCE'de kullanılan kimlik).

3. Ali, her Pazartesi ders verdi

Pazartesi'nin çoklu örneklerini temsil eden TIMEX3 aşağıdaki gibi görünür:

```
<TIMEX3 tid="t1" type="SET" value="XXXX-WXX-1" quant="EVERY">
```

her Pazartesi

```
</TIMEX3>
```

Olayla zamansal ifade arasında bulunan zamansal ilişkiyi temsil eden TLINK şöyle görünür:

```
<TLINK eventInstanceID="ei9" relatedToTime="t1"
```

```
relType="IS_INCLUDED"/>
```

4. Ali, Pazartesi günü 20 dakika ders verdi.

Ders verdi olayını temsil eden MAKEINSTANCE şöyle görünür:

```
<MAKEINSTANCE eiid="ei4" eventID="e4"/>
```

İki TLINK sunulmalıdır⁴. Bir TLINK, *ders verdi* olayının 20 dakika boyunca devam ettiği gerçeğini yakalar, ve bir TLINK *20 dakika* zamansal ifadesinin *Pazartesi*'ne dahil olduğu gerçeğini yakalar.

```
<TLINK eventInstanceID="ei4" relatedToTime="t2" signalID="s5"
```

```
relType="DURING"/>
```

⁴ Genelde olduğu gibi, bunlar metinden çıkarılabilecek olan tek TLINK'ler değildir. Ancak, işlem-sonrası algoritmalar, anlatıcının dışarıda bıraktığı bağlantıları doldurabilmelidir.


```
<TLINK tid="t2" relatedToTime="t3" signalID="s6"
relType="IS_INCLUDED"/>
```

5. Ali, Boston'a kadar arabayla gitti. Arabayla gidişi esnasında poğaça yedi.
Arabayla gitti ve *arabayla gitmek* olaylarını sunan MAKEINSTANCE'lar aşağıdaki gibi görünmektedir:

```
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
<MAKEINSTANCE eiid="ei2" eventID="e2"/>
```

Bu iki olayın kimliğini temsil etmek üzere sunulması gereken TLINK, şöyle görünmektedir:

```
<TLINK eventInstanceID="ei1" relatedToEvent="ei2" relType="IDENTITY"/>
```

Yedi ile *arabayla gitti* arasındaki zamansal ilişkiyi temsil edecek olan TLINK'i dahil etmediğimizi lütfen not ediniz.

6. Polis, 14 kadının öldürülmesini inceledi. Bu örneklerin altısında, şüpheliler zaten yakalandı.
Öldürdü ve *örnekler* olaylarını temsil eden MAKEINSTANCE'lar aşağıdaki gibi görünmektedir:

```
<MAKEINSTANCE eiid="ei1" eventID="e1" cardinality="14"/>
<MAKEINSTANCE eiid="ei2" eventID="e2" cardinality="6">
```

Bu iki olay arasındaki ilişkiyi temsil etmek üzere sunulması gereken TLINK, aşağıdaki gibi görünmektedir:

```
<TLINK                                eventInstanceID="ei1"                                relatedToEvent="ei2"
```

relType="INCLUDES"/>

6.8.2 TLINK etiketi için BNF

```

attributes ::= [lid] [origin] (eventInstanceID | timeID) [signalID] [syntax]
(relatedToEventInstance | relatedToTime) relType

lid ::= ID
{lid ::= LinkID
LinkID ::= l<integer>}

origin ::= CDATA
eventInstanceID ::= IDREF
{eventInstanceID ::= EventInstanceID}

timeID ::= IDREF
{timeID ::= TimeID}

signalID ::= IDREF
{signalID ::= SignalID}

relatedToEventInstance ::= IDREF
{relatedToEventInstance ::= EventInstanceID}

relatedToTime ::= IDREF
{relatedToTime ::= TimeID}

relType ::= 'BEFORE' | 'AFTER' | 'INCLUDES' | 'IS_INCLUDED' |
'DURING' | 'DURING_INV' | 'SIMULTANEOUS' | 'IAFTER' | 'IBEFORE' |
'IDENTITY' | 'BEGINS' | 'ENDS' | 'BEGUN_BY' | 'ENDED_BY' (önce, sonra,
içerir, dahil edildi, esnasında, _INV esnasında, eşzamanlı, sonrasında, öncesinde,
kimlik, başlar, sona erer, tarafından başlatıldı, tarafından sona erdirildi)

syntax ::= CDATA

```

6.8.3 TLINK'ler için Nitelikler

1. eventInstanceID (olay örnek kimliği) yada timeID (zaman kimliği)

Zorunlu nitelik (bu gereksinimlerin biri yada diğer mevcut bulunmalıdır). Bu,

zamansal bağlantıya dahil olan eventInstance yada timeID'nin kimliğidir.

2. signal ID (sinyal kimliği)

Seçmeli nitelik. Eğer taraflar arasında mevcut bulunan zamansal ilişki, metinde açıkça sinyal verilmişse, o sinyalin kimliğinin burada temin edilmesi gerekir.

3. relatedToEventInstance (Olayla İlgili Örneği) yada relatedToTime (Zamanla İlgili)

Zorunlu nitelik (bu gereksinimlerin biri yada diğeri mevcut bulunmalıdır). Bu, ID=eventInstanceID taşıyan olay örneğiyle yada ID=timeID taşıyan zaman ifadesiyle ilişkili tarafın kimliğidir.

4. relType

Zorunluluk niteliği. Bu, taraflar arasında geçerli bulunan zamansal ilişkidir. Muhtemel değerler şunlardır: BEFORE, AFTER, INCLUDES, IS INCLUDED, DURING, DURING INV, SIMULTANEOUS, IAFter, IBEFORE, IDENTITY, BEGINS, ENDS, BEGUN BY, ENDED BY. Bunlar, TLINK'lere dair bölüm 3.1'in başında verilen talimatlara göre tahsis edilir. Her bir TLINK başına tahsis edilen sadece bir ilişki olacaktır.

5. origin (köken)

Seçmeli nitelik. Bu nitelik, bir TLINK'in manüel olarak yada kapanmayla oluşturulup oluşturulmadığını gösterir.

6.9 <SLINK> Etiketi

İki olay arasındaki ilişkileri sunan bağlamlar için SLINK yada Bağlı Kılma Bağlantısı kullanılır⁵. SLINK'ler aşağıdaki türlerden biridir:

- **Kipli:**

Bu ilişki, muhtemel bir dünyaya atıf sunan olaylarla öne çıkarılmaktadır – başlıca, I_ACTION'lar ve I_STATE'ler:

Ali, Mary'ye bira alma sözü verdi.

Mary, Ali'un biraz şarap almasını istedi.

- **Keyfiyetli:**

⁵ Kutupsallığı ve kipliliği temsil etmek üzere bir olayla sinyal arasında bulunan SLINK'ler için izin verilen TimeML'in önceki versiyonları, fakat bu bilgiler halen MAKEINSTANCE etiketinde temsil edilmektedir – altbölüm 2.4.3'e bakınız.

Belli fiiller, kendi tezlerinin doğruluğunun bir gerekliliğini (yada ön varsayımını) sunar. Bunlar, **unutmak** (zamanlı tamamlayıcısıyla), **üzüntü duymak** yada **yönetmek**'i içerir:

Ali, geçen yıl Boston'da olduğunu unuttu.

Mary, Ali'la evlenmediğine üzülüyor.

Ali, partiden ayrılmayı başardı.

- **Karşı-keyfiyetli:**

Önceki ilişkinin aksine, bu durumda olay, kendi tezinin doğru olmadığı hakkında bir ön-varsayım sunar: unutmak, -mamak (geçmiş zamanda), önlemek, iptal etmek, kaçınmak, reddetmek, v.s.

Ali, şarap almayı unuttu.

Mary, Ali ile evlenemedi.

Ali, boşanmayı önledi.

- **Kanıtsal:**

Kanıtsal ilişkiler, olağan olarak REPORTING (bildirme) yada PERCEPTION (kavrama) olaylarıyla sunulur:

Ali, şarap aldığını söyledi.

Mary, Ali'ü sadece bira taşırken gördü.

- **Olumsuz kanıtsal:**

Olumsuz kutupsallık taşıyan BİLDİRME ve KAVRAMA olaylarıyla sunulur:

Ali, sadece bira aldığını reddetti.

- **Şartlılık:**

Herhangi iki olay örneği arasında şartlı bir ilişki oluşabilir ve genellikle *eğer* gibi bir sinyal eşlik eder. Şartlı'nın sonradan geleni, sunan olay örneğinin yerini alır (koyu karakterle) ve sondaki, (altı çizili) bağlı olayın yerini alır:

Eğer Ali, birayı getirirse, Mary patatesleri getirecek.

SLINK’ler aşağıdaki özelliklerde olabilir:

1. Sözlük-esaslı:

Bunlar, genelde cümlecik tamamlayıcı yada isim türünü işaret eden olayla başlık atılan NP alan bir sınıf olayıyla I_ACTION, I_STATE, PERCEPTION yada REPORTING başlatılır. SLINK, bu olaylarla tamamlayıcı arasında oluşturulur.

Her bir RAPORLAMA ve KAVRAMA olayı için, bir SLINK sunulmalıdır. Aşağıdaki örnekte, RAPORLAMA ve KAVRAMA olayları koyu karakterdeyken bağlı olayların altı çizilmiştir:

15 dakika sonra, diğer uçağın Dünya Ticaret Merkezi’ne çarptığını gördüm.

“Jet yada roket gibi ses çıkarıyordu,” dedi Eddie Gonzalez.

Benzer biçimde, her bir I_ACTION yada I_STATE için kastedilen olayla (koyu karakterde) bağlı olay (altı çizili) arasındaki ilişkiyi ifade eden bir SLINK sunulur:

Komutan Mehmet, bu farkı 1995’teki bir mektubunda açıklamaya teşebbüs etti.

Bizimle ilgili olan kararlara doğrudan katılmak, bizi yönetenleri kontrol etmek istiyoruz.

Bağlı olay sınıfı, aşağıdaki biçimde SLINK ilişkisi türünü kısıtlamaktadır:

i. KAVRAMA olayları:

Her zaman, EVIDENTIAL (kanıtsal) yada NEG_EVIDENTIAL (olumsuz kanıtsal) türde SLINK’leri sunacaklardır.

ii. I_ACTION, I_STATE olayları:

MODAL (kipli), FACTIVE (keyfiyetli) ve COUNTER_FACTIVE (karşı keyfiyetli) türde SLINK’leri sunabilirler.

iii. RAPORLAMA olayları:

Herhangi türde SLINK sunabilirler.

2. Yapısal-esaslı:

(a) Amaç Cümlecikleri:

Amaç cümlecği içeren bir cümlede, SLINK ana cümlecikteki olayı (koyu karakterli) ve onu değiştiren amaç cümlecğindeki (altı çizili) ilişkilendirir.

Çevre komisyonu, insanların radyoaktif atıklara maruz kalmamasını temin etmek için mevzuatı kabul etmelidir.

(b) Şartlılık yapıları:

Şartlı bir yapıda SLINK, sonraki cümlecikteki olayı sonuç cümlecğindekiyle ilişkilendirir.

2 Aralık'ta Mehmet, eğer çatışma bölgesi askersizleştirilirse görüşme masasına geri dönmeyi taahhüt etti.

6.9.1 SLINK'lerin nasıl anlatılacağı

1. Sözlük-esaslı SLINK'ler:

Anlatıcı, kendisi tarafından bağlı kılınan her bir olaya bağlı olayı ilişkilendiren bir SLINK yerleştirmelidir:

- Ali, Pazartesi günü ders verdiğini söyledi.

Ders verdi olayının *söyledi* olayıyla bildirildiği keyfiyetini ifade etmek için aşağıdaki SLINK oluşturulmuştur:

```
<SLINK eventInstanceID="ei2" subordinatedEventInstance="ei3"
relType="EVIDENTIAL"/>
```

- Ali, Pazartesi günü ders verdiğini yalanladı.

Ders verdi olayının *yalanlandı* olayıyla bildirildiği keyfiyetini ifade etmek için, aşağıdaki SLINK oluşturulmuştur:

```
<SLINK eventInstanceID="ei1" subordinatedEventInstance="ei2"
```

relType="NEG_EVIDENTIAL"/>

Bazı durumlarda, aynı bağlı kılıcı olay, birden fazla SLINK sunacaktır. Örneğin, aşağıdaki örnekte *söyledi* olayı iki olaya s-link yapılmıştır: *sayıldı* ve *verdi*.

- Rita, adını doğru olarak saydıklarını fakat kendisi için yanlış bir adres verdiklerini söyledi.

<SLINK eventInstanceID="ei1" subordinatedEventInstance="ei2"

relType="EVIDENTIAL"/>

<SLINK eventInstanceID="ei1" subordinatedEventInstance="ei3"

relType="EVIDENTIAL"/>

2. Yapısal-esaslı SLINK'ler:

(a) Amaç Cümlecikleri:

Ana cümlecikteki olay, eventInstanceID niteliğinin değerine karşılık gelecektir. Amaç cümleciğindeki olay, subordinatedEvent (bağlı kılınan Olay) değeri olarak alınacaktır. Bu SLINK'ler her zaman relType="MODAL" olacaktır. –e edatı, signalID niteliğinin değeri olarak anılacaktır.

- Çevre komisyonu, insanların radyoaktif atıklara maruz kalmamasını temin etmek için mevzuat kabul etmelidir.

<SLINK eventInstanceID="ei1" subordinatedEventInstance="ei2"

signalID="s1" relType="MODAL"/>

(b) Şartlı yapılar:

Sonraki cümlecikteki olay, eventInstanceID niteliğindeki değere karşılık gelir. Sondaki, subordinatedEvent değerine. Şartlı bağlaç (*eğer, zamanda*) signalID niteliğinin değeri olarak alınacaktır. Bu SLINK'ler her zaman relType="ŞARTLI" olacaktır.

- Meksika, eğer BM İnsan Hakları Komisyonu'nda oya sunulursa Guantanamo'ya dair bir araştırmayı desteklemeyi taahhüt etti.

```
<SLINK eventInstanceID="ei1" subordinatedEventInstance="ei2"
signalID="s1" relType="CONDITIONAL"/>
```

Örneğin sonradan gelenin koordinasyon olduğu aşağıdaki şartlı yapıda, yapısal-esaslı SLINK'lerde aynı olayın birkaç SLINK'te mevcut bulunması da mümkündür. Bu durumda, tekrar edilen olay, bağlı olandır (*dönme*), zira son cümlecikteki olaydır.

- 2 Aralık'ta Mehmet, çatışma bölgesinin askersizleştirilmesi halinde görüşme masasına geri dönmeye söz verdi, Kongre yerli halk haklarına ve kültürüne dair bir yasayı geçirdi ve 100 civarında Zapatistalı tutuklu serbest bırakıldı.

```
<SLINK eventInstanceID="ei2" subordinatedEventInstance="ei1"
signalID="s1" relType="CONDITIONAL"/>
```

```
<SLINK eventInstanceID="ei3" subordinatedEventInstance="ei1"
signalID="s1" relType="CONDITIONAL"/>
```

```
<SLINK eventInstanceID="ei4" subordinatedEventInstance="ei1"
signalID="s1" relType="CONDITIONAL"/>
```

6.9.2 SLINK etiketi için BNF

attributes ::= [lid] eventInstanceID

[signalID] subordinatedEventInstance relType [syntax]

lid ::= ID


```

{lid ::= LinkID
LinkID ::= l<integer>}
eventInstanceID ::= IDREF
{eventInstanceID ::= EventInstanceID}
subordinatedEventInstance ::= IDREF
{subordinatedEventInstance ::= EventInstanceID}
signalID ::= IDREF
{signalID ::= SignalID}
relType ::= 'MODAL' | 'EVIDENTIAL' | 'NEG_EVIDENTIAL'
| 'FACTIVE' | 'COUNTER_FACTIVE' | 'CONDITIONAL'
syntax ::= CDATA

```

6.9.3 SLINK'ler için Nitelikler

1. eventInstanceID (olay Örnek Kimliği)

Zorunlu nitelik. Bu, bağlı kılma bağlantısında bulunan eventInstance kimliğidir.

2. subordinatedEventInstance (bağlı kılınan Olay Örneği)

Zorunlu nitelik. Bu, ID=eventInstanceID taşıyan olay örneğinin ilişkili olduğu, bağlı kılınan olay örneğinin kimliğidir.

3. signalID (sinyal kimliği)

Seçmeli nitelik. Eğer olaylar arasında mevcut bulunan bağlı kılma ilişkisi, metinde açıkça sinyal verilmişse, o sinyalin kimliğinin burada doldurulması gerekir.

4. relType

Zorunlu nitelik. Bu, taraflar arasında bulunan zamansal ilişkidir. Muhtemel değerler şunlardır: MODAL (kipli), EVIDENTIAL (kanıtsal), NEG_EVIDENTIAL (olumsuz kanıtsal), FACTIVE (keyfiyetli), COUNTER_FACTIVE (karşı keyfiyetli), CONDITIONAL (şartlı). Bunlar, SLINK'lere dair bölüm 3.2'nin başında verilen talimatlara göre tahsis edilir.

6.10 <ALINK> Etiketi

Bir ALINK yada Yönsel Bağlantı, bir yönsel olayla tez olayı arasındaki ilişkiyi temsil eder. Kodu çözülecek yönsel ilişkilerin örnekleri şunlardır:

1.Başlatma:

Ali okumaya başladı.

2.Zirveye Çıkma:

Ali, masanın montajını bitirdi.

3.Sona Erme:

Ali, konuşmayı bitirdi.

4.Devam Etme:

Ali, konuşmaya devam etti.

5.Yeniden Başlama:

Ali, konuşmaya yeniden başladı.

6.10.1 ALINK'lerin nasıl anlatılacağı

ALINK'ler yönsel fiiller (ve diğer ifadeler) ile atıfta bulundukları olay arasındaki bağlantıyı temsil eder. Örnekler şunlardır:

1. Ali, okumaya başladı.

İki olayın iki MAKEINSTANCE'ı aşağıdakilerdir:

```
<MAKEINSTANCE eiid="ei5" eventID="e2"/>
```

```
<MAKEINSTANCE eiid="ei6" eventID="e3"/>
```

Başlatılan yönsel fiille *okunan* olay arasında oluşturulması gereken ALINK, aşağıdakidir:

```
<ALINK eventInstanceID="ei5" relatedToEventInstance="ei6"
relType="INITIATES"/>
```

2. Ali, okumayı bitirdi.

İki olayın iki MAKEINSTANCE'ı aşağıdakilerdir:

```
<MAKEINSTANCE eiid="ei5" eventID="e2"/>
```

```
<MAKEINSTANCE eiid="ei6" eventID="e3"/>
```

Başlatılan yönsel fiille öğretme olayı arasında oluşturulması gereken ALINK, aşağıdakidir:

```
<ALINK eventInstanceID="ei5" relatedToEventInstance="ei6"
relType="TERMINATES"/>
```

6.10.2 ALINK etiketi için BNF

```
attributes ::= [lid] eventInstanceID [signalID]
```

```
relatedToEventInstance relType [syntax]
```

```
lid ::= ID
```

```
{lid ::= LinkID
```

```
LinkID ::= l<integer>}
```

```
eventInstanceID ::= ID
```

```
{eventInstanceID ::= EventInstanceID}
```

```
signalID ::= IDREF
```

```
{signalID ::= SignalID}
```

```
relatedToEventInstance ::= IDREF
```

```
{relatedToEventInstance ::= EventInstanceID}
```

```
relType ::= 'INITIATES' | 'CULMINATES' | 'TERMINATES'
```

```
| 'CONTINUES' | 'REINITIATES' (başlatır, zirveye çıkarır, sona erdirir, devam eder,
yeniden başlatır)
```

syntax ::= CDATA

6.10.3 ALINK'ler için Nitelikler

1. eventInstanceID (olay Örnek Kimliği)

Zorunlu nitelik. Yönsel bağlantıda bulunan (yönsel) eventInstance kimliğidir.

2. signalID (sinyal kimliği)

Seçmeli nitelik. Eğer olaylar arasındaki yönsel ilişki, metinde açıkça sinyal verilmişse, o sinyalin kimliğinin burada doldurulması gerekir. Sinyal örneği, *Gemi batmaya başladı*da –ya'dır.

3. relatedToEventInstance (olayla ilgili Örnek)

Zorunlu nitelik. Yönsel olayla ilgili olay örneğinin kimliğidir.

4. relType

Zorunlu nitelik. Taraflar arasında mevcut bulunan zamansal ilişkidir. Muhtemel değerler şunlardır: INITIATES, CULMINATES, TERMINATES, CONTINUES, REINITIATES.

6.10.4 Tüm LINK etiketleri için ek nitelikler

Bağlantı Kimliği ve Sözdizimi (Link ID, Syntax)

Kapak ve sözdizimi nitelikleri, tüm TimeML bağlantılarında seçmelidir. Kapak, her bir TimeML bağlantısına bir kimlik numarası verir. Sözdizimi, bağlantı oluşturmada (örn. otomatik bir anlatım aracından) kullanılan her türlü sözdizimi düzenini korur.

6.11 Karmaşık Timex Örnekleri

1. Ali, dün 2 gün önce ayrıldı.

Ali

<EVENT eid="e1" class="OCCURRENCE" >

ayrıldı

</EVENT>

<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PAST"
aspect="PERFECTIVE"/>

<TIMEX3 tid="t1" type="DURATION" value="P2D" beginPoint="t2" endPoint="t3">

2 gün

</TIMEX3>

<SIGNAL sid="s1">

önce

</SIGNAL>

<TIMEX3 tid="t2" type="DATE" value="2002-07-10" temporalFunction="true"
anchorTimeID="t0">

dün

</TIMEX3>

<TIMEX3 tid="t3" type="DATE" value="2002-07-08" temporalFunction="true"
anchorTimeID="t1"/>

<TLINK timeID="t1" relatedToTime="t2" signalID="s1" relType="BEFORE"/>

<TLINK eventInstanceID="ei1" relatedToTime="t3" relType="IS_INCLUDED"/>

DURATION (süre) olarak anlatılan TIMEX3, başlangıç ve son nokta bilgilerini içerir. Anlatıcı, aynı zamanda bu bilgilerle birlikte ilave TLINK'ler içermiş olabilir.

“Dün” için TIMEX3’ün nitelik türü, zamansal çıpa “t0”a (DCT) nispi olan zamansal bir işlevle hesaplanabilecek bir TARİH belirtir. Benzer biçimde, nihai TIMEX3 de “t+1” olarak anlatılan, başlangıç SÜRESİNE nispi olan zamansal işlevle hesaplanabilir.

2.Önümüzdeki Perşembe’den sonraki iki hafta içinde tatile çıkıyorum.

Ben

<EVENT eid="e1" class="OCCURRENCE">

çıkıyorum

</EVENT>

<MAKEINSTANCE eiid="ei1" eventID="e1" tense="FUTURE"
aspect="PROGRESSIVE"/>

tatile

<TIMEX3 tid="t1" type="DURATION" value="P2W" beginPoint="t2" endPoint="t3">

iki hafta

</TIMEX3>

<SIGNAL sid="s1">

den itibaren

</SIGNAL>

<TIMEX3 tid="t2" type="DATE" value="2002-07-02" temporalFunction="true"
anchorTimeID="t0">

önümüzdeki Salı

</TIMEX3>

<TIMEX3 tid="t3" type="DATE" value="2002-07-23" temporalFunction="true"
anchorTimeID="t1"/>

<TLINK timeID="t1" relatedToTime="t2" signalID="s1" relType="AFTER"/>

<TLINK eventInstanceID="ei1" relatedToTime="t3" relType="IS_INCLUDED"/>

3.Üç yıl önce bugün büyük bir deprem, Los Angeles'i vurdu.

Büyük bir

<EVENT eid="e1" class="OCCURRENCE" >

depren

</EVENT>

<MAKEINSTANCE eiid="ei1" eventID="e1" tense="NONE" aspect="NONE"
pos="NOUN"/>

<EVENT eid="e2" class="OCCURRENCE" >

vurdu

</EVENT>

<MAKEINSTANCE eiid="ei2" eventID="e2" tense="PAST"
aspect="PERFECTIVE"/>

Los Angeles'ı

<TIMEX3 tid="t1" type="DURATION" value="P3Y" beginPoint="t2" endPoint="t3">

üç yıl önce

</TIMEX3>

<TIMEX3 tid="t2" type="DATE" value="2002-07-12" temporalFunction="true"
anchorTimeID="t0">

bugün

</TIMEX3>

<TIMEX3 tid="t3" type="DATE" value="1999-07-12" temporalFunction="true"
anchorTimeID="t1"/>

<TLINK eventInstanceID="ei1" relatedToEvent="ei2" relType="IBEFORER"/>

<TLINK eventInstanceID="ei1" relatedToTime="t3" relType="IS_INCLUDED"/>

4. Ali 2 gün önce ayrıldı.

Ali

<EVENT eid="e1" class="OCCURRENCE">

ayrıldı

</EVENT>

<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PAST" aspect="PERFECTIVE"/>

<TIMEX3 tid="t1" type="DATE" value="2002-07-08" temporalFunction="true" anchorTimeID="t0">

2 gün önce.

</TIMEX3>

<TLINK eventInstanceID="ei1" relatedToTime="t1" relType="IS_INCLUDED"/>

önce'nin sinyal DEĞİL fakat TIMEX3 ifadesinin bir parçası olduğunu not ediniz.

TIMEX3 ifadesi, DCT yada Konuşma süresiyle nispi olan zamansal bir işlevle hesaplanması gereken bir TARİHi iade eder. *2 gün önce* iki olayı birbiriyle ilişkilendiren ve dolayısıyla magnitude (genişlik) niteliğiyle TLINK sunan “2 gün önce” gibi ifadelerin aksine HER ZAMAN, DCT ile nispi olarak hesaplanmış bir TARİHtir. Bu, aşağıdaki 3 örnekte gözlemlenebilir.

5. Ali, saldırıdan 2 gün önce ayrıldı.

Ali

<EVENT eid="e1" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">

ayrıldı

</EVENT>

<MAKEINSTANCE eiid="ei1" eventID="e1"/>

<TIMEX3 tid="t1" type="DURATION" value="P2D" temporalFunction="false">

2 gün

</TIMEX3>

<SIGNAL sid="s1">

önce

</SIGNAL>

the

<EVENT eid="e2" class="OCCURRENCE" tense="NONE" aspect="NONE">

saldırı

</EVENT>

<MAKEINSTANCE eiid="ei2" eventID="e2"/>

<TLINK eventInstanceID="ei1" signalID="s1" relatedToEvent="ei2"

relType="BEFORE" magnitude="t1"/>

6. Geri gelişinden 5 gün sonra Mary hastalandı.

<TIMEX3 tid="t1" type="DURATION" value="P5D" temporalFunction="false">

5 gün

</TIMEX3>

<SIGNAL sid="s1">

sonra

</SIGNAL>

o

<EVENT eid="e1" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">

geldi

</EVENT>

<MAKEINSTANCE eiid="ei1" eventID="e1"/>

geri Mary

<EVENT eid="e2" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">

-landı

</EVENT>

<MAKEINSTANCE eiid="ei2" eventID="e2"/>

Hasta.

```
<TLINK eventInstanceID="ei1" signalID="s1" relatedToEvent="ei2"
relType="BEFORE" magnitude="t1"/>
```

7.Saldırıdan iki ay önce, bir rapor gönderildi.

```
<TIMEX3 tid="t1" type="DURATION" value="P2M" temporalFunction="false">
```

İki ay

```
</TIMEX3>
```

```
<SIGNAL sid="s1">
```

önce

```
</SIGNAL>
```

the

```
<EVENT eid="e1" class="OCCURRENCE" tense="NONE" aspect="NONE">
```

saldırı

```
</EVENT>
```

```
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
```

bir rapor

```
<EVENT eid="e2" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
```

gönderildi

```
</EVENT>
```

```
<MAKEINSTANCE eiid="ei2" eventID="e2"/>
```

```
<TLINK eventInstanceID="ei1" signalID="s1" relatedToEvent="ei2"
```

```
relType="AFTER" magnitude="t1"/>
```

Buradaki TIMEX3, type=DURATION olarak kabul edilmektedir, zira 2 olayı ayıran aralığın uzunluğunu oluşturmaktadır. Böylece, değer niteliğinin değeri zaten

bilinmektedir (P2D, P5M, v.s.) ve bu nedenle temporalFunction (zamansal İşlev) niteliği, değeri olarak hatalı döner.

İki olayı birbirine bağlayan, hem (TIMEX3 ifadesinin kimliğine işaret eden) genişlik niteliğini hem de signalID (sinyal kimliği) niteliğini sunan sadece tek bir TLINK bulunmaktadır.

6.12 Karmaşık TLINK Ve SLINK Örnekleri

1. Her ne kadar 2 ay önce bir rapor gönderilmiş olsa da saldırı hiç beklenmiyordu.

The

```
<EVENT eid="e1" class="OCCURRENCE" tense="NONE" aspect="NONE">
```

saldırı

```
</EVENT>
```

```
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
```

idi

```
<SIGNAL sid="s1">
```

-miyordu

```
</SIGNAL>
```

```
<EVENT eid="e2" class="I_STATE" tense="PAST" aspect="PERFECTIVE">
```

beklen

```
</EVENT>
```

```
<MAKEINSTANCE eiid="ei2" eventID="e2"/>
```

hiç, her ne kadar bir rapor

```
<EVENT eid=e3>
```

gönderilmiş

```
</EVENT>
```

```
<MAKEINSTANCE eiid="ei3" eventID="e3"/>
```

```
<TIMEX3 tid="t1" type="DURATION" val="P2M" temporalFunction="false">
```

2 ay

```
</TIMEX3>
```

<SIGNAL sid=s2>

önce

</SIGNAL>

<SLINK eventInstanceID="ei2" signalID="s1" relType="NEGATIVE"/>

<SLINK eventInstanceID="ei2" subordinatedEvent="e1" relType="MODAL"/>

<TLINK eventInstanceID="ei1" relatedToEvent="ei3" relType="AFTER"
magnitude="t1" signalID="s2"/>

2. Mary dün vardı, ancak Ali 2 gün önce ayrıldı.

Mary

<EVENT eid="e1" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">

vardı

</EVENT>

<MAKEINSTANCE eiid="ei1" eventID="e1"/>

<TIMEX3 tid="t1" type="DATE" value="2002-07-09"

temporalFunction="true" anchorTimeID="t0"

valueFromFunction="tf1">

dün

</TIMEX3>

fakat Ali

<EVENT eid="e2" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">

ayrıldı

</EVENT>

<MAKEINSTANCE eiid="ei2" eventID="e2"/>

<TIMEX3 tid="t2" type="DURATION" value="P2D" temporalFunction="false">

2 gün

</TIMEX3>

<SIGNAL sid="s1">

önce.

</SIGNAL>

<TLINK eventInstanceID="ei1" signalID="s1" relatedToEvent="ei2"

relType="AFTER" magnitude="t2"/>

<TLINK eventInstanceID="ei1" relatedToTime="t1" relType="IS_INCLUDED"/>

İki olay, TLINK vasıtasıyla ilişkilidir. Buna ilaveten, tarihle bağlantılı olayı (*vardı*) ve bu tarihi (dün) bağlayan ikinci bir TLINK bulunmaktadır.

3.Oyundan sonra hastaydı.

O, -ydı

<EVENT eid="e1" class="STATE" tense="NONE" aspect="NONE">

hasta

</EVENT>

<MAKEINSTANCE eiid="ei1" eventID="e1"/>

<SIGNAL sid="s1">

sonra

</SIGNAL>

the

<EVENT eid="e2" class="OCCURRENCE" tense="NONE" aspect="NONE">

oyundan

</EVENT>

<MAKEINSTANCE eiid="ei2" eventID="e2"/>

```
<TLINK eventInstanceID="ei1" signalID="s1" relatedToEvent="ei2"
relType="AFTER"/>
```

TLINK genişliği sunmaz.

4.Oyundan sonra 2 saat hastaydı.

O, -ydı

```
<EVENT eid="e1" class="STATE" tense="NONE" aspect="NONE">
```

hasta

```
</EVENT>
```

```
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
```

```
<SIGNAL sid="s1">
```

boyunca

```
</SIGNAL>
```

```
<TIMEX3 tid="t1" type="DURATION" value="P2H" temporalFunction="false">
```

2 saat

```
</TIMEX3>
```

```
<SIGNAL sid="s2">
```

sonra

```
</SIGNAL>
```

the

```
<EVENT eid="e2" class="OCCURRENCE" tense="NONE" aspect="NONE">
```

oyun

```
</EVENT>
```

```
<MAKEINSTANCE eiid="ei2" eventID="e2"/>
```

```
<TLINK eventInstanceID="ei1" signalID="s1" relatedToTime="t1"
```

```
relType="HOLDS"/>
```

```
<TLINK eventInstanceID="ei1" signalID="s2" relatedToEvent="ei2"
```

```
relType="AFTER"/>
```

İki TLINK vardır: Birincisi, hasta olmakla bunun aldığı zaman (2 saat) arasındaki devam eden ilişkiyi ortaya koymaktadır. İkincisi, iki olayın sırasını belirtmektedir. İki olay arasındaki aralığın süresine hiçbir atıfta bulunulmadığı için, ikinci TLINK, genişlik niteliğini sunmamaktadır.

5. Ali, her Perşembe 20 dakika ders verdi.

Bir OLAY, iki TIMEX3 ve iki SIGNAL (sinyal)in oluşturulması gerekmektedir. Buna ilaveten, aşağıdaki etiketlere gereksinim vardır:

- (a) *her* ifadesiyle sinyali verildiği şekilde *her* kardinalliğini taşıyan bir MAKEINSTANCE.
- (b) *20 dakika* süresini olaya bağlayan bir TLINK
- (c) TIMEX3 *Pazartesi*'ni TIMEX3 *20 dakikasına* bağlayan bir TLINK

Ali

```
<EVENT eid="e4" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
```

```
ders verdi </EVENT>
```

```
<SIGNAL sid="s5">for</SIGNAL>
```

```
<TIMEX3 tid="t2" type="DURATION" value="PT20M" anchorTimeID="3">
```

```
20 dakika</TIMEX3>
```

```
<SIGNAL sid="s6">every</SIGNAL>
```

```
<TIMEX3 tid="t3" type="DATE" value="XXXX-WXX-1">Monday</TIMEX3>
```

```
<MAKEINSTANCE eiid="ei4" eventID="e4" signalID="s6" cardinality="EVERY"/>
```

```
<TLINK eventInstanceID="ei4" relatedToTime="t2" signalID="s5"
```

```
relType="HOLDS"/>
```

```
<TLINK timeID="2" relatedToTime="t3" relType="IS_INCLUDED"/>
```

6. Ali Pazartesi ile Çarşamba arasında ayrıldı

Ali

```
<EVENT eid="e1" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
```

ayrıldı

```
</EVENT>
```

```
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
```

```
<SIGNAL sid="s1"/>
```

arasında

```
</SIGNAL>
```

```
<TIMEX3 tid="t1" type="DATE" value="2002-07-15"
```

```
temporalFunction="true" anchorTimeID="t0"
```

```
valueFromFunction="tf3"/>
```

Pazartesi

```
</TIMEX3>
```

ve

```
<TIMEX3 tid="t2" type="DATE" value="2002-07-17"
```

```
temporalFunction="true" anchorTimeID="t0"
```

```
valueFromFunction="tf3"/>
```

Çarşamba

```
</TIMEX3>
```

```
<TLINK eventInstanceID="ei1" relatedToTime="t1" signalID="s1"
relType="IAFTER"/>
```

```
<TLINK eventInstanceID="ei1" relatedToTime="t2" signalID="s1"
relType="IBEFORER"/>
```

Bu mevcut çözüm, tamamıyla yeterli değildir, fakat geçici olarak koruyacağız.

7. Ali, 1994'ten 1999'a kadar ders verdi.

Bu durumda, bir OLAY ve iki TIMEX3'ün oluşturulması gerekiyor. Buna ilaveten, aşağıdaki etiketlere gereksinim vardır:

- (a) Olay için otomatik olarak oluşturulan bir MAKEINSTANCE.
- (b) Olayın 1994'te başladığı keyfiyetini yakalamak için bir TLINK.
- (c) Olayın 1999'da bittiği gerçeğini yakalamak için bir TLINK.

Bu, sonrasında aletin kapama bölümüyle otomatik olarak oluşturulan bir süreye yol açacaktır.

Ali

```
<EVENT eid="e4" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
ders verdi</EVENT>
```

```
<SIGNAL sid="s5">from</SIGNAL>
```

```
<TIMEX3 tid="t2" type="DATE" value="1994">1994</TIMEX3>
```

```
<SIGNAL sid="s6">through</SIGNAL>
```

```
<TIMEX3 tid="t3" type="DATE" value="1999">1999</TIMEX3>
```

```
<MAKEINSTANCE eiid="ei4" eventID="e4"/>
```

```
<TLINK eventInstanceID="ei4" relatedToTime="t2" signalID="s5"
relType="BEGUN_BY"/>
```

```
<TLINK eventInstanceID="ei4" relatedToTime="t3" signalID="s6"
relType="ENDED_BY"/>
```

8. Ali, Pazartesi değil fakat Salı günü ayrıldı.

Bir OLAY, iki olay örneği, ve üç SİNYALin oluşturulması gerekiyor:

```
Ali -dı<SIGNAL sid="s5">not</SIGNAL>
```

```
<EVENT eid="e4" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
```

ayrıl-</EVENT>

<SIGNAL sid="s6">on</SIGNAL>

<TIMEX3 tid="t3" type="DATE" value="XXXX-WXX-1">Monday</TIMEX3>

fakat

<SIGNAL sid="s7">on</SIGNAL>

<TIMEX3 tid="t4" type="DATE" value="XXXX-WXX-2">Tuesday</TIMEX3>

<MAKEINSTANCE eiid="ei1" eventID="e4"/>

<MAKEINSTANCE eiid="ei2" eventID="e4"/>

<SLINK subordinatedEventInstance="ei1" signalID="s5"

relType="NEGATIVE" polarity="false"/>

<TLINK eventInstanceID="ei1" relatedToTime="t3" signalID="s6"

relType="IS_INCLUDED"/>

<TLINK eventInstanceID="ei2" relatedToTime="t4" signalID="s7"

relType="IS_INCLUDED"/>

6.13 Nedensel Örnekler

1. Yağmurlar, sele yol açtı.

<EVENT eid="e1" class="OCCURRENCE">

yağmurlar

</EVENT>

<EVENT eid="e2" class="OCCURRENCE">

yol açtı

</EVENT>

the

<EVENT eid="e3" class="OCCURRENCE">

sele

</EVENT>

<TLINK eventInstanceID=e1 relatedtoEvent=e3 relType="BEFORE" />

<TLINK eventInstanceID=e1 relatedtoEvent=e2 relType="IDENTITY" />

2. Ali yangına yol açtı.

Ali

<EVENT eid="e1" class="OCCURRENCE">

yol açtı

</EVENT>

<EVENT eid="e2" class="OCCURRENCE">

yangına

</EVENT>

<TLINK eventInstanceID=e1 relatedtoEvent=e2 relType="BEFORE" />

3. Kissinger, büyük bir maliyetle barışı güvence altına aldı.

Kissinger

<EVENT eid="e1" class="OCCURRENCE">

güvence altına aldı

</EVENT>

<EVENT eid="e2" class="OCCURRENCE">

barışı

</EVENT>

<TLINK eventInstanceID=e1 relatedtoEvent=e2 relType="BEFORE" />

büyük bir maliyetle

4. Topa vurdu ve havaya fırladı.

Nedensel olarak hareket eden söylem ilişkileri, halen nedenseller olarak ele alamıyoruz, fakat metinde bulunan zamansal ilişkiyi kesinlikle anlatabiliriz.

O

```
<EVENT eid="e1" class="OCCURRENCE">
```

vurdu

```
</EVENT>
```

topa

ve

o

```
<EVENT eid="e2" class="OCCURRENCE">
```

yükseldi

```
</EVENT>
```

havaya

```
<TLINK eventInstanceID=e1 relatedtoEvent=e2 relType="BEFORE" />
```

7. TÜRKÇE İÇİN ZAMANSAL MANTIK VE TIMEML

Bu bölümün amacı, Türk dilinde kullanılan zamansal mantığı çıkarmak ve bu bölüme kadar anlatılan zamansal mantıklar ile olan ilişkisini, benzerliklerini ve farklılıklarını ortaya koymaktır. Ayrıca bu tezin bir özelliği, bilindiği kadarıyla, zamansal mantıkların doğal dil üzerinde uygulanması açısından 1947 yılındaki Reichenbach zamansal mantık çalışmasından beri, yapılan ilk çalışma olmasıdır. Dolayısıyla doğal dil işleme çalışmaları sırasında kullanılan zamansal modelleme ihtiyaçları ve yukarıda sıralanan bu modellerin eksikleri de bu bölümde ele alınacaktır.

Bu çalışmanın bir amacı da, bilgisayarlar tarafından işlenebilir bir dil çıkarımıdır. Bu amaçla mantıksal seviyede yapılan çalışmalar, bilgisayarlar tarafından işlenebilir bir hale getirilmiş ve TimeML dili üzerinde değişiklik halinde uygulanmıştır.

Bu bölümde, TimeML seviyelerine benzer olarak, çalışma 2 seviyede anlatılacaktır.

Alt seviyede Reichenbach zamansal mantığını kullanan TimeML, <event> etiketleri için bu zamansal mantıkta bulunan operatörleri kullanır.

Üst seviyede ise olayları bağlarken kullanılan mantık Allen Zamansal Mantığı olup, bu mantıkta bulunan olay ilişkileri arasında tanımlı olan operatörleri kullanır.

Ne yazık ki bu iki mantık da Türkçe doğal dil işlemesi sırasında karşılaşılan problemleri çözememektedir.

Bu durumlar ileriki bölümlerde incelenecek ve örnekler üzerinden açıklanacaktır. Ardından problemlere çözüm olarak doktora tezi kapsamında sunulan öneriler anlatılacak ve çözüm yollarının hayata geçirilmesi açıklanacaktır.

Bu bölümde zamansal mantıklar ve TimeML üzerinde karşılaşılan durumları ve ihtiyaç duyulan değişiklikleri gösterdikten sonra bir sonraki bölümde bu yapılan değişikliklerin bir derlem (corpus) üzerinde başarısı tartışılacaktır.

7.1 Timeml Neleri Yapamaz

TimeML dili temel olarak zamansal mantıkların bir derlemi niteliğinde olan Allen zamansal mantığı ve Reichenbach zamansal mantığının üzerine kuruludur. Bu durumda bu zamansal mantıkların yetenekleri ile sınırlıdır.

7.1.1 Terslik ve Tekrarlılık ilişkisi

Terslik ve Tekrarlılık ilişkisi, TimeML tarafından yapılamayan ancak Türkçede rastlanan bir durumdur.

Örneğin 3. ve 4. bölümlerde anlatılan zamansal mantıkların hepsinde zamansal olarak bir doğrusallık söz konusudur. Bunun anlamı zaman çizgisi üzerinde bir geçmiş ve gelecek yönünde zamansal olarak olayların sıralı olmasıdır.

TimeML dilinde de barındırılan bu özellik aslında Türkçe için doğru olan bir durumdur.

Ancak Türkçede çeşitli durumlarda yaşanmakla birlikte diğer dillerde de nadiren de olsa yaşanan olayların tekrar edilmesi problemi bulunmaktadır.

Aslında zamanın tekrar etmesi söz konusu değildir ama zamanı modelleyen doğal dildeki cümlelerin olayları tekrarlı göstermesi mümkündür. Bu anlamda doğal dil ile iyi şekillendirilmiş (WFF) bir dilin farkı ortaya çıkar. Örneğin, Türkçede “Arabanın sinyali yanıp sönüyordu” cümlesinden “yanıp sönme” eyleminin tekrar ettiği anlaşılmaktadır. Bu süreklilik hali dışında anlaşılan bir diğer anlam ise arabanın sinyalinin yandığı sonra söndüğü sonra tekrar yandığı ve tekrar söndüğüdür.

O halde yanma ve sönme olayları için ayrı ayrı tekrarlanma durumu söz konusudur.

Bu doğal dilde mümkün olan ifade ne yazık ki herhangi bir zamansal mantıkta gösterimi mümkün olmayan bir durum ortaya çıkarır.

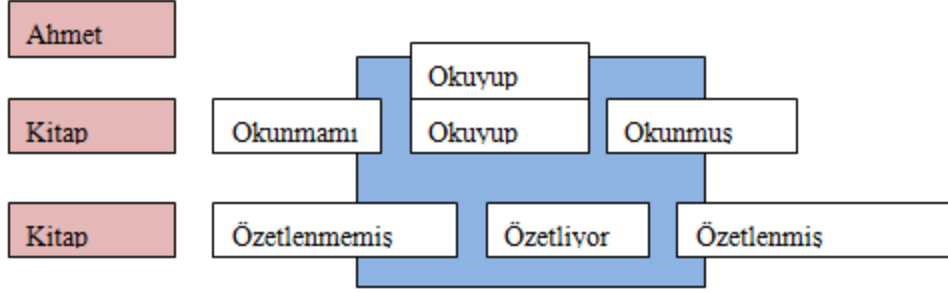
Belirli (deterministic) ve doğrusal yaklaşım uygulanan dillerde bu durum gerçekleştikçe kaydedilebilir.

Yani arabanın sinyalinin her yanması ve her sönmesi belirli bir şekilde bu zamansal süreçte gösterilerek problem çözülebilir. Ancak sürekliliği olan bu eylem ne yazık ki modellenemez.

Özellikle İngilizce’nin etkisi altında kalan yukarıdaki model dillerde ne yazık ki böyle bir ifade mümkün olmamaktadır. Bunun sebebi İngilizce gibi Latin kökenli dillerde fiillerin tekrar veya tersleri ile süreklilik ifadesinin mümkün olmamasıdır. Örneğin, İngilizce için “Yol boyunca gidip geliyordu” şeklinde bir cümlenin tercümesi “Walking up and down the street” şeklinde yapılabilir.

Burada dikkat edilecek fark, Türkçede fiil ile gösterilen tersliğin, İngilizcede edatlar ile gösterilmesidir.

“Ahmet kitapları okuyup okuyup özetliyordu. Ahmet hastaneye gidip gidip Ayşe’yi görüyordu.
 “Ahmet kitapları okuyup okuyup özetliyordu.”



Şekil 7.1 Örnek tekrarlı cümle modellemesi

Ayrıca yukarıdaki zamansal mantıklardan esinlenerek oluşturulan TimeML dilinde de bu durumu ifade etmek için *Included* özelliği bulunur. Yani bir olayın farklı bir olay tarafından içerilmesi, TimeML tarafından gösterilmesi mümkün bir durumdur.

Ancak ne yazık ki hangi olayın hangi diğer olayı içereceği belirsiz olmakla birlikte bu iki olay aslında aynı seviyeden olup, birisi diğerinden önce veya sonra başlamıştır denilemez.

7.1.2 Kendine atıf durumu

Bu durum zamansal mantıklar için tam bir kaos oluşturmaktadır. Özellikle makineler tarafından ifade edilmeye çalışılan ve çoğu zaman basitleştirmek ve modellemek amacıyla kullanılan zamansal mantıklarda karşılaşılmayan bu durum, doğal dilde görülebilmektedir.

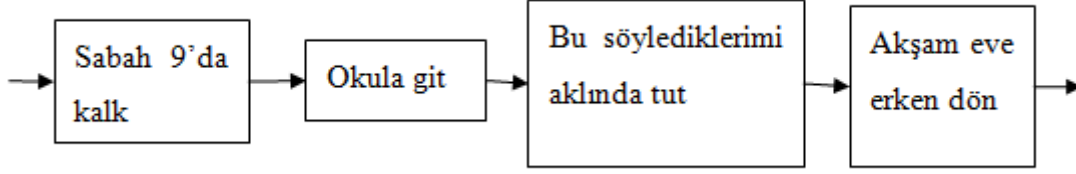
Örneğin bir konuşma esnasında, konuşmanın kendisi hakkında konuşulabilir.

Benzer şekilde bu dokümanı yazarken bu dokümanın kendisi hakkında bir şeyler yazmak mümkündür.

Örneğin, Ali’ye aşağıdaki emirler verilmiş olsun:

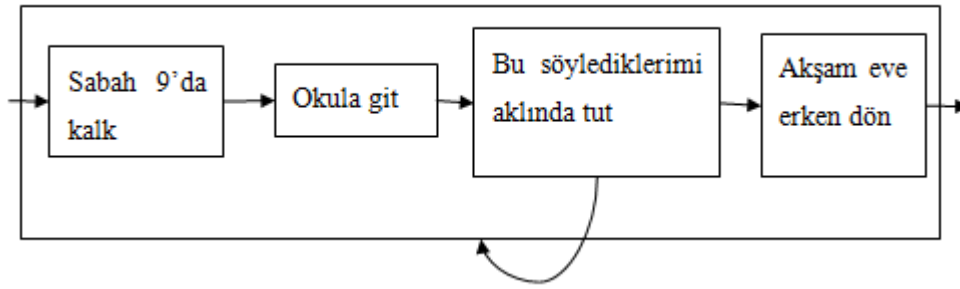
- Sabah 9’da kalk
- Okula git
- Bu söylediklerimi aklında tut
- Akşam eve erken dön

Yukarıda sıralanan bu emirleri görsel olarak gösterirsek aşağıdaki gibi bir şekil ortaya çıkar:



Şekil 7.2 Örnek emir sıralaması

Şekil 7.2’de gösterilen bu sıralama emirlerin veriliş ve yapılış sırasını temsil etmektedir. Bu örnekte emirlerin yapılmasının beklendiği sıra ile emirlerin veriliş sırası basitleştirmek açısından aynı verilmiştir. Bu sıra değiştirilerek daha karmaşık örnekler üretilebilir. Ancak yukarıdaki örnekte özel olarak problemlili olan bir durum vardır. Üçüncü emir, bu emirler sırasının tamamına atıfta bulunmaktadır.



Şekil 7.3 Kendine atıfta bulunan örnek olay sıralaması

Değiştirilerek çizilen yeni Şekil 7.3’de görüldüğü üzere atıfta bulunulan olay, atıfta bulunma olayını da kapsamaktadır.

Bu durumda atıfta bulunma olayı ile bu emirler zinciri arasındaki ilişki ne yazık ki mevcut zamanlarda gösterilemez. Çünkü, bu atıfta bulunma eylemi, emirler zincirinde önce denilirse hatadır, sonra denilirse yine hatalıdır, içerir denilirse hatalıdır ve içerilir denilirse yine hatalı olur.

7.1.3 Zamanda tekrar durumu

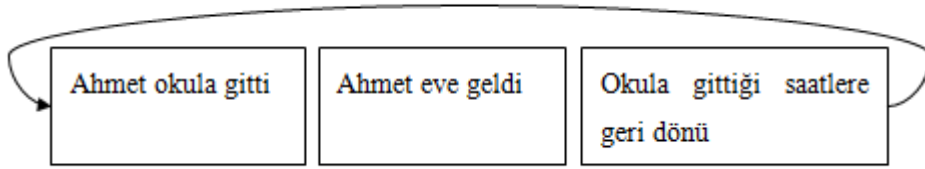
Doğal metinlerde karşılaşılan bir problem de bu metinlerde zamanla ilgili her şeyin özgürce anlatılabilmesidir. Belki bir makine veya zaman programlama sisteminde bu tip bağlayıcı şartlar bulunmaz ancak doğal dilde anlatılabilecekler, insanların hayal güçleri ve dilin ifade yeteneği ile sınırlıdır.

Zamanda yolculuk yapılmasının anlatıldığı bir doğal dil metninde zaman üzerinde bir döngü oluşturulması söz konusudur.

Ayrıca reenkarnasyon gibi kavramlar doğal dilde olan metinlerde geçebilmektedir.

Örneğin aşağıdaki cümleleri ele alalım:

- Ahmet okula gitti.
- Sonra Ahmet eve döndü
- Sonra Ahmet, okula gittiği saatlere geri döndü.



Şekil 7.4 Zamanda geri dönülmesi durumu

Görüldüğü üzere bazı metinlerde ve doğal dil ifadelerinde Şekil 7.4'e benzer bir döngü oluşabilmektedir.

Kısacası zamanda yolculuk veya reenkarnasyon gibi zamanın tekrar etmesini ifade eden kavramlar, ne yazık ki zamansal mantıklarla modellenemez. Zamansal mantıklarda bu tip ifadeleri modelleyen operatörler yoktur.

Burada ifade problemi aynı zamanda mantıksal olarak mantığın ispatlanabilir ve bilgisayar tarafından işlenebilir olması koşuluyla birleşince daha büyük bir probleme dönüşmektedir. Zaten özellikle işlenebilir zamansal mantıklarda bu gösterimlerin bulunmamasının sebebi de burada yaşanan çözümleme problemidir.

7.1.4 Türkçedeki zamanlar ve fiil yapılar

Doğal dillerdeki zamanların modellenmesi sırasında, yukarıdaki problemlere ilave olarak Türkçe dilinde bazı özel durumlar bulunmaktadır. Bu durumlar özellikle işlenebilir zamansal mantık modellerinde Latin kökenli dillerden esinlenildiği için bugüne kadar göz ardı edilmiştir.

Örneğin en gelişmiş işlenebilir dil örneklerinden olan TimeML dilinin olaylar arasındaki

ilişkileri anlatırken kullandığı 13 farklı ilişki modeli Reichenbach modeline dayanmaktadır. Bu modelde Türkçe için karşılaştırmalı Çizelge 7-1’de verilmiştir.

Çizelge 7.1 Reichenbach zamansal mantığı ile Türkçe karşılaştırılması

İhtimaller	Reichenbach Zaman İsmi	Türkçe Zamanları	Örnek
E<R<S	Anterior past	Geçmişin hikayesi	Uyumuştum
E=R<S	Simple past	Geçmiş	Uyudum
R<E<S		Gelecek hikayesi	Uyuyacaktı
R<S=E	Posterior past		
R<S<E		Gelecek Rivayeti	Uyuyacakmış
E<S= R	Anterior present	Şimdiki Hikayesi	Uyumuşum
S= R= E	Simple present	Şimdiki	Uyuyorum
S= R<E	Posterior present	Gelecek	Uyuyacağım
S<E<R			
S=E<R	Anterior future		Uyuyor olacağım
E<S<R			Uyumuş olacağım
S<R=E	Simple future	Gelecek Zaman	Uyuyacağım
S<R<E	Posterior future		Uyuyacak olacağım

Çizelge 7-1’de görüldüğü üzere, bazı zamanlar, Türkçede bulunmamakta bazıları ise Türkçede bulunup İngilizcede bulunmamaktadır. Bu fark, Türkçenin İngilizce ve diğer Latin kökenli dillerden farklı bir zamansal mantığa sahip olduğunu gösterir.

Çizelge 7-1’in tamamı, TimeML tarafından ifade edilebilmektedir. Ancak problem, Çizelge 7-

1’de gösterilemeyen bazı durumların da Türkçede bulunmasından kaynaklanmaktadır.

Örneğin Türkçede bulunan rivayet geçmiş zaman ve hikaye geçmiş zaman arasındaki fark ifade edilmemektedir.

Rivayet ve hikaye geçmiş zamanlar arasında kişinin ilgili olayı yaşayıp yaşamaması veya diğer bir ifadeyle o olayı bilip bilmemesi vurgulanmaktadır.

Örneğin, “ Dün, Ali okula geldi” cümlesi ile “ Dün, Ali okula gelmiş” cümlesi arasında Reichenbach zamansal mantığına göre bir fark bulunmamaktadır. Bu durum Reichenbach modelinde ifade edilen 3 özellik hatırlanırsa:

1. olayın geçtiği zaman (E)
2. olayın yazıldığı veya anlatıldığı zaman (S)
3. referans zamanı (R)

Aşağıdaki şekilde ifade edilebilir.

((Dün) R (Alinin okula gelmesi)E) S

Örnekte referans zamanı “Dün”, olay zamanı “Ali’nin okula geldiği zaman” ve anlatılma zamanı bu iki olaydan sonradır.

Bu modelde R ve E için aynı zamanda kabulü yapılabilir. Yani “Dün” kavramı bir aralık belirtmekte ve “Ali’nin gelmesi” olayı bu aralıkta herhangi bir zamanda gerçekleşmektedir.

Bu örnek için $R=E$ olduğu söylenebilir.

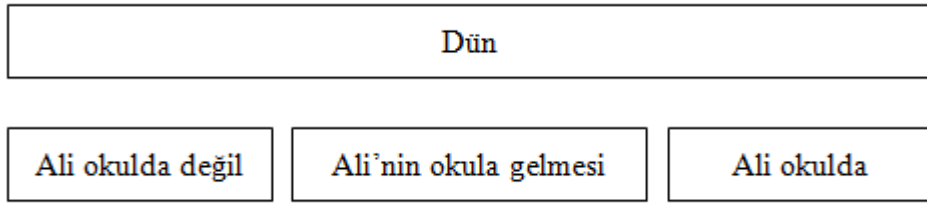
Ayrıca olayın anlatılması veya yazılması sırasında geçmiş zaman kullanılmasından veya Dün kelimesinin geçmişte bir tarihi ifade etmesinden $S > R$ veya $S > E$ sonuçları çıkarılabilir. Bu durumda yukarıdaki cümle için aşağıdaki Reichenbach modeli uygundur:

$R = E < S$

Yukarıdaki iki farklı cümle de aynı Reichenbach zamanı ile gösterilmekte olup, Türkçede farklı zamanlar olan bu iki cümle için Reichenbach modelinde bir farklılık bulunmamaktadır.

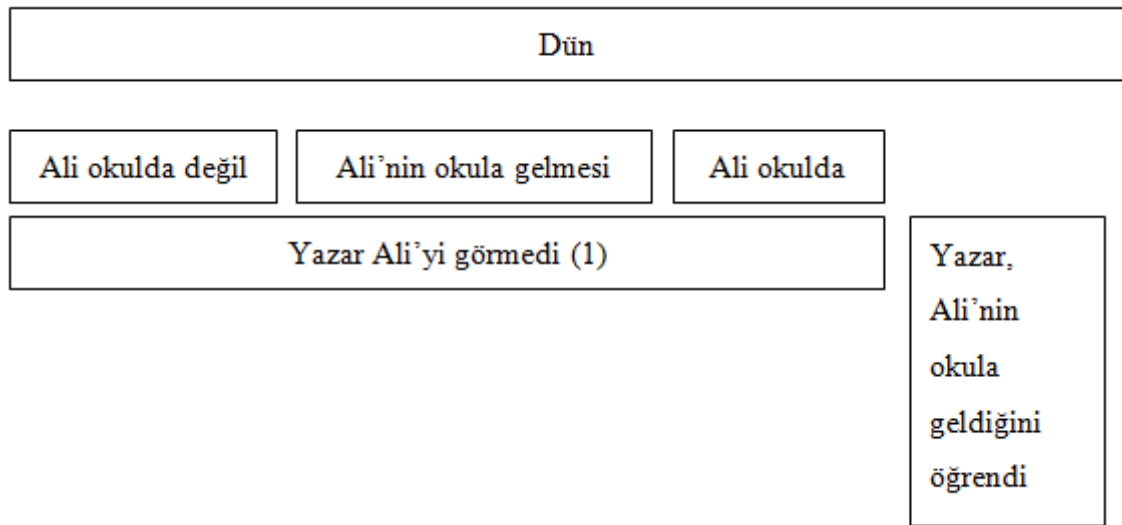
Yukarıda verilen bu örnek cümle için durum ifade edilmek istenirse iki cümle için ayrı ayrı aşağıdaki şekiller çizilebilir:

Öncelikle “Dün, Ali’nin okula gelmesi” şeklinde iki cümlede de ortak olan ifade Şekil 7.5’de gösterilmiştir.



Şekil 7.5 Örnek zaman olay ilişkisi

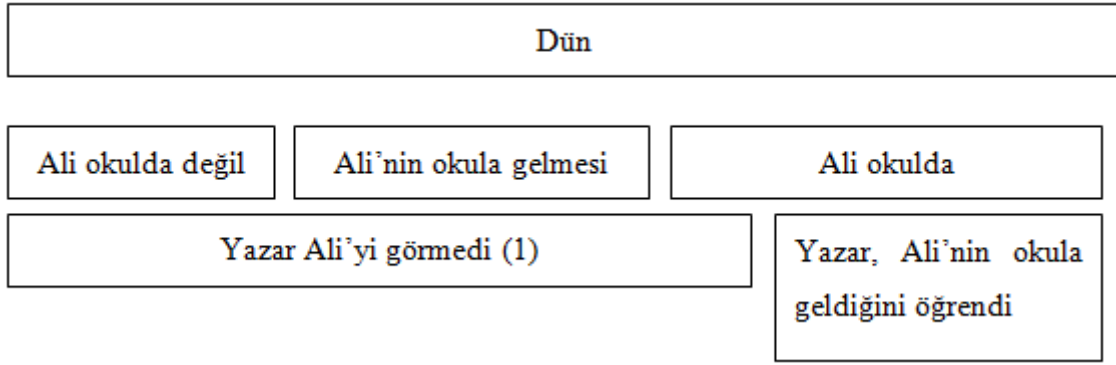
Bu modele Ali'nin okula geldiğinin bilinmesi veya görülmesi ifadesini içeren rivayet geçmiş zaman çekimini eklendiğinde Şekil 7.6 çizilebilir.



Şekil 7.6 Örnek zaman olay ve yazar rivayet ilişkisi

Yukarıdaki modelde rivayet geçmiş zamanın ifade ettiği durum olarak, yazarın Ali'nin okula gelmesi ve okulda olmasından sonra, herhangi bir zamanda bu eylemi öğrenmesi gösterilmiştir.

Türkçe açısından yukarıdaki durum doğru olduğu gibi aşağıdaki durum da doğrudur.

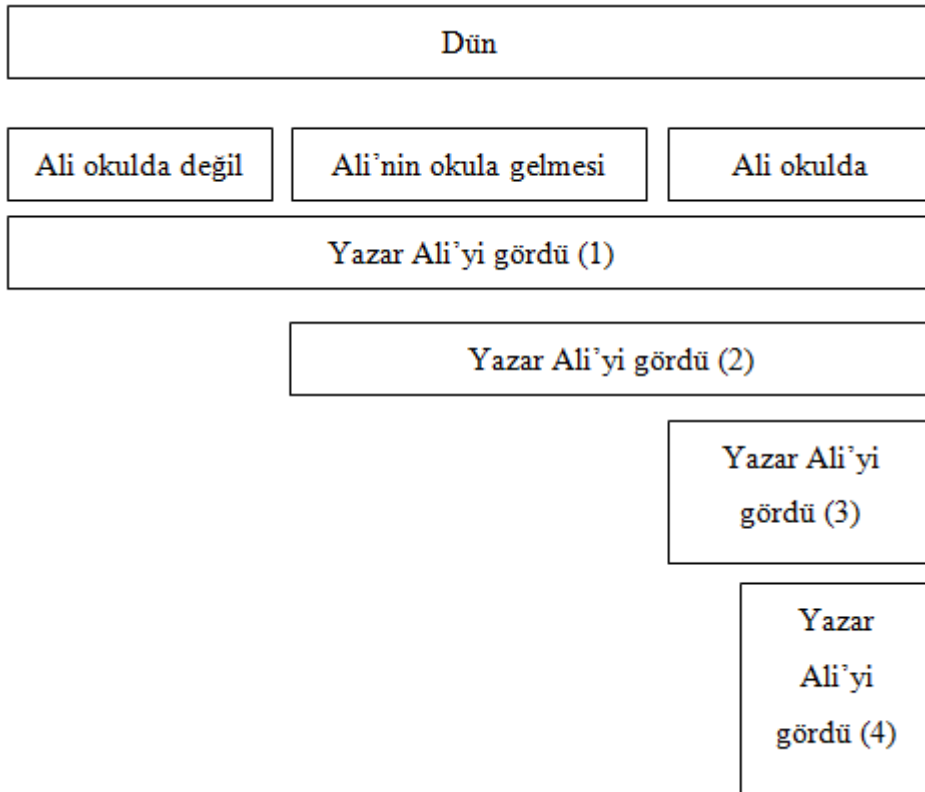


Şekil 7.7 Örnek zaman olay ve yazar rivayet ilişkisi

Şekil 7.7'deki modelde, bir öncekinden farklı olarak, Ali'nin okula gelmesi eyleminden sonra, henüz Ali okuldayken yazar durumu öğrenmiştir.

Rivayet geçmiş zamanda, yukarıdaki bu iki durum da çıkarılabilir.

Hikaye geçmiş zamanda çıkarılabilecek durumlar ise Şekil 7.8'de verilmiştir.



Şekil 7.8 Örnek zaman olay ve yazar rivayet ilişkisi

Şekil 7.8'deki 6 farklı ihtimal hikaye geçmiş zaman kullanıldığında çıkarılabilmektedir.

Buna göre:

- (1) Yazar henüz Ali okulda yok iken okulda olup, sırasıyla, Ali'nin okulda olmamasına, Ali'nin okula gelmesine ve Ali'nin okulda olmasına şahitlik etmiştir.
 - (2) Yazar duruma, Ali'nin okula gelmesi eylemi ile birlikte şahitlik etmiş olup, bundan sonra Ali'nin okulda olduğu duruma da şahitlik etmiş olabilir
 - (3) Yazar duruma, Ali'nin gelme eylemi biter bitmez şahitlik etmiş olabilir.
 - (4) Yazar duruma, Alinin gelme eylemi bittikten bir müddet sonra şahitlik etmiş olabilir.
- Bu durumda, yazar Ali'yi okulda görmüş ancak gelişini görmemiştir. Yazara henüz okulda iken durumu anlatması istense rivayet geçmiş zaman kullanacak olduğu halde olay zamanını olan “Dün” etkisinden çıkınca hikayet geçmiş zaman kullanmaktadır.

Yukardaki verilen örnekte yaşanan durumlar ne yazık ki İngilizce gibi Latin kökenli dillerde ifade edilememekte ve dolayısıyla TimeML dilinin de temelini oluşturan Reichenbach zaman modelinde bu fark bulunmamaktadır.

Bu durum sonucunda Türkçe’de bulunan önemli bir ifade farkı zamansal modelde ortadan kalkmaktadır.

7.2 Yaprak Seviyesi İyileştirmeler

TimeML dilinde bulunan olay seviyesi, Reichenbach zamansal mantığı kullanılarak geliştirilmiştir. Bu zamansal mantıkta ne yazık ki Türkçede bulunan Hikaye-Rivayet geçmiş zaman arasındaki fark belirtilmez.

Bu farkın açıklanması açısından TimeML içerisinde bulunan <EVENT> etiketine ilave bir özellik konulmuştur.

TimeML başlıklı bölümde anlatıldığı üzere <EVENT> etiketinin “tense” ismindeki özelliği geçmiş zamanları ifade ederken sadece PAST şeklinde bir değer alabilmektedir. Ancak Türkçede bu geçmiş zaman durumu iki farklı şekilde gösterilebilmektedir.

Mevcut TimeML uygulamaları ile bir çelişkiye sebep olunma kaygısından dolayı “tense” özelliği ve bu özelliğin bir değeri olan PAST aynen bırakılmış. Rivayet geçmiş zamanı için MIŞLI şeklinde yeni bir değer alternatifi TimeML üzerine eklenmiştir.

Bu sayede doğal dildeki bir metin etiketlenirken bu fark belirtilebilecektir.

Çözüm olarak TimeML dilinin üzerine kurulu olduğu Reichenbach zamansal mantığında değişiklik yapıp Çizelge 7-2'deki şekilde ekleme yapılmasını öneriyoruz.

Çizelge7.2 Reichenbach Zamansal Mantığının Türkçeye Uygulanması

İlişki	Örnek
B,E<R<S	Dün Geleceğim dediydi
Ö,E<R<S	Dün Geleceğim demişti
B,E=R<S	Geliyorum dedi
Ö,E=R<S	Geliyorum demiş
B,R<E<S	Geleceğim dedi
Ö,R<E<S	Geleceğim demiş
B,R<S=E	Geldim diyor
Ö,R<S=E	Geldim diyor
B,R<S<E	Geldim diyecek
Ö,R<S<E	Geldim diyecek
B,E<S= R	Şimdi döneceğini söyledi
Ö,E<S= R	Şimdi döneceğini söylemişti
B, S= R= E	Geliyorum diyor
Ö, S= R= E	Geliyorum diyor
B,S= R<E	Şimdiye dönecek
Ö,S= R<E	Şimdiye dönecek
B,S<E<R	Geleceğim diyecek
Ö,S<E<R	Geleceğim diyecek
B,S=E<R	Geleceğim diyor
Ö,S=E<R	Geleceğim diyor
B,E<S<R	Yarın geleceğini söyledi
Ö,E<S<R	Yarın geleceğini söylemişti
B,S<R=E	Geliyorum diyecek
Ö,S<R=E	Geliyorum diyecek
B,S<R<E	Geldim diyecek
Ö,S<R<E	Gelmişim diyecek

Çizelge 7-2’de bulunan B fonksiyonu, bilinen ve Ö fonksiyonu ise öğrenilen durumu ifade etmektedir. Türkçede desteklenmeyen bazı durumlar teorik olarak mümkün olsa da çizelgede gösterilmemiştir. Ayrıca Çizelge 7-2’de görüldüğü üzere bazı mantıksal durumlar Türkçe de aynı anlama gelmektedir. Örneğin “Geliyorum diyor” şeklindeki bir cümlede, gelme eyleminin, anlatan kişi tarafından biliniyor olup olmaması anlatımı değiştirmez. Bu şekilde anlam değişmeyen durumlar elendiğinde, Türkçe için genişletilen Reichenbach Zamansal Mantiğının tablosu 20 alternatif içerir.

Çizelge 7.3 Reichenbach Zamansal Mantiğının Türkçeye Uygun Hali

İlişki	Örnek
$B, E < R < S$	Dün Geleceğim dediydi
$\bar{O}, E < R < S$	<i>Dün Geleceğim demişti</i>
$B, E = R < S$	Geliyorum dedi
$\bar{O}, E = R < S$	<i>Geliyorum demiş</i>
$B, R < E < S$	Geleceğim dedi
$\bar{O}, R < E < S$	<i>Geleceğim demiş</i>
$R < S = E$	Geldim diyor
$R < S < E$	Geldim diyecek
$B, E < S = R$	Şimdi döneceğini söyledi
$\bar{O}, E < S = R$	<i>Şimdi döneceğini söylemişti</i>
$S = R = E$	Geliyorum diyor
$S = R < E$	Şimdiye dönecek
$S < E < R$	Geleceğim diyecek
$S = E < R$	Geleceğim diyor
$B, E < S < R$	Yarın geleceğini söyledi
$\bar{O}, E < S < R$	<i>Yarın geleceğini söylemişti</i>
$S < R = E$	Geliyorum diyecek
$B, S < R < E$	Geldim diyecek
$\bar{O}, S < R < E$	Gelmişim diyecek

Şekil 7.3’de, Şekil 7.2’de bulunan ve Türkçe için anlam farklılığı oluşturmeyen durumlar

elenmiş ve en sade haliyle 20 ihtimale indirilmiştir.

Reichenbach Zamansal Mantığına yapılan bu ilave Türkçe'deki zamansal gösterim zenginliğinden kaynaklanmaktadır.

7.3 İlişki Seviyesi İyileştirmeler

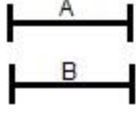
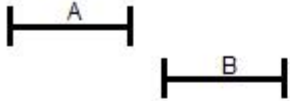
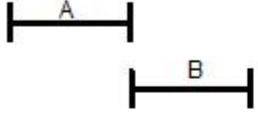
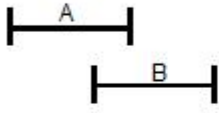
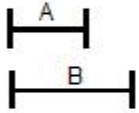
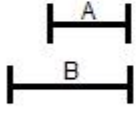
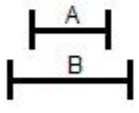
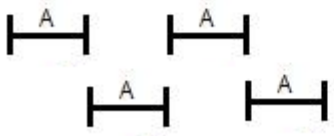
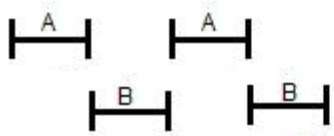
İlişki seviyesi TimeML dilinde Allen zaman mantığı ile modellenmektedir. Bu konuda yaşanan problemler bir önceki alt bölümde ifade edilmiştir. Bu problemlerin çözülmesi için bağlantı modellerinde değişikliğe gidilmiş ve TimeML dilindeki LINK seviyesine yeni eklentiler yapılmıştır.

Bu eklentiler aşağıda verilmiştir:

- Tekrarlı olayların çözümü için TLINK ve SLINK etiketlerine daha önceden bulunmayan RECURRING seçeneği konulmuştur.
- Doğrusallığı bozan ve zamanda hareket edilmesini veya kendine atıfta bulunan ilişkileri göstermek için de TimeML çalışması sırasında doğrusal olmayan çözümleme yeteneği eklenmiştir.

Buna göre yapılan eklentilerden önce TimeML dili doğrusal bir özellik taşıırken, bu eklentilerden sonra hem geçmişe hem de geleceğe doğru doğrusal olmayan daireler (cycle) içerebilmekte ayrıca olayların birbirini sürekli olarak tekrarladığını belirten bir ilave özellik kazandırılmaktadır.

Bu eklentilerden sonra Allen zamansal mantığı aşağıdaki ilave operatörü kazanmış ve Allen zamansal mantığında bulunan ilişkiler Şekil 7.10'da gösterilmiştir:

	A ile B eşit B ile A aynı zamanda
	A, B'den önce B, A'dan sonra
	A'yı B takip eder B, A tarafından izlenir
	A ile B çakışır B, A ile çakışır
	A, B'yi başlatır B, A tarafından başlatılır
	A, B'yi bitirir B, A ile biter
	A esnasında B olur B, A'yı kapsar
	A kendi kendini tekrar eder
	A ile B olayları birbirini tekrar eder

Şekil 7.3 Türkçe için iyileştirilmiş AZM

Şekil 7.10'da eklenen son iki bağlantı şekli, Türkçe doğal dil işleme sırasında karşılaşılan problemler neticesinde ve Türkçe dilbilgisinin TimeML diline adapte edilmesi sırasında

eklenmiştir.

Ayrıca TimeML dilinin doğrusallığı anlamına gelen x ve y gibi iki olayı bağlarken kullanılan

$$\forall xy(x R y \vee x = y \vee y R x)$$

Özellik, aşağıdaki şekilde değiştirilmiştir:

$$\forall xy(x R y \vee x = y \vee y R x \vee y R x \vee y R x \wedge x R y)$$

Bu değişiklikler gerek TimeML diline gerekse TimeML üzerinde çalışan bilgisayar uygulamalarına işlenmiştir.

8. DERLEM ÇALIŞMASI

Bundan önceki bölümlerde anlatılan zamansal mantıkların doğal dilde çözümlenmesi için işlenebilir dil geliştirilmesi ve iyileştirilmesi aşamalarının bir derlem üzerinden başarısının ölçülmesi gerçekleştirilmiştir.

Derlem sırasında 10 adet çocuk hikayesi kullanılmıştır. Bu çocuk hikayeleri şunlardır:

1. Orman perisinin gülleri
2. Obur İnek
3. Bezelye prenses
4. Kelebek
5. Yardımsever adam ve sihirli taş
6. Pinokyo
7. Vefalı aşıklar
8. Obur kaplumbağa
9. Yoksul kunduracı
10. Gül Cenneti

Bu çocuk hikayeleri ile ilgili istatistikler Çizelge 8-1’de sunulmuştur:

Çizelge 8.1 Derlem, Cümle, Kelime ve Fiil Sayıları

Hikaye İsmi	Cümle Sayısı	Kelime Sayısı	Fiil Sayısı	Fiil Oranı
Orman Perisinin Gülleri	36	403	45	%11
Obur İnek	32	197	45	%23
Bezelye Prenses	17	229	44	%19
Kelebek	10	101	22	%22
Yardımsever adam ve sihirli taş	131	901	150	%17
Pinokyo	136	1217	155	%13
Vefalı Aşıklar	109	898	160	%18
Obur Kaplumbağa	43	372	65	%17
Yoksul Kunduracı	31	365	55	%15
Gül Cenneti	35	259	45	%17
	580	4942	786	%16

Çocuk hikalelerinde geçen fiillerin Türkçe zamanlara göre sınıflandırılması Çizelge 8-2’de yapılmıştır.

Çizelge 8.2 Derlem, Türkçe Zamanlara göre fiil sayıları

Hikaye İsmi	Mişli geçmiş zaman	Dili geçmiş zaman	Geniş zaman	Gelecek zaman	Şimdiki zaman
Orman perisinin gülleri	22	0	3	3	3
Obur İnek	3	26	0	0	3
Bezelye prenses	23	5	4	1	3
Kelebek	0	13	2	3	0
Yardımsever adam ve sihirli taş	96	2	15	3	4
Pinokyo	98	6	3	7	8
Vefalı aşıklar	78	6	7	15	7
Obur kaplumbağa	28	1	3	0	3
Yoksul kunduracı	33	0	0	3	0
Gül Cenneti	23	1	5	3	1

Çizelge 8-2’de bulunan sayılardan da anlaşılacağı üzere, örnek derlemimizde en çok kullanılan zaman mişli geçmiş zamandır. Bunun sebebi, derlemin genel olarak hikayelerden oluşması ve hikaye anlatımında en çok mişli geçmiş zamanın kullanılmasıdır.

Ayrıca Çizelge 8-2’deki değişik zamanlarda olan fiillerin yüzdesel dağılımı, Çizelge 8-3’de verilmişti.

Çizelge 8.3 Derlem, Türkçe Zamanlara göre fiil oranları

Hikaye İsmi	Mişli geçmiş zaman	Dili geçmiş zaman	Geniş zaman	Gelecek zaman	Şimdiki zaman
Orman perisinin gülleri	%71	-	%10	%10	%10
Obur İnek	%9	%81	-	-	%9
Bezelye prenses	%64	%14	%11	%3	%8
Kelebek	-	%72	%11	%17	-
Yardımsever adam ve sihirli taş	%80	%2	%13	%3	%3
Pinokyo	%80	%5	%2	%6	%7
Vefalı aşıklar	%69	%5	%6	%13	%6
Obur kaplumbağa	%80	%3	%9	-	%9
Yoksul kunduracı	%92	-	-	%8	-
Gül Cenneti	%70	%3	%15	%9	%3

Hikayelerdeki birleşik cümle yapısı ise Çizelge 8-4’de verilmiştir:

Çizelge 8.4 Derlem, Birleşik Cümle Sayıları

Hikaye İsmi	Birleşik Cümle Sayısı	Birleşik Cümle Oranı
Orman perisinin gülleri	13	36%
Obur İnek	7	22%
Bezelye prenses	2	12%
Kelebek	2	20%
Yardımsever adam ve sihirli taş	22	17%
Pinokyo	14	10%
Vefalı aşıklar	8	7%
Obur kaplumbağa	6	14%
Yoksul kunduracı	1	3%
Gül Cenneti	0	0%

Çizelge 8-3’de görüleceği üzere çocuk hikayelerinde kullanılan birleşik cümle oranı oldukça düşüktür. Bu oran %0 ile %12 arasında değişmekte olup, çoğu hikayede çocukların anlamasını kolaylaştırmak için cümleler basitleştirilmiş veya basit olarak kurgulanmıştır.

Yukarıdaki örneklerde de görüldüğü üzere derlem içerisindeki cümlelerin çoğunluğu TimeML işaretleme dili tarafından zamansal olarak modellenirken problem oluşturur. Bunun sebebi TimeML dilinde mişli geçmiş zamanın ifade problemidir.

TimeML dili ile, derlem üzerinde başarı çalışması yapılırken iki farklı ölçüm yapılmıştır.

- Reichenbach seviyesi
- Allen Zamansal Mantık seviyesi

Bu iki seviyedeki çalışmalarda TimeML üzerinde Türkçe iyileştirilmesi yapılmadan önce ve sonra diye ayırabiliriz.

Çizelge 8.5 Derlem Reichenbach Seviyesi iyileştirme

Hikaye İsmi	İşaretlenebilir <EVENT>	İşaretlenemez <EVENT>	TimeML başarısı	TimeML Eksik
Orman perisinin gülleri	9	57	%14	%86
Obur İnek	3	20	%13	%87
Bezelye prenses	8	31	%21	%79
Kelebek	5	5	%50	%50
Yardımsever adam ve sihirli taş	22	144	%13	%87
Pinokyo	18	138	%12	%88
Vefalı aşıklar	29	114	%20	%80
Obur kaplumbağa	6	55	%10	%90
Yoksul kunduracı	3	49	%6	%94
Gül Cenneti	9	31	%23	%78
Ortalama			%18	%82
Toplam	112	644		

Çizelge 8-5’de, derlem çalışması kapsamındaki hikayelerde bulunan <EVENT> etiketlerinin Reichenbach seviyesine göre işaretlenebilirliği gösterilmiştir. Ne yazık ki Türkçede bulunan

geçmiş zaman ayrımını içermeyen TimeML dili, iyileştirilme yapılmadan önce, derlemde bulunan etiketlerin yalnızca %18'i gibi çok ufak bir kısmını etiketleyebilmektedir.

Yapılan iyileştirme sonucunda işaretlenebilir <EVENT> etiketlerinin durumu Çizelge 8-5'de verilmiştir.

Çizelge 8.6 İyileştirme Sonrası Reichenbach Seviyesi Başarılar

Hikaye İsmi	İşaretlenebilir <EVENT>	İşaretlenemez <EVENT>	Başarı
Orman perisinin gülleri	66	0	%100
Obur İnek	23	0	%100
Bezelye prenses	39	0	%100
Kelebek	10	0	%100
Yardımsever adam ve sihirli taş	166	0	%100
Pinokyo	156	0	%100
Vefalı aşıklar	143	0	%100
Obur kaplumbağa	61	0	%100
Yoksul kunduracı	52	0	%100
Gül Cenneti	40	0	%100

Görüldüğü üzere, yapılan bu iyileştirme ile TimeML dilinin başarısı derlem çalışması üzerine %100 gibi bir sayıya çıkarılmıştır.

Klasik bir makine öğrenme çalışmasında, derlemin eğitim (train) ve test kümeleri olarak ikiye ayrılması ve bu kümelere eğitim kümesi üzerinden sistemin öğrenmesi, ardından test kümesi üzerinden de bu öğrenmenin başarısının ölçülmesi beklenebilir. Bu tez çalışmasında, derlemin eğitim ve test olarak ikiye ayrılmaması, bunun yerine, derlemdeki bütün bilgilerin önce eğitim için taranması ve ardından da derlemin tamamının oluşan sistem tarafından kapsandığının gösterilmesinin iki önemli sebebi bulunmaktadır.

Birincisi, bu tez çalışmasının sonucunda işlenebilir bir dil çıkarılmıştır. Doğal dilden bu işlenebilir dile dönüşüm ise, 9. Bölümde anlatıldığı üzere, otomatik olarak yapılamamaktadır. Bu dönüşümün, elle yapılması, derlem çalışmasının, ikiye ayrılarak test kümesi üzerinde başarı sınamasını imkansız hale getirmektedir. Çünkü test kümesi üzerindeki başarının

ölçümü, zaten elle yapılan sonuçlara yakınlığa göre belirlenecektir. Oysaki, çalışma zaten elle yapılmaktadır.

İkincisi, Türkçede bulunan bütün zamansal olay ilişkilerinin tez kapsamında incelenmesi ve TimeML dili üzerinde genişletilmesi hedeflenmiştir. Derlem çalışması sırasında %100'den daha düşük bir sonuç bulunması, ancak TimeML üzerinde Türkçe için geçerli bir zamansal ifadenin bulunmaması durumunda olabilir. Bu durumda ise tez çalışmasında başarısızlık olduğundan bahsedilebilir. Diğer bir deyişle, derlem üzerinde, TimeML tarafından işlenemeyen bütün durumlar ele alınmış ve TimeML dilinin kapsamı sağlanmıştır.

9. TIMEML ÜZERİNDE UYGULAMA GELİŞTİRİLMESİ

TimeML dili tanım itibariyle işlenebilir bir zaman işaretleme dilidir. Bu durumda bilgisayarlar tarafından TimeML ile işaretlenmiş bir dosyanın işlenebilmesi beklenir.

Zaman işaretleme dillerinin en sık kullanıldığı konular, kronoloji çıkarımı ve soru cevaplama'dır. Elbette bu tez çalışmasından önce de bu konularda TimeML üzerinde yazılmış çeşitli uygulamalar bulunmaktaydı. Bu tez kapsamında, yapılan iyileştirmeler mevcut çalışmalara eklenerek, bu iki önemli konuda, Türkçe metinlerdeki zaman işaretlemesini de içeren uygulamalar geliştirilmiştir. Bu uygulamaların teknik detayları ve kullanımları bu bölümde açıklanacaktır.

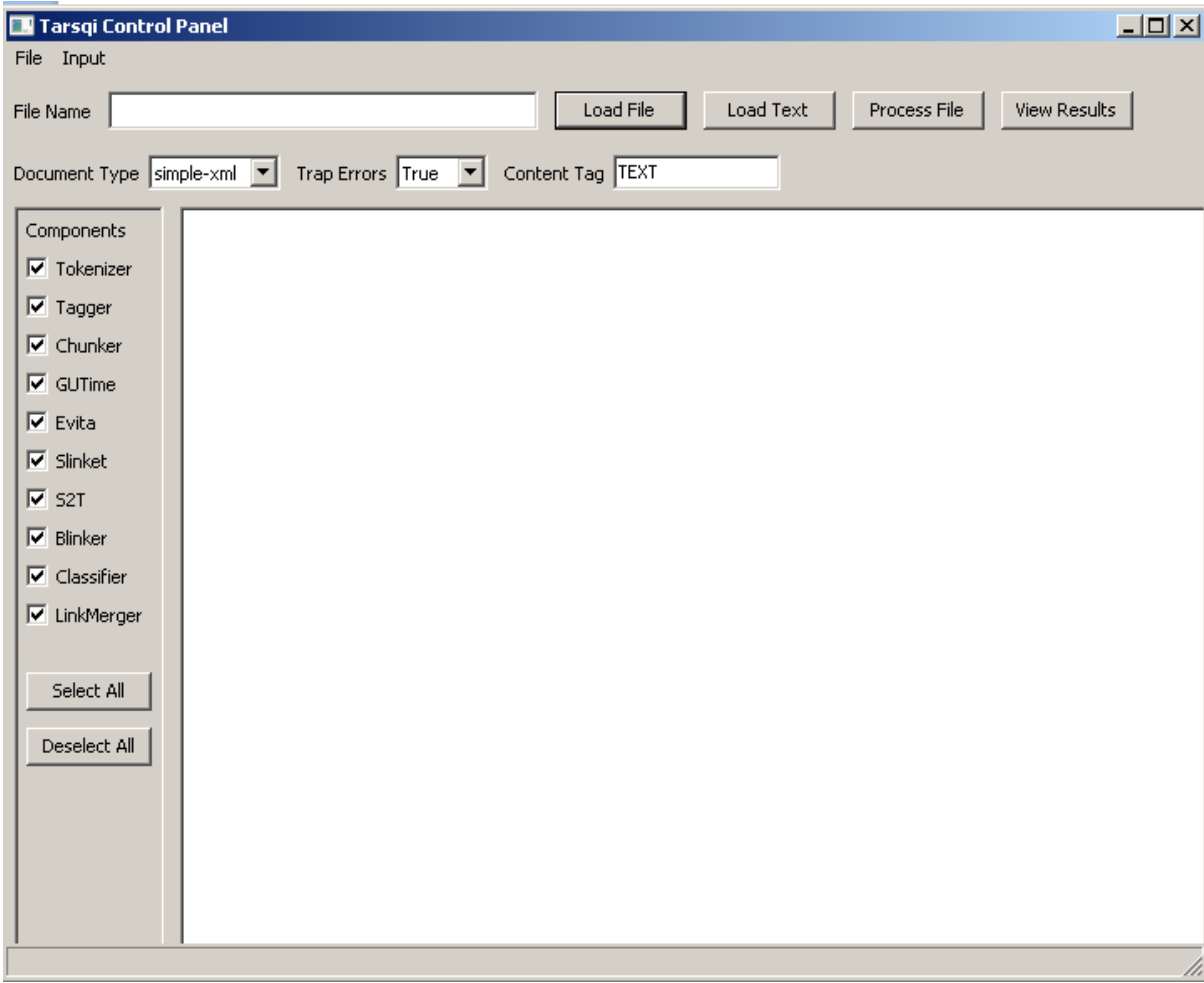
9.1 Kronoloji Çıkarımı

Kronoloji çıkarımı bir metindeki çeşitli olayların arka arkaya sıralanması ile elde edilir. Diğer bir deyişle bir metindeki olayların kronolojik gösterimidir. Elbette Türkçe gibi doğal dilde yazılmış bir metindeki olayların kronolojik olarak anlatılması beklenemez. Genelde doğal dilde yazılan metinlerde olayların aktarılması sırasında zamansal olarak bir sıralama yapılıyor olsa da, metnin herhangi bir noktasında bu zamansal dizilim bozularak sıralamanın dışına çıkılabilir.

Bu tip metinlerde zamansal işaretleme dilleri sayesinde olayların birbiri ile olan ilişkisi, işlenebilir bir gösterim ile işaretlenir.

Bu anlamda yazılan ve TimeML üzerinde çalışan en kapsamlı uygulamalardan birisi de Tarsqi ismi verilen projedir. (Tarsqi 2010)

Bu proje JAVA dilinde geliştirilmiş olmasına karşılık metin üzerindeki ön işlemleri Python dilinde yapmaktadır.



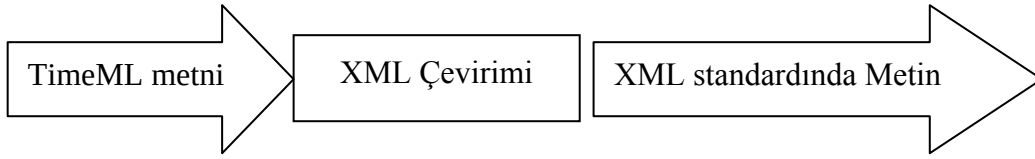
Şekil 9.1 Tarsqi Kontrol Paneli Görüntüsü

İngilizce üzerinde doğal dil işleme çalışmaları, Türkçedekilere oranla daha başarılı sonuçlar üretebilmektedir. Örneğin sözdizim (syntax) açısından İngilizce metinlerde tam olarak bir cümlenin analizi yapılabilmekte ve bu analiz sonucu TimeML işaretleme diline otomatik bir girdi olabilmektedir.

Türkçe için ne yazık ki sözdizim aşamasında henüz bu seviyede bir başarı yakalanamamıştır. Bunun sebebi, Türkçe'nin diğer dillere göre daha esnek yapıda olması ve şekilbilimsel olarak bir kelimenin alabileceği ek sayısının fazla olmasıdır. Sözdizimsel olarak Türkçe'deki cümle öğeleri, cümlenin içinde hemen her yerde bulunabilir, bu esnek yapı, doğal dil işleme çalışmalarını güçleştirmektedir. Bu çalışma kapsamındaki elle yapılan işaretleme işlemi, İngilizce için phytonda yazılmış bir ön işlemde geçirilmektedir.

Bu ön işlemeden sonra TimeML dilindeki dosyalar XML diline çevrilmektedir. Aslında kendisi de XML standardına uygun olarak yazılmış olan TimeML dili, zamansal bilgiler dışındaki bilgileri etiketlemez. Bu yüzden metin içerisinde etiketsiz bir kısım bilgi oluşur.

İşte Tarsqi projesindeki metnin işlenmesi için yapılması gereken ilk aşama bu metnin XML standardına çevrilmesidir.



Şekil 9.2 TimeML dönüşümü

Bu durum bir metin için aşağıdaki şekilde çalışmaktadır.

Örneğin girdi olarak aşağıdaki metnin verildiğini kabul edelim:

<DOC>

<DOCID> Simple Test </DOCID>

<TEXT>

Ahmet, Ayşe ittikten sonra düştü.

</TEXT>

</DOC>

Yukarıdaki bu metin TimeML ile etiketlendikten sonra aşağıdaki hali alır:

<DOC>

<DOCID>

Simple Test

</DOCID>

<TEXT>

<s>

Ahmet, Ayşe

<EVENT eid="e2" class="OCCURRENCE">

ittikten

</EVENT>

<MAKEINSTANCE eventID="e2" polarity="POS" pos="VERB" eiid="ei2"
tense="PAST" aspect="NONE />

sonra

<EVENT eid="e1" class="OCCURRENCE">

düştü

</EVENT>

<MAKEINSTANCE eventID="e1" polarity="POS" pos="VERB" eiid="ei1"
tense="PAST" aspect="NONE />

<TLINK lid="l88" relatedToEventInstance="ei1" relType="BEFORE"
eventInstanceID="ei2" origin="Blinker - Event:Signal:Event i />

</TEXT>

Görüldüğü üzere bu metinde “Ahmet”, ”Ayşe” veya “sonra” gibi kelimeler işaretlenmeden XML yapısının dışında bırakılmıştır. Bu metindeki diğer işaretlenmemiş kelimeler de işaretlendikten sonra aşağıdaki şekilde bir metin elde edilir.

<DOC>

<DOCID>

Simple Test

</DOCID>

<TEXT>

<s>

<NG>

<lex pos="NNP">

Ahmet

</lex>

</NG>

<lex pos=",">

,

</lex>

<NG>

<lex pos="NNP">

Ayşe

</lex>

</NG>

<VG>

<lex pos="VBD">

<EVENT eid="e2" class="OCCURRENCE">

ittikten

</EVENT>

</lex>

<MAKEINSTANCE eventID="e2" polarity="POS" pos="VERB" eiid="ei2"
tense="PAST" aspect="NONE />

</VG>

<VG>

<lex pos="VBD">

<EVENT eid="e1" class="OCCURRENCE">

düştü

</EVENT>

</lex>

<MAKEINSTANCE eventID="e1" polarity="POS" pos="VERB" eiid="ei1"
tense="PAST" aspect="NONE />

</VG>

<lex pos="IN">

sonra

</lex>

<lex pos=".">

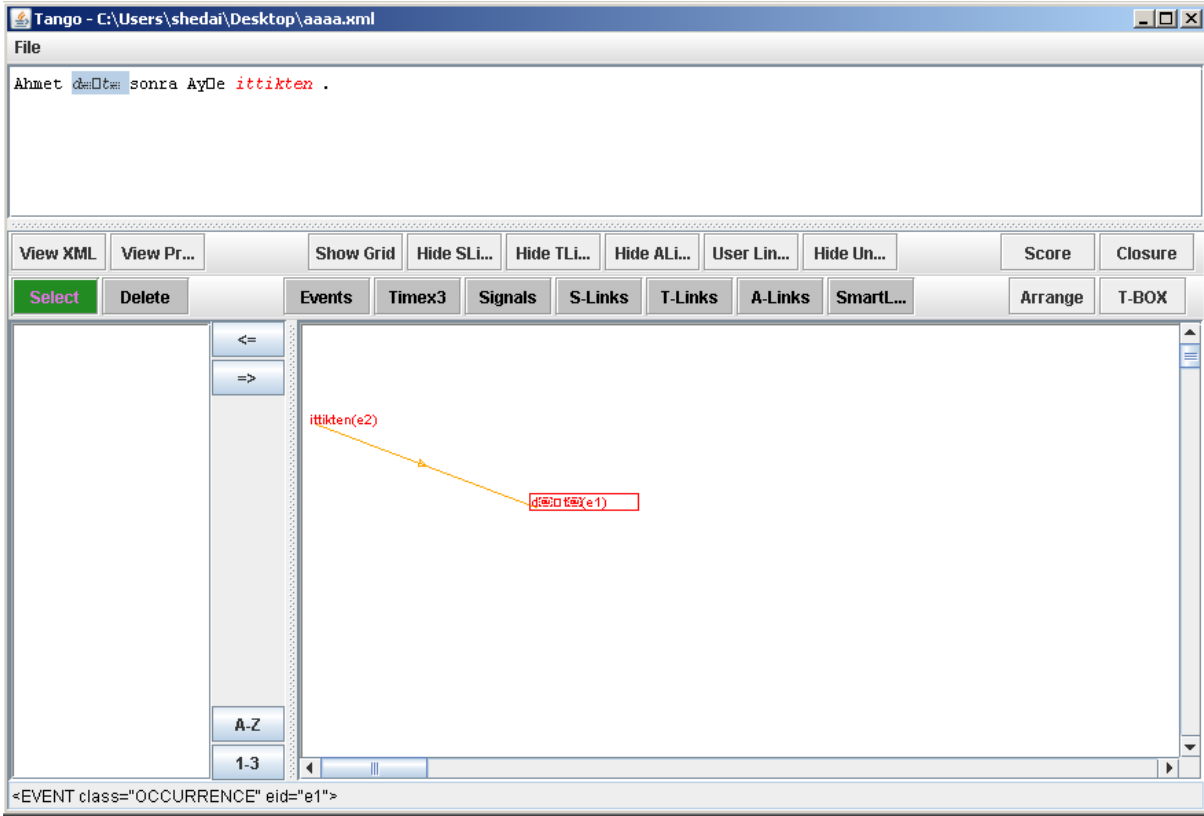
.

</lex>

</s>

Yukarıdaki bu yeni halinde, metin üzerinde bir önceki halde işaretlenmemiş kelimeler de işaretlenmiştir.

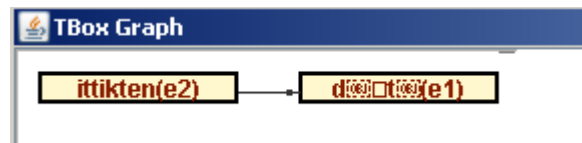
Yukarıdaki bu çalışma sonucunda işaretlenmiş metin üzerinden bir görsel kronoloji gösterimi mümkündür.



Şekil 9.3 Tango ekran görüntüsü

Tarsqi uygulamasında işlenmiş bu metnin gösterilmiş hali Şekil 9.3’de verilmiştir.

Ayrıca bu görsel çalışma alanı dışında ekranda tam olarak kronoloji gösterimi yapılması da T-BOX isimli uygulama ile mümkündür.



Şekil 9.4 T-Box Ekran Görüntüsü

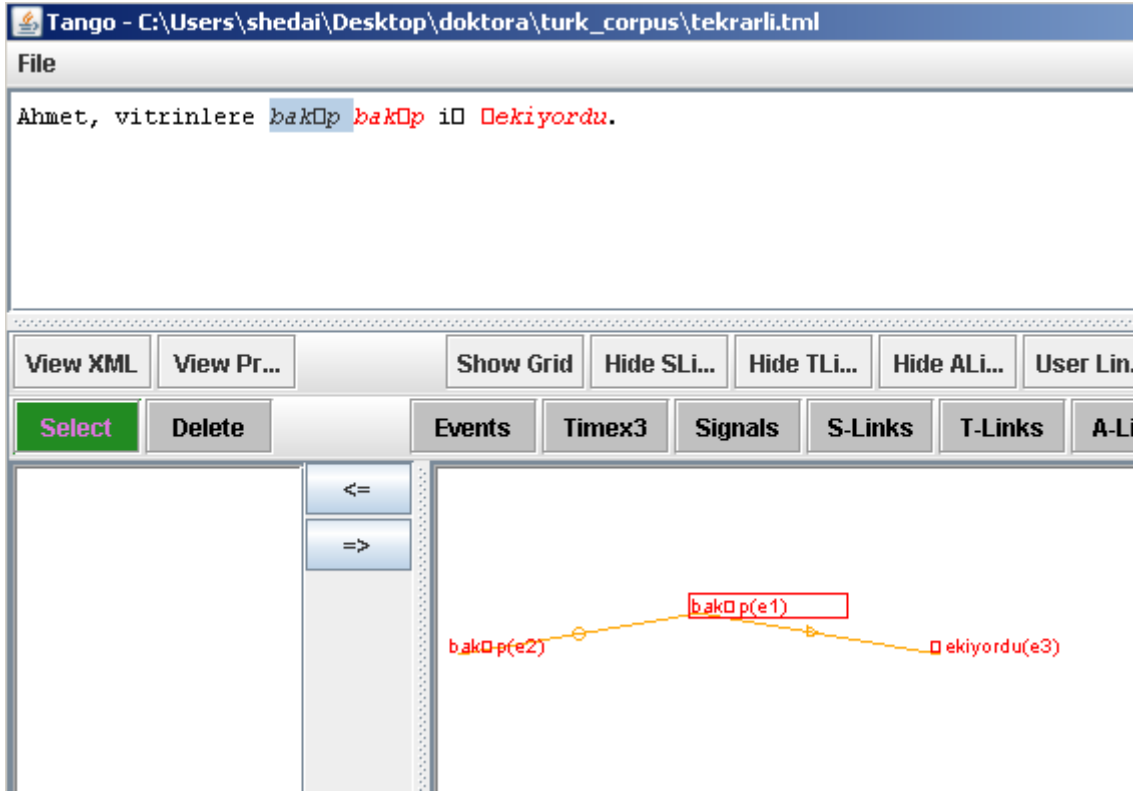
Şekil 9.4’de görülen bu mevcut çalışan koda, bu tez kapsamında, daha önceki bölümlerde bahsedilen Türkçe konusundaki ilaveler yapılmış, ayrıca bu ilavelerin görsel olarak gösterilmesi mümkün hale getirilmiştir.

Örneğin bu çalışma kapsamında doğrusal mantık üzerine kurulu olan TimeML dilini işleyen Tarsqi ve T-BOX uygulamalarına tekrarlı ilişki gösterme yeteneği kazandırılmıştır.

Bu durum aşağıdaki uygulama ile görülebilir.

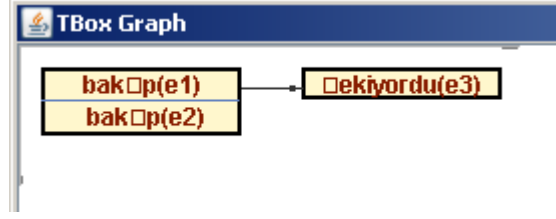
Örnek cümlemiz bu defa “Ahmet, vitrinlere bakıp bakıp iç çekiyordu” olsun. Bu cümlede “bakıp bakıp” tekrarı bir tekrarlı olayı ifade etmektedir.

Normalde bu iki eylem arasında önce ve sonra ilişkisi olması beklenir. TimeML iyileştirilme yapılmadan önce bu durumu Şekil 9.6’da göstermektedir:



Şekil 9.5 TANGO örnek çalışma

Gösterim olarak bu durum Şekil 9.6’da çözülmüştür:



Şekil 9.6 Örnek çalışmanın TBox Çıktısı

Görüldüğü üzere iki eylemin birbirini içermesi ve tekrar etmesi durumu için ilave bir gösterim şekli eklenmiş ve TimeML üzerinde ilgili işaretleme ile bu gösterimin Tarsqi projesine adapte edilmesi sağlanmıştır.

10. SONUÇLAR VE GELECEK ÇALIŞMALAR

Bu tez çalışması kapsamında zaman üzerinde modelleme yapan zamansal mantıklar incelenmiş bu zamansal mantıkların Türkçedeki zamansal ifadeler ile olan uyumu ve farklılıkları ortaya konulmuş, bir zamansal mantığın işlenebilir bir dile çevrilmesi incelenmiş ve bu anlamda TimeML dili üzerine Türkçedeki olayları sıralayabilecek ve bu olaylar arasında zamansal modelleme yapabilecek özellikler eklenmiştir.

Bu eklenen özellikler, AZM ve Reichenbach zamansal mantık seviyelerinde incelenebilir. Bu seviyelerden Reichenbach seviyesinde bulunan 13 ilişki tipi, Türkçe için genişletilerek 20'ye çıkarılmıştır. İlave 7 ilişki tipi, derlem çalışmalarında %82'lik başarı artışı sağlamıştır.

Ayrıca AZM seviyesinde 4 yeni ilişki tipi önerilmiştir. Bu ilişki tiplerinin derlem çalışmasını etkilememe sebebi, derlem kapsamındaki eserlerin, basit çocuk hikayeleri olmasıdır.

Bu çalışma üzerine yapılan testler, mevcut TimeML başarısının oldukça düşük olduğunu ancak önerilen geliştirmelerle bu oranın yüzde yüz başarıya kadar ulaştığını göstermiştir.

Bu çalışmada yapılan Türkçe iyileştirmesi ne yazık ki her durumda aynı şekilde %100 başarı sağlayamaz. Birleşik zamanlı kullanımlar, Türkçe için uygun olsa da genelde yabancı dilden yapılan çevirimlerle Türkçeye kazandırılmıştır.

Örneğin, “beklemiş olacağım” ifadesi, Türkçedeki “bekleyeceğim” ifadesi ile anlatılabilir. Bu iki ifade gerek anlam ve gerekse de zamansal kurgu açısından aynı ifadedir. “bekleyeceğim” ifadesinde, vurgu kayması olmuştur. Buradaki “bekledi” kelimesi, bekleme eyleminin karşı tarafın bileceğini ve ileride bu eylemi karşı tarafın beyan edeceğini vurgular.

Bu anlamda iki kullanım, zamansal kurguda fark oluşturur. Bu fark ne yazık ki çalışmamız kapsamında yapılan iyileştirmede bulunmamaktadır. Bu iyileştirmeye ayrıca ihtiyaç da duyulmamıştır.

Gelecek çalışmalar olarak, daha geniş ve bu konuda ileride çalışacak kişilere temel teşkil edecek bir derlemin hazırlanması sağlanabilir. Bu tez çalışmasındaki derlem genişletilerek, basit çocuk hikayelerinin dışında, gazete haberleri, akademik yayınlar, ansiklopedi veya çeviri eserler gibi farklı alanlarda bulunan kaynaklardan faydalanılması, çalışmanın başarısını arttıracak gibi, bu tez çalışması kapsamında karşılaşılmayan zamansal ifadelerin ortaya çıkmasını da sağlayabilir.

Ayrıca güncel olarak kullanılan, görsel zaman ilişkilendirme modelleri için bir temel oluşturulabilir ve TimeML dili üzerinde genişletilen yeni mantık kullanılarak, bir görsel çizim programı geliştirilebilir.

KAYNAKLAR

- Andrews, P., (2002), *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof*, Dordrecht, Kluwer Academic Publishers, New York.
- Bailhache, P.,(1998), “How to Mix Alethic, Deontic, Temporal, Individual Modalities.”, *Logica Trianguli*, 2: 3–16.
- Bellini, P., Mattolini, R. ve Nesi, P., (2000), “Temporal Logics for Realtime System”, *Specifications ACM Computing Surveys*, 32(1):
- Ben-ari, M., Pnueli, A. ve Manna, Z. (1981), “The temporal logic of branching time”, *Annual Symposium of Principle of Programming Languages*, 164-176.
- Boolos, G., (1975), “On Second-Order Logic,” *The Journal of Philosophy*, 72: 509–527.
- Cantini, A., (1996), “Logical Frameworks for Truth and Abstraction: An Axiomatic Study, *Studies in Logic and the Foundations of Mathematics*”, Amsterdam, Elsevier.
- Clarke, E. M., Emerson, E. A., ve Sistla, A. P. , (1986), “Automatic verification of finite-state concurrent systems using temporal logic specifications”. *ACM Trans. Program. Lang. Syst.* 8, 2, 244–263.
- Clarke, E. M., ve Grumberg, O., (1987), “The model checking problem for concurrent systems with many similar processes. In *Proceedings of the International Conference on Temporal Logic in Specification*” Altrincham, 8–10
- Copi, I. M., (1971), “*Introduction to Logic*”, Macmillan, New York
- Özcan, E., (2003), “Towards an XML based standard for Timetabling Problems: TTML” *MISTA* 10-14 Mar 2003, 566-570
- Fischer, M., ve Ladner, R., (1979), “Propositional dynamic logic of regular programs”, *Journal of Computer and System Sciences*, 18: 194-211.
- Gochet, P., ve Gribomont, P., (2003), *Epistemic Logic* forthcoming in *Handbook of the History and Philosophy of Logic*, edited by Gabbay, D.M. and Woods, J. Amsterdam: Elsevier Science.
- Hintikka, J., ve Halonen, I., (1998), *Epistemic Logic*, *Routledge Encyclopedia of Philosophy*, volume 1. London: Routledge.
- Hirschman, L., ve Gaizauskas, R., (2001), “*Natural Language Question Answering. The View from Here. Natural Language Engineering*”, Cambridge University Press.
- Kingston, J., (1999), *STTL:A Standard Timetabling Language*, Basser Department of Computer Science, The University of Sydney Australia
- Jerry, R., Hobbs, J., Pan, F., (2006), “*Institute Time Ontology in OWL W3C*”, Working Draft 27 September 2006
- Mani, Inderjeet (2001), “*Automatic Summarization*”, ISBN 1-58811-060-5.
- Melliar-Smith, P. M. (1987), “Extending interval logic to real time systems. In *Proceedings of the Conference on Temporal Logic Specification*”, B. Banieqbal, H. Barringer, and A. Pnueli, Eds. Springer-Verlag, New York, 224–242.

Meyer, J.,-J., (2001). *Epistemic Logic*, The Blackwell Guide to Philosophical Logic L. Goble, ed., Oxford: Blackwell Publishers.

Moszkowski, B. ve Manna, Z., (1984), “Reasoning in interval logic”, Proceedings of the ACM/NSF/ONR Workshop on Logics of Programs, Springer-Verlag, Secaucus, NJ, 371–384.

Moszkowski, B., C., (1986), “Executing temporal logic programs.” Ph.D. Dissertation. Cambridge University Press, New York, NY.

Penev, Wong, A., R., (2006), “Shallow NLP techniques for internet search”, ACM International Conference Proceeding Series; Vol. 171 Proceedings of the 29th Australasian Computer Science Conference - Volume 48: 167 - 176

Pnueli, A., (1971), “The temporal logic of programs.”, Proceedings of the 18th Annual Symposium on Foundations of Computer Science (Providence, RI.). IEEE, New York, pp. 46-57.

Reichenbach, H., (1947), “Elements of Symbolic Logic”, Macmillan, New York

Rosner, R. ve Pnueli, A., (1986), “A choppy logic”, Proceedings of the First IEEE Symposium on Logic in Computer Science, IEEE Press, Piscataway, NJ, 306–313.

Razouk, R. ve Gorlick, M., (1989). “Real-time interval logic for reasoning about executions of real-time programs”. SIGSOFT Softw. Eng. Notes 14, 8 (Dec. 1989), 10–19.

Schwartz, R. L., Melliar-Smith, P. M., ve Vogt, F. H., (1983), “An interval logic for higherlevel temporal reasoning”, Proceedings of the Second ACM Symposium on Principles of Distributed Computing (Aug.), ACM Press, New York, NY, 173–186.

Stirling, C., (1987), “Comparing linear and branching time temporal logics.”, Proceedings of the International Conference on Temporal Logic in Specification, Altrincham, UK, Apr. 8–10

TERQAS, (2002) ,“Time and Event Recognition for Question Answering Systems” An ARDA Workshop on Advanced Question Answering Technology January - July, 2002

TARSQI, (2010),. “Temporal Awareness and Reasoning Systems for Question Interpretation.” Prime Contractor Brandeis University (PI: James Pustejovsky) Sub-Contractor 1 Georgetown University (PI: Inderjeet Mani) Sub-Contractor 2 ISI (PI: Jerry Hobbs), <http://www.timeml.org/site/tarsqi/>, tarama tarihi Mayıs 2010.

Udo, H. ve Stefan, S., (2002), “Massive bio-ontology engineering for NLP”, Proceedings of the second international conference on Human Language Technology Research : 68 - 76

Väänänen, J., (2001), “Second-Order Logic and Foundations of Mathematics”, The Bulletin of Symbolic Logic, 7: 504–520

EKLER

EK 1 Örnek Türkçe TimeML hikayesi

Hikaye ismi : Orman Perisinin Gülleri

<s>Yemyeşil ağaçlarla kaplı ormanın birinde genç bir peri <EVENT eid="e1" class="DURUMSAL" stem="yaşa" aspect="MİŞLİ" tense="PRESENT" polarity="POS" pos="VERB">yaşarmış </event></s> <s>Bu peri çiçeklerden en çok gülleri <EVENT eid="e2" class="I STATE" stem="sev" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB">severmiş</event></s>. <s>Evinin bahçesinde renk renk güller <EVENT eid="e3" class="I ACTION" stem="yetiştir" aspect="MİŞLİ" tense="PRESENT" polarity="POS" pos="VERB">yetiştirmiş</s>. <s>Bu güller o kadar taze ve <EVENT eid="e4" class="STATE" stem="güzel" aspect="MİŞLİ" tense="PRESENT" polarity="POS" pos="NOUN">güzellermiş ki gören herkes perinin güllerine hayran<EVENT eid="e5" class="DURUMSAL" stem="kal" aspect="MİŞLİ" tense="PRESENT" polarity="POS" pos="VERB">kalırmış</s>. <s> Peri de güllerini çok <EVENT eid="e6" class="I STATE" stem="sev" aspect="" tense="PRESENT" polarity="POS" pos="VERB">sever, her sabah onları hem<EVENT eid="e7" class="I ACTION" stem="sula" aspect="MİŞLİ" tense="PRESENT" polarity="POS" pos="VERB"> sular hem de onlarla<EVENT eid="e8" class="DURUMSAL" stem="konuş" aspect="MİŞLİ" tense="PRESENT" polarity="POS" pos="VERB"> konuşmuş</s>. <s>Genç peri gülleriyle çok<EVENT eid="e9" class="STATE" stem="mutlu" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="NOUN"> mutluymuş, ama onu üzen bir durum<EVENT eid="e10" class="I STATE" stem="var" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB"> varmış</s>. <s> Peri güllerini çok sevdiği için onların solmalarına <EVENT eid="e11" class="I STATE" stem="dayan" aspect="MİŞLİ" tense="PRESENT" polarity="NEG" pos="VERB">dayanamamış</s>. <s>Güllerin bir süre sonra solması çok <EVENT eid="e12" class="STATE" stem="doğal" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="NOUN">doğalmış, fakat genç peri güllerinin solmasına çok<EVENT eid="e13" class="I STATE" stem="üzül" aspect="CONTINUOUS" tense="PRESENT" polarity="POS" pos="VERB"> üzülüyor, güllerinin hep ilk günkü gibi taze ve diri kalmalarını<EVENT eid="e14" class="I STATE" stem="iste" aspect="MİŞLİ" tense="PRESENT" polarity="POS" pos="VERB"> istiyormuş</s>. <s>Kendi kendine “güllerim hep böyle güzel<EVENT

eid="e15" class="DURUMSAL" stem="kal" aspect="SİMPLE" tense="PRESENT" polarity="POS" pos="VERB"> kalsa! O zaman hiç mutsuz <EVENT eid="e16" class="DURUMSAL" stem="ol" aspect="SİMPLE" tense="PRESENT" polarity="NEG" pos="VERB">olmam.> <EVENT eid="e17" class="BİLDİRME" stem="de" aspect="MİŞLİ" tense="PRESENT" polarity="POS" pos="VERB">diyormuş</s>. <s>Bir sabah çiçeklerini yine <EVENT eid="e18" class="I ACTION" stem="sula" aspect="SİMPLE" tense="PRESENT" polarity="POS" pos="VERB">sularken perinin dikkatini sarı renkte bir gül tomurcuğu <EVENT eid="e19" class="KAVRAMA" stem="çek" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB">çekmiş</s>.<s> Bu tomurcuk da diğer gül tomurcukları gibi pek<EVENT eid="e20" class="STATE" stem="güzel" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="NOUN"> güzelmış</s>.<s> Fakat rengi diğerlerinden <EVENT eid="e21" class="STATE" stem="apayrı" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="NOUN">apayrımış</s>.<s> Çok daha güzel ve değişik bir <EVENT eid="e22" class="STATE" stem="ton" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="NOUN">tondaymış tomurcuğun rengi</s>.<s> Bu yüzden, genç peri sarı tomurcuğa daha özenli bakmaya<EVENT eid="e23" class="DURUMSAL" stem="başla" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB"> başlamış</s>. <s>Her <TIMEX>sabah ona “küçük sarı tomurcuk <EVENT eid="e24" class="DURUMSAL" stem="büyü" aspect="" tense="FUTURE" polarity="POS" pos="VERB">büyüyecek, kocaman güzel bir gül<EVENT eid="e25" class="DURUMSAL" stem="ol" aspect="MİŞLİ" tense="FUTURE" polarity="POS" pos="VERB"> olacak” diye güzel sözler <EVENT eid="e26" class="BİLDİRME" stem="söyle" aspect="MİŞLİ" tense="PRESENT" polarity="POS" pos="VERB">söylüyormuş</s>.<s> Tomurcuk da bunu <EVENT eid="e27" class="I STATE" stem="anla" aspect="MİŞLİ" tense="PRESENT" polarity="POS" pos="VERB">anıyormuş gibi <TIMEX>günden güne daha da güzelleşerek<EVENT eid="e28" class="DURUMSAL" stem="büyü" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB"> büyümüş</s>. <s>Kocaman bir gül <EVENT eid="e29" class="DURUMSAL" stem="ol" aspect="DİLİ" tense="PAST" polarity="POS" pos="VERB">olduğunda </event> ise bahçedeki diğer güllerin arasında tıpkı gökyüzündeki güneş gibi<EVENT eid="e30" class="I ACTION" stem="ışıl" aspect="MİŞLİ" tense="PRESENT" polarity="POS" pos="VERB"> ışıldıyormuş</event></s>.<s> O kadar <EVENT eid="e31" class="STATE" stem="güzel" aspect="MİŞLİ" tense="PAST"

polarity="POS" pos="NOUN">güzelmiş ki onu görenler sarı güle bakmaya <EVENT eid="e32" class="KAVRAMA" stem="doy" aspect="MİŞLİ" tense="PRESENT" polarity="NEG" pos="VERB">doyamıyorlarmış</event></s>.<s> Peri de bunun <EVENT eid="e33" class="STATE" stem="farkında" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="NOUN">farkındaymış</event> ve çok <EVENT eid="e34" class="STATE" stem="mutlu" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="NOUN">mutluymuş</event></s>.<s> Fakat sarı gülün de bir gün solacağını bildiği için, içten içe bir üzüntü <EVENT eid="e35" class="KAVRAMA" stem="duy" aspect="MİŞLİ" tense="PRESENT" polarity="POS" pos="VERB">duyuyormuş</event></s>.<s> Aradan bir gün <EVENT eid="e36" class="DURUMSAL" stem="geç" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB">geçmiş</event>, bir hafta <EVENT eid="e37" class="DURUMSAL" stem="geç" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB">geçmiş</event>, bir ay <EVENT eid="e38" class="DURUMSAL" stem="geç" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB">geçmiş</event></s>.<s> Bu süre içinde bahçedeki bütün güller <EVENT eid="e39" class="I ACTION" stem="sol" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB">solmuş</event>, yerlerini yeni tomurcuklara <EVENT eid="e40" class="DURUMSAL" stem="bırak" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB">bırakmışlar</event>: güzel, sarı gül dışında</s>! <s> Bir ay geçmesine rağmen sarı gül <EVENT eid="e41" class="I ACTION" stem="sol" aspect="MİŞLİ" tense="PAST" polarity="NEG" pos="VERB">solmamış</event>, benzersiz güzelliğinden hiçbir şey <EVENT eid="e42" class="I STATE" stem="kaybet" aspect="MİŞLİ" tense="PAST" polarity="NEG" pos="VERB">kaybetmemiş</event></s>.<s> Peri ilk başta bu işe çok <EVENT eid="e43" class="I STATE" stem="şaşıır" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB">şaşırmış</event> fakat yine de <EVENT eid="e44" class="STATE" stem="sevinçli" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="NOUN">sevinçliymiş</event></s>.<s> Çünkü güllerinin en güzeli <EVENT eid="e45" class="I ACTION" stem="sol" aspect="MİŞLİ" tense="PAST" polarity="NEG" pos="VERB">solmamışmış</event></s>.<s> İyi yürekli peri, her gün onu evinin penceresinden <EVENT eid="e46" class="KAVRAMA" stem="seyret" aspect="CONTINIOUS" tense="PRESENT" polarity="POS"

pos="VERB">seyrediyor</event>, onu özenle <EVENT eid="e47" class="I ACTION" stem="sula" aspect="CONTINUOUS" tense="PRESENT" polarity="POS" pos="VERB">suluyor</event>, ona güzel sözler <EVENT eid="e48" class="BİLDİRME" stem="söyle" aspect="MİŞLİ" tense="PRESENT" polarity="POS" pos="VERB">söylüyormuş</event></s>.<s> Gel zaman git zaman; peri, bu işten sıkılmaya <EVENT eid="e49" class="DURUMSAL" stem="başla" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB">başlamış</event></s>.<s> Sarı gül hiç<EVENT eid="e50" class="I ACTION" stem="sol" aspect="MİŞLİ" tense="PRESENT" polarity="NEG" pos="VERB"> solmuyormuş</event>, fakat bu periye artık XXXXmutluluk vermemeye <EVENT eid="e51" class="DURUMSAL" stem="başla" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB">başlamış</event></s>.<s> Çünkü peri sarı güle dair hiçbir umut <EVENT eid="e52" class="I ACTION" stem="taşı" aspect="MİŞLİ" tense="PRESENT" polarity="POS" pos="VERB">taşıymıyormuş</event> içinde</s>.<s> Önceden gülleri XXXXsolduğu vakit, yeni tomurcukların ne zaman çıkacağınıXXXX merak ederek onlarla sabırla <EVENT eid="e53" class="I ACTION" stem="ilgilen" aspect="SIMPLE" tense="PRESENT" polarity="POS" pos="VERB">ilgilenir</event>, umutla güllerinin XXXaçılacağı zamanı <EVENT eid="e54" class="I STATE" stem="bekle" aspect="MİŞLİ" tense="PRESENT" polarity="POS" pos="VERB">beklermiş</event></s>.<s> Fakat şimdi sarı gül hiç XXXXsolmadığı için böyle düşünceleri<EVENT eid="e55" class="DURUMSAL" stem="kal" aspect="MİŞLİ" tense="PAST" polarity="NEG" pos="VERB"> kalmamış</event></s>.<s> Bu da periye birXXXX zaman sonra XXXmutsuz<EVENT eid="e56" class="I STATE" stem="et" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB"> etmiş</event></s>.<s> Yetiştirdiği güllerinin solmamasını XXXXisteyerek ne kadar yanlış düşündüğünü <EVENT eid="e57" class="I STATE" stem="anla" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB">anlamış</event></s>.<s> Her şeyi doğal haliyle sevmek en <EVENT eid="e58" class="STATE" stem="güzel" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="NOUN">güzeliymiş</event></s>.<s> Bu yüzden XXXo günden sonra orman perisi, doğadaki her şeyi XXXolduğu gibi kabul etmeye <EVENT eid="e59" class="I ACTION" stem="ver" aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB">karar vermiş</event></s>.<s> Orman perisiXXX uzun yıllar, bahçesinde yetiştirdiği güllerle beraber evinde mutlu bir hayat <EVENT eid="e60" class="DURUMSAL" stem="sür"

aspect="MİŞLİ" tense="PAST" polarity="POS" pos="VERB">sürmüş</event></s>.

<SLINK lid="l1" eventInstanceID="e29" subordinatedEventInstance="<EVENT eid="e29"
class="I ACTION" stem="ışıl-da" aspect="MİŞLİ" tense="PRESENT" polarity="POS"
pos="VERB">e30" relType="EVIDENTIAL"/>

YAYINLAR

Bu bölümde, bu doktora çalışması kapsamında yapılan yayınlara yer verilmiştir. Ayrıca makalelerin tam metinleri ekte sunulmuştur.

Ulusal Konferanslar

Akademik Bilişim 2010 (Muğla Üniversitesi, 10-12 Şubat 2010) Bildiri kitabı, syf. 88
“Reichenbach ve Allen Zamansal Mantığı ile TimeML”, Şadi Evren ŞEKER, Banu DİRİ

Uluslararası Konferanslar

CSW 2010, (Koç Üniversitesi , 21 Şubat 2010) “Event Ordering for Turkish Natural Language Texts” , Şadi Evren ŞEKER, Banu DİRİ ,Paper

ICAI 2010, WolrdComp 2010,(July 12-15, 2010, USA) , “TimeML and Turkish Temporal Logic”, Şadi Evren ŞEKER, Banu DİRİ, Accepted , Regular Research Paper

ÖZGEÇMİŞ

Doğum tarihi	01.01.1979	
Doğum yeri	İstanbul	
Lise	1993-1996	Üsküdar Beylerbeyi Lisesi
Lisans	1996-2000	Yeditepe Üniversitesi Mühendislik Fak. Bilgisayar Mühendisliği Bölümü
Yüksek Lisans	2000-2003	Yeditepe Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Müh. Programı
Yüksek Lisans	2001-2004	İstanbul Teknik Üniversitesi Sosyal Bilimler Enstitüsü Bilim Teknoloji ve Toplum Programı
Doktora	2005-2010	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Müh. Programı

Çalıştığı kurumlar

2000-2004	Yeditepe Üniv. Bilgiayar Mühendisliği Araştırma Görevlisi
2005-Devam	Set Eğitim ve Danışmanlık, Bilgisayar Bilimleri Danışmanı