

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

79218

**WORLD-WIDE-WEB ÜZERİNDE
ÇOKLUORTAM VERİTABANI
UYGULAMASI**

Bilgisayar Mühendisi F. Önder YILDIRIM

F.B.E.Bilgisayar Bilimleri Mühendisliği Anabilimdalı Bilgisayar Bilimleri Mühendisliği
Programında Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd.Doç.Dr. Selim AKYOKUŞ

İSTANBUL, 1998

İÇİNDEKİLER

İÇİNDEKİLER	II
KISALTMALAR	VI
ŞEKİLLER LİSTESİ.....	IX
TABLolar LİSTESİ.....	XI
ÖZET.....	XII
ABSTRACT	XIII
1. GİRİŞ.....	1
1.1 Çokluortam VTYS Nedir?.....	3
1.2 Arkaplan Veritabanı.....	3
1.3 Çokluortam Dokümanlarının Modellenmesi	6
1.4 Zamana Bağlı İlişkiler.....	8
1.5 Yersel İlişkiler.....	10
1.6 Etkileşimli Sunum Nesneleri	10
1.7 Eşzamanlama İlişkileri.....	11
1.8 Servis Kalitesi Parametreleri	12
2. ÖNCEKİ ÇALIŞMALAR.....	14
3. AMAÇ.....	17
3.1 Çokluortam Haberler Veritabanı Sisteminin Temel Fonksiyonları 18	
3.1.1 <i>Sorgulama</i>	18
3.1.2 <i>Veri Girişi</i>	18
4. TASARIM	20
4.1 Nesne Tabanlı Tasarım	20
4.2 Sınıf Açıklamaları.....	20
4.3 Nesne Mimarisi.....	31
4.3.1 <i>Kap hiyerarşisi</i>	31
4.3.2 <i>Nesne hiyerarşisi</i>	32
4.4 Nesne Tabanlı Yapıdan İlişkisel Yapıya Geçiş.....	37
4.4.1 <i>Nesne düzleme</i>	39

4.5	Nesne Belirteçlerinin Yapısı	41
4.6	İlişkisel Şema	44
5.	UYGULAMA	58
5.1	Uygulama Programı	60
5.2	Veri Giriş Programı	60
5.2.1	<i>Stiller menüsü</i>	62
5.2.2	<i>Kişiler menüsü</i>	63
5.2.3	<i>Haber tipleri menüsü</i>	64
5.2.4	<i>Haber konuları menüsü</i>	65
5.2.5	<i>Önem seviyeleri menüsü</i>	66
5.2.6	<i>Coğrafi bölgeler menüsü</i>	67
5.2.7	<i>Saklama ortamı türleri menüsü</i>	68
5.2.8	<i>MIME türleri menüsü</i>	69
5.2.9	<i>Rol türleri menüsü</i>	70
5.2.10	<i>Resimler menüsü</i>	72
5.2.11	<i>Videolar menüsü</i>	73
5.2.12	<i>Audiolar menüsü</i>	74
5.2.13	<i>İçerik nesnelere menüsü</i>	75
5.2.14	<i>Metinler menüsü</i>	77
5.2.15	<i>Arşiv nesnelere menüsü</i>	78
5.2.16	<i>Belge listesi</i>	79
5.2.17	<i>Belge düzenleme ekranı</i>	81
5.2.18	<i>Sayfa Düzenleyicisi</i>	89
5.2.19	<i>Nesne seçim formu</i>	96
5.2.20	<i>Video nesnesi formu</i>	97
5.2.21	<i>Video karakteristik vektörleri formu</i>	98
5.2.22	<i>Ses nesnesi formu</i>	99
5.2.23	<i>Ses karakteristik vektörleri formu</i>	101
5.2.24	<i>Resim nesnesi formu</i>	102
5.2.25	<i>İçerik nesnesi formu</i>	103
5.2.26	<i>Web bağlantısı nesne formu</i>	106
5.3	Veri Yayıncısı	107
5.3.1	<i>Mimari</i>	108
5.3.1.1	Web modülü	108
5.3.1.2	Veritabanı	108
5.3.1.3	Oturum	108
5.3.1.4	Tablolar ve sorgular	109
5.3.1.5	Sayfa üreteçleri	109
5.3.2	<i>Çok-sicimli bağlantı yönetimi</i>	109
5.3.3	<i>WWW sitesi yönetimi</i>	110
5.3.4	<i>WWW sunucu etkileşimi</i>	112
5.3.4.1	Süreç-dışı etkileşim	112
5.3.4.2	Süreç -içi etkileşim	113
5.3.5	<i>Sorgu yönetici</i>	113
5.3.6	<i>HTML üretici</i>	118
5.3.6.1	Ana sayfa	121
5.3.6.2	Belge listesi	121

5.3.6.3	Belge açıklamaları	122
5.3.6.4	Sayfa	124
5.3.6.5	Video açıklamaları.....	125
5.3.6.6	Ses açıklamaları.....	127
5.3.6.7	Resim açıklamaları	128
5.3.6.8	İçerik açıklamaları	130
5.3.6.9	Ortamin istemciye gönderilmesi.....	131
5.3.6.10	Özelleştirilmiş açıklama sorguları	132
5.3.6.11	Özelleştirilmiş belge sorgusu.....	133
5.3.6.12	Özelleştirilmiş ortam sorgusu	135
5.3.6.13	Kelime arama.....	138
5.4	Veritabanının Uygulanması	139
5.4.1	<i>Nesne tabanlı veritabanı üzerinde nesne tabanlı tasarımın uygulanması</i>	139
5.4.2	<i>Dayanıklı nesne dosya sistemi üzerinde nesne tabanlı tasarımın uygulanması</i>	140
5.4.3	<i>İVTYS üzerinde nesne tabanlı tasarımın uygulanması — Nesnelerin İVTYS'te depolanması</i>	140
5.4.4	<i>İVTYS üzerinde nesne tabanlı tasarımın uygulanması — Nesne tabanlı sistemin ilişkisel sisteme çevrilmesi</i>	141
5.4.5	<i>İVTYS uygulaması</i>	141
5.4.5.1	Tablolar	142
5.4.5.2	İndeksler	143
5.4.5.3	Tetikleyiciler	146
5.4.5.4	Üreticiler.....	147
5.4.5.5	Domenler	147
5.4.5.6	İlişkiler.....	152
6.	SONUÇ VE ÖNERİLER	154
6.1	İnternet Arama	154
6.2	Metin Arama.....	155
6.3	İçerik Arama (Content Based Retrieval – CBR).....	156
6.4	WWW bilgi girişi	156
6.5	Servis Kalitesi.....	157
6.6	Bilgi Girişi	157
7.	KAYNAKLAR	159
8.	EKLER	164
8.1	A : Tanım Listeleri.....	164
8.2	B : Veritabanı Şeması SQL Cümleleri.....	167
8.2.1	<i>Domen tanım cümleleri</i>	167
8.2.2	<i>Üretici tanım cümleleri</i>	169
8.2.3	<i>Tablo ve indeks tanım cümleleri</i>	170
8.2.4	<i>Tetikleyici tanım cümleleri</i>	175

9. ÖZGEÇMİŞ 179



KISALTMALAR

ADT	: Abstract Data Type
API	: Application Programming Interface
ATM	: Asynchronous Transfer Mode
BDE	: Borland Database Engine
BLOB	: Binary Large Object
CBR	: Content Based Retrieval
CGI	: Common Gateway Interface
COM	: Common Object Model
CORBA	: Common Object Request Broker Architecture
CSS	: Cascading Style Sheets
DBMS	: Database Management System
DDL	: Data Definition Language
DLL	: Dynamic Link Library
DML	: Data Manipulation Language
FDDI-II	: Fiber Distributed Data Interface – II
FR	: Frame Relay
FTR	: Full Text Retrieval
FTS	: Full Text Search
HTML	: Hypertext Mark-up Language
HTTP	: Hypertext Transfer Protocol
HyTime	: Hypermedia/Time-based Structuring Language
IIS	: Internet Information Server

IP : Internet Protocol

IPng : IP Next Generation veya IPv6

ISAPI : Internet Server Application Programming Interface

ISQL : InterBase SQL

İVTYS : İlişkisel Veritabanı Yönetim Sistemi

JPEG : Joint Photographers Expert Group

LAN : Local Area Network

LOB : Large Object

MHEG : Multimedia and Hypermedia Information Encoding Expert Group

MIME : Multipurpose Internet Mail Extensions

MPEG : Motion Pictures Expert Group

NSAPI : Netscape Server Application Programming Interface

OCPN : Object Composition Petri Net

ODA : Office Document Architecture

OID : Object Identity

OLE : Object Linking and Embedding

OMG : Object Management Group

PSOM : Persistent System Object Model

QOS : Quality of Service

RDBMS : Relational Database Management System

SGML : Standard Generalised Markup Language

SOM : System Object Model

SQL : Structured Query Language

SS : Style Sheets

TCP : Transmission Control Protocol
UDP : User Datagram Protocol
URI : Unified Resource Indicator
URL : Unified Resource Location
VOD : Video on Demand
VTYS : Veritabanı Yönetim Sistemi
WAN : Wide Area Network
WWW : World Wide Web



ŞEKİLLER LİSTESİ

Şekil 4-1 : Kap hiyerarşisi	31
Şekil 4-2 : Nesneler arasındaki ilişkiler	32
Şekil 4-3 : Nesne Hiyerarşisi	33
Şekil 4-4 : Nesne hiyerarşisi (devam)	34
Şekil 4-5 : Nesne tabanlı'dan ilişkisel'e çevrim süreci	38
Şekil 4-6 : Nesne Belirtecinin (OID) Yapısı	41
Şekil 5-1 : Sistem mimarisi	59
Şekil 5-2 : Ayarlar menüsü	61
Şekil 5-3 : Stil tanımları	62
Şekil 5-4 : Kişi tanımları	63
Şekil 5-5 : Haber tipi tanımları	64
Şekil 5-6 : Haber konusu tanımları	65
Şekil 5-7 : Önem seviyeleri tanımları	66
Şekil 5-8 : Coğrafi bölge tanımları	67
Şekil 5-9 : Saklama ortamı türü tanımları	68
Şekil 5-10 : MIME tipi tanımları	69
Şekil 5-11 : Rol tanımları	70
Şekil 5-12 : Nesnelere menüsü	71
Şekil 5-13 : Resim nesnelere	72
Şekil 5-14 : Video nesnelere	73
Şekil 5-15 : Audio nesnelere	74
Şekil 5-16 : İçerik nesnelere	76
Şekil 5-17 : Metin nesnelere	77
Şekil 5-18 : Arşiv nesnelere	78
Şekil 5-19 : Belge listesi	80
Şekil 5-20 : Belge ekranı - genel	81
Şekil 5-21 : Belge ekranı - özet/anahtar kelimeler	82
Şekil 5-22 : Belge ekranı - arşiv bilgisi	83
Şekil 5-23 : Belge ekranı - sayfalar	85
Şekil 5-24 : Belge ekranı - ilgili kişiler	86

Şekil 5-25 : Belge ekranı - ilgili belgeler	87
Şekil 5-26 : Belge ekranı - diğer alanlar	88
Şekil 5-27 : Sayfa Düzenleyicisi	89
Şekil 5-28 : Sayfa Düzenleyicisi - stiller menüsü	94
Şekil 5-29 : Sayfa Düzenleyicisi - hiperlink menüsü	95
Şekil 5-30 : Nesne seçim formu – Video listesi	96
Şekil 5-31 : Video nesnesi formu	97
Şekil 5-32 : Video karakteristik vektörleri formu	99
Şekil 5-33 : Ses nesnesi formu	100
Şekil 5-34 : Ses karakteristik vektörleri formu	101
Şekil 5-35 : Resim nesnesi formu	102
Şekil 5-36 : İçerik nesnesi formu	104
Şekil 5-37 : Web bağlantısı nesne formu	106
Şekil 5-38 : Çokluortam haberler sitesi - ana sayfa	111
Şekil 5-39 : Belge listesi	122
Şekil 5-40 : Belge açıklamaları	124
Şekil 5-41 : Belge sayfası	125
Şekil 5-42 : Video açıklamaları	127
Şekil 5-43 : Ses açıklamaları	128
Şekil 5-44 : Resim açıklamaları	130
Şekil 5-45 : İçerik açıklamaları	131
Şekil 5-46 : İstemcide video oynatılması	132
Şekil 5-47 : Özelleştirilmiş açıklama sorgusu	133
Şekil 5-48 : Özelleştirilmiş belge sorgusu	135
Şekil 5-49 : Özelleştirilmiş ortam sorgusu	137
Şekil 5-50 : Kelime arama	138

TABLolar LİSTESİ

Tablo 4-1 : Nesnelere ve alanları	30
Tablo 4-2 : Nesne tipleri ve OID örnekleri	41
Tablo 4-3 : İlişkisel Şema	44
Tablo 5-1 : Sayfa Düzenleyicisi tip etiketleri	91
Tablo 5-2 : Veri Yayıncısı eylemleri	114
Tablo 5-3 : İlişkisel şema isimleri ile veritabanı nesne isimlerinin eşleştirilmesi	142
Tablo 5-4 : İndeksler	143
Tablo 5-5 : Domenler	148
Tablo 5-6 : Tablolararası ilişkiler	152



ÖZET

Son yıllarda entegre devre ve iletişim teknolojilerindeki ilerleme ile bilgi işlemede daha önce görülmemiş işlem ve iletişim hızlarına erişilebilmiştir. Bunun doğal bir sonucu olarak da önceden işlemesi çok pahalı olan çokluortam veri de bilgi işlemede yaygın olarak kullanılmaya başlandı. Ses, video ve resim gibi çokluortam verisi daha çok gerçek hayatın aynen bilgi işleme hazır hale getirilmesi ile oluşturulur ve sekizli başına en düşük anlamı ifade eden veri tipidir.

Sonraki kuşak bilgisayar sistemleri ise daha da büyük bilgiyi yine daha hassas ve yaygın işleme yeteneğine sahip olacaklardır.

Bu gelişmeler doğrultusunda, çokluortam verisini daha etkin işlemek için yeni bilgisayar algoritmaları ve işlem teknikleri geliştirilmeye başlanmıştır. Yeni teknikler çokluortam verisinin hatalara karşı neredeyse duyarsız olmasından faydalanıp işlem ve/veya iletişim hızını arttırmaya yönelik olarak geliştirilmişlerdir.

Öte yandan Internet dünya çapında ve her geçen gün yayılarak, özellikle uzak mesafeler arasında haberleşmenin en önemli araçlarından biri haline gelmiştir. Internet'in böylesi genişlemesindeki en önemli etmen ise basitleştirilmiş bazı iletişim kurallarına uyabilen her bilgisayara kullanım imkanı vermesidir.

Bu tezde yukarıda sayılan gelişmelerin genel yapısı üzerinde durulup, çokluortam verisinin yönetimi, düzenlenmesi ve WWW üzerinde yayınlanmasını sağlayacak bir "Çokluortam Haberler Veritabanı" geliştirilmiştir.

ABSTRACT

In recent years, innovations in integrated circuit and communication technologies led to unparalleled processing and communication speed. As an inevitable result, multimedia data types that were computationally very expensive before, has been employed widely in computer systems. Those data types largely include data extracted from real world, like images, video and audio, which all required vast storage space, impressive communication and processing speeds and have little meaning per byte.

The next generation of computers systems is predicted to employ even larger data with more precision and thoroughness.

As a matter of fact, new computer algorithms and processing techniques was needed to deal with multimedia data more effectively. Those new techniques made use of the fact that, data is almost immune to errors. Therefore, the trade-off lied on behalf of speed.

On the other hand, Internet has been one of the most important means of communication beyond large distances, widening every other day. The most outstanding factor behind this impressive accumulation, has tended to be its ability to allow any host use Internet with a minimum set of communication protocols.

The thesis aims to examine general nature of those technological advances and to design a system called “Multimedia News Database” to manage, organise and publish multimedia data over the WWW. Consequently, new system is feasible in that it makes use of existing infrastructure, and innovative since it implements core of new technologies.

1. GİRİŞ

Son yıllarda, bilgisayar sistemleri bilgiyi giderek daha çok ve göze hoş gelen şekillerde sunmak için kullanılmaktadırlar. Bilgi iletimi artık sadece metin, resim gibi geleneksel veri tipleri ile değil aynı zamanda video ve ses ile gerçekleştirilmektedir. Çokluortam kullanımı ile bilgi sunumu daha çekici hale gelmiş ve sunulan bilginin değeri artmıştır. Metin ve resim gibi bazı veri tipleri zamandan bağımsız olduğu halde video ve ses gibi bir diğer ortam tiplerinde veri zamanla değişen yani sürekli bir karakter gösterir. (Steinmetz, Nahrstedt 1997)'nin tanımından hareketle, çokluortam uygulamalarını zamana bağlı ve zamandan bağımsız veri tiplerini birarada içeren uygulamalar olarak tanımlayabiliriz.

Değişik çokluortam uygulamaları çözüm getirdikleri sorunlar bakımından değişik karakterler gösterirler. Aşağıda bu tür değişik karakterde çokluortam uygulamalarından örnekler verilmiştir.

İstek üzerine video (Video-On-Demand - VOD) uygulamalarında kullanıcılar bir veya daha fazla uzak sunucudaki video bilgisine erişirler. VOD sistemlerindeki son gelişmeler, büyük oranda tek bir ortamın zamana bağlı sunumunu desteklemekle birlikte, tek ve çokluortam veri tiplerinin bilgisayara alınması, düzenlenmesi ve modellenmesini ihmal etmişlerdir. VOD uygulamaları video verisini sunucudan alıp bu sürekli veriyi tek yönlü olarak istemcilerle ulaştırmakla sunum özellikleri açısından basit karakter gösterirler. Aynı zamanda, VOD uygulamaları zamana bağlı ortam tiplerinin yüksek yoğunluklu verisi yüzünden yüksek performanslı depolama ve ağ sistemleri gerektirirler.

Diğer bir sınıf uygulamalar ise bilgisayar oyunlarıdır. Oyunlar, bilgisayar ve kullanıcı arasındaki etkileşime dayanır ve karmaşık yapılar gösterebilirler. Bir veya daha fazla oyuncu, yerel veya uzaktan oyunlar oynayabilirler. Karşılıklı etkileşime dayalı oyunlar, etkileşimi sağlamak için, dallanma teknikleri veya ileri algoritmalar kullanırlar. (Steinmetz, Nahrstedt 1997). Oyunlar “özel

yapım”dır ve oyunun arkasındaki fikre, arabirimin davranış biçimine, kararlılığa ve dayanıklılığa vurgu yaparlar.

Video konferans uygulamaları video ve ses veri katarlarının gerçek zamanlı iletişimleri ve canlı eşzamanlanmaları ile uğraşırlar. Bu uygulamalar veri katarlarının üretildiği veya dış dünyadan alındığı kaynaktan farklı bir hedef sistemde veriyi kullanmak zorunda olduklarından, eşzamanlılık hata götürmezdir. Video konferans sadece sıkı bağlı, sürekli video ve ses katarları ile kullanılabilir. İleri video konferans teknolojisinin uygulanması büyük ölçüde ağ ve iletişim teknolojisinin gelişimine bağlıdır.

Bilgisayar destekli eğitim, öğrenme programları ve elektronik ansiklopedi gibi çokluortam uygulamaları bireysel karmaşık içerik modelleme ve kullanıcı ihtiyaçlarına göre düzenlenmiş özel sunum teknikleri gerektirirler. Bu, aynı zamanda evden alışveriş, bilet rezervasyonu, ve elektronik ürün katalogları uygulamalarında da geçerlidir. Bu uygulamalar hem ayrık (discrete) hem de sürekli (continuous) veri tiplerini içerirler ve çokluortam verinin etkileşimli sunumu üzerine odaklıdır. Bu uygulamalar ancak çok sayıda kullanıcının çokluortam verisine eriştiği, çok kullanıcıli sistemlerde gereklidirler. Bazı durumlarda ise sistem kullanıcıları bilginin kendisini göndermektense, bilginin kaynağına yönlendirir.

Sonraki uygulamalar, çokluortam sistemlerini farklı ve iddali uygulamalarda kullanır ve geniş çokluortam sistemleri için cesareti arttıran özelliklere sahiptirler. Yine bu uygulamalar çokluortam uygulamalarının (Steinmetz, Nahrstedt 1997)’de gösterildiği eğilimlerle çok kullanıcıli, karşılıklı etkileşimi destekleyen ve çokluortam bilginin iki yönlü akışına imkan veren bir yöne gittiğini göstermektedirler.

“Çokluortam Haberler”, haber dağıtımının bilgisayar alanına uyarlanmasını gösteren soysal bir terimdir. “Çokluortam Haberler”de haberler okunabilir metin olarak ve resim, video vs. gibi herhangi çokluortam veri ile birlikte ağ üzerinden “alıcı”ya gönderilir.

1.1 Çokluortam VTYS Nedir?

(Steinmetz, Nahrstedt 1997) ve (Meyer, Wegener 1995) tanımlarına uyarak çokluortam VTYS'yi (Veritabanı Yönetim Sistemi) şu özelliklere sahip olarak tanımlayacağız.

1. Zamansal, yersel ve anlamsal ilişkileri tanımlayarak değişik ortam tiplerinin biraraya toplanması ile çokluortam kompozisyonları oluşturulmasına imkan tanır.
2. Çokluortam kompozisyonlarının, yeni bir tek ortam tipi olarak sunulması ve düzenlenmesine izin verir.
3. Çokluortam kompozisyonunda kullanımlarında ortamın parçalarının ve ortamın kendisinin bağımsız olarak sunumuna ve düzenlenmesine imkan tanır.
4. Entegre bir çokluortam sistemin seçimlik bir özelliği olarak ortam çevrimine imkan tanır.
5. Çokluortam bilginin etkin yönetimini sağlar.
6. Tek ve çok kullanıcı için uygun bir sistem mimarisi sağlar.

1.2 Arkaplan Veritabanı

Sürekli veri tiplerinin programlama dilleri ve veri modellerine entegre edilmesi ile bir çokluortam sisteminde video ve ses gibi yüksek veri yoğunluklu ortamlar kullanılabilir. Bunun doğal sonucu da sürekli veri tiplerinin etkin yönetiminin gerekliliğidir. Birçok uygulama çokluortam veriyi saklamak için dosya sistemlerini kullanır. Ancak, şu anlaşılmıştır ki, ne geleneksel dosya sistemleri ne de veritabanı sistemleri çokluortam uygulamalarının verisini etkin bir biçimde saklama, değiştirme ve sunma yeteneklerine sahiptirler. (Löhr et al., 1995 and Galchuk et al., 1995). Öte yandan, veritabanı yönetim sistemi kavramları çokluortam verinin tekdüze işlenmesi için mantıklı görünmektedirler. Karmaşık veri yapılarının yönetimi ve çok kullanıcının

veriye erişme ihtiyacı olduğu durumlarda veritabanı yönetim sistemi desteği özel ilgi gerektirmektedir. Çokluortam veriye güvenilir çok kullanıcı desteği sağlamak için, kontrollü fazlalık (controlled redundancy), veri bağımsızlığı (data independence), eşzamanlı erişim kontrolü (concurrency control), tutarlılık (integrity), veri güvenliği (data security), veri koruması (data protection) ve sorgulama dilleri gibi veritabanı özelliklerine ihtiyaç vardır.

Çokluortam verinin veritabanı sistemleri kullanarak etkin yönetimi problemini çözerken şu yaklaşımlar kullanılabilir: varolan veritabanı sisteminin yapılacak eklerle geliştirilmesi, yeni bir veritabanı sistemi geliştirilmesi ve geleneksel veritabanı sistemlerinin ortam destekli bir veritabanı sistemi ile birleştirilmesi. Varolan veritabanı sisteminin eklerle geliştirilmesi, baştan yeni bir veritabanı geliştirilmesinin çok karmaşık olması ve şu ana kadar geliştirilen sistemlerde kullanılan çabayı ziyan etmesi düşünülerek, en iyi seçenek gibi görünmektedir. Çünkü, geleneksel veritabanı sistemlerinin ortam destekli bir veritabanı sistemi ile birleştirilmesi için de veritabanı sistemi servislerinin ortam destekleyen servislerle entegrasyonuna ihtiyaç vardır.

Sürekli verinin modellenmesi değişik şekillerde gerçekleştirilebilir. Geleneksel veritabanı modellerinde, video ve ses gibi sürekli verinin kullanılması için doğrudan bir destek yoktur. Bu tip modellerde en kolay çözüm, veritabanında büyük verinin saklanmasına yarayan uzun alanlara veya BLOB'lara (Binary Large Object) dayanır. Bu çözümlerin kötü tarafı veritabanının bilgiyi anlamına önem vermeden saklaması ve böylece uygulamaya ek servisler verememesidir. Örneğin, bir uygulama veritabanı sisteminden sadece sekizli katarını alır ve (MPEG-Motion Pictures Expert Group veya JPEG-Joint Photographers Expert Group gibi) uygun veri formatına ve uygun işlem tekniğine kendi başına karar vermek zorundadır. Açıktır ki, böyle bir sistem artık uygulamaya veri bağımsızlığını sağlayamaz.

Nesne tabanlı veritabanı sistemleri, veritabanı sisteminin çokluortam veritabanı sistemine geliştirilmesi için uygun bir temel teşkil ediyor görülmektedirler. Nesne tabanlı veritabanı yönetim sistemlerinde soyut veri tipleri (abstract data

types - ADT) kullanılarak sürekli veri modellenenebilir. Yani örneğin bir video veri katarının modellenmesi için özelliklere (attributes) ve yordamlara (methods) sahip sınıflar (classes) tanımlanabilir. Böylece video veriye ait özel anlamların veritabanında saklanması sağlanır. Örneğin sınıf I-, B- ve P- çerçeveleri ile bir MPEG videonun yapısını yansıtabilir veya sınıf, veri biçimine özel detayları saklayıp, uygulamaya bir çeşit standart biçimde (tabi uygun çevrim seçenekleri ile birlikte) sunabilir. Sürekli verinin daha ileri işlenmesine izin verilse de, bu yolla hala, veritabanı sistemi sürekli verinin özelliklerini anlamamaktadır. Örneğin, veritabanı sistemi hala video ve ses gibi zaman bağımlı veri için tampon ve nesne yönetimi sağlayamamaktadır. Bunun sebebi, sürekli verinin bu şekilde modellenmesinin veri modelinin sadece bir uygulaması olmasıdır. Modelleme temelleri veri modeline veya sisteme entegre edilmemiştir.

Sonuç olarak, çoklu ortam verinin entegre yönetimine doğru bir diğer yaklaşım sisteme ait (built-in) sürekli veri tiplerinin nesne tabanlı bir veri modeline entegrasyonudur. Ya veritabanı MPEG1 tabanlı video, MPEG2 tabanlı video, JPEG tabanlı video, 16-bit ses, 8-bit ses vs. için sisteme ait veri tipleri ile genişletilir veya birçok sürekli veri tipi için ortak bir taban oluşturacak daha sosyal bir veri tipi sisteme dahil edilir. Tek tek sisteme ait veri tiplerinin aksine, sosyal veri tipi sürekli tampon ve nesne yönetimi ile daha kolay entegre edilip veritabanının zaman bağımlı veriyi kendi özelliklerine göre işlemesine izin verir. Sosyal veri tipi sadece veri modeli ile sağlanan sınıflar, özellikler ve yordamlar gibi basit modelleme elemanları kullanılarak çeşitli tiplerde sürekli verinin modellenmesine izin verir. (Jagadish and Story, 1993)'de sürekli veri için bir sosyal sisteme ait veri tipi, video ve ses gibi özel ortamların modellenmesinde taban teşkil etmektedir. Sonuçta ortaya çıkan tiplere "ortam tipleri" adı verilebilir. Sürekli verinin ortam tipleri ile modellenmesi işlenmemiş ortam verisi ile birlikte veri biçimi, kodlama ve sıkıştırma teknikleri, uzunluk, saniyedeki örnek sayısı, örnek boyu, çerçeve boyu, renk tablosu vs. gibi çeşitli meta-veri (meta-data) de içerir. Bundan başka uygulamalar sahneler, videonun içeriğinin tanımı gibi sürekli bilginin içeriğine

ilişkin ek bilginin de modellenmesini isteyebilirler. Bu tip yapısal bilgilere ek olarak, veri tipleri veri tipine özel işlemlerin tanımlanmasına da izin verirler. Veri tiplerinin, nesnelere (instances) yaratılması (örneğin tarayıcıdan alınarak), yaratılmış nesnelere değiştirilmesi (örneğin ses nesnesinin uzunluğunun ve kalitesinin değiştirilmesi) veya nesnelere sunumu (örneğin bir video nun oynatılması) için gerekli işlemler buna örnektirler.

Veri tiplerinin sistemde yaratılması ile veri nesnelere elde edilirler. Veritabanında saklanan bir veri nesnesi tek video veya ses gibi tek bir ortamı temsil eder. Çokluortam kompozisyonları çoğu zaman özel yersel veya zamansal ilişkilerle birbirlerine bağlı birden fazla veri nesnesini içerirler. Yani veri nesnelere çokluortam kompozisyonlarının oluşturulmasındaki temel yapı taşlarıdır.

1.3 Çokluortam Dokümanlarının Modellenmesi

Entegre bir çokluortam sistemden beklenen özelliklerden biri ortam tiplerinin sınırsız kombinasyonları ile çokluortam kompozisyonlarını düzenleme ve sunum olanağı sağlamasıdır. Bu da böyle çokluortam kompozisyonlarını destekleyen modelleri gerektirir. Bu modellere *çokluortam doküman modelleri* denir. Doküman modellerinin en çok bilinen ve en sık kullanılanları SGML (Standard Generalised Markup Language), HyTime (Hypermedia/Time-based Structuring Language), MHEG (Multimedia and Hypermedia Information Encoding Expert Group), ODA (Office Document Architecture), ve OCPN (Object Composition Petri Net)'dir. Bir çokluortam kompozisyonum bir çokluortam sistemde somutlaştırılmış haline *çokluortam doküman* denir. Bu somutlaştırma değişik ortam nesnelere zamansal, yersel ve anlamsal bağımlılıklarla yeni bir ortam nesnesi içinde yeniden düzenlenmelerini içerir. Genel olarak çokluortam kompozisyonları sentetiklerdir. Yani ortam nesnelere sunumdan önce veritabanından seçilip düzenlenmişlerdir. Canlı video konferans gibi bazı uygulamalarda ise kompozisyonlar gerçek zamanlı olarak oluşturulur. Veritabanında saklanan ortam nesnelere çokluortam

kompozisyonun temelini oluřtururlar. Her veritabanı nesnesi kendisinin sunumu için gerekli bilgiyi içerir.. Her ortam nesnesi resim, video veya ses gibi bir ortam tipine aittir. Öte yandan, veri nesnelerinin modellenmeleri çokluortam kompozisyonları için yeterli bilgi üretmezler. Çokluortam kompozisyonları oluřturulurken, kullanılan ortam nesnelere parça parça da kullanılabilirler. Örneğin, bir çokluortam kompozisyon iki saatlik bir videonun sadece birkaç saniyesini veya bir resmin sadece detayını içerebilir. Sonuç olarak, özel sunum bilgisi ilgili ortam nesnelerindeki orjinal bilginin yanısıra seçilen ortam parçalarını da tanımlar. Bundan başka, sunum bilgisi sadece veritabanındaki ortam nesnesinin modellenmesi için gerekli bilgileri değil değil ancak sunum için gerekli ortam bilgilerini de içerir. Bu bilgiler örneğin ortama özel oynatma veya çalma hızı, resimlerin dönme miktarı, oynatma veya çalmaya başlama ve bitiş noktaları gibi bilgiler olabilir.

Çokluortam dokümanları çokluortam kompozisyonda kullanılan ortam nesnelerini içermeli veya onlara referans vermelidirler. Ortam nesnelerinin ve çokluortam dokümanlarının ayrı ayrı saklanması mantıklı görünmektedir. Veritabanındaki ortam nesnelere sistem bazında tek olan nesne belirteçleri (Object Identifier - OID) belirlenebilirler. O zaman, çokluortam dokümanlar her ortam nesnesi için sunum bilgisini ve ortam nesnesine bir referans içerirler. Nesne belirteçleri veritabanındaki nesnelere ile çokluortam dokümanları arasındaki ilişkiyi kurarlar. Böylece ortam nesnelere dokümanlardan bağımsız hale getirilmiş olur ve ortam nesnelerinin çokluortam kompozisyonlarında kullanımlarından bağımsız olarak düzenlenmesine olanak tanır. Aynı zamanda, bu ayırım bir ortam nesnesinin değişik çokluortam kompozisyonlarında ve değişik kalite seviyelerinde kullanımına izin vererek fazlalığı (redundancy) önler ve böylece veritabanı yönetim sisteminin gerekliliklerini yerine getirmiş olur.

Çokluortam dokümanları ile ortam nesnelerinin ayrı ayrı yönetimi, öte yandan, çokluortam dokümanları ile ortam nesnelerinin uyumluluğunu sağlama açısından zorluklara yol açar. Bir ortam nesnesi veritabanından silindiğine, o

nesneye yapılan tüm referanslar geçersiz hale gelir. Ortam nesnelerinde yapılan değişiklikler çokluortam dokümanlarındaki sunum bilgisini geçersiz hale getirebilir. Örneğin, bir kompozisyonda tamamı kullanılan 20 sn'lik bir video nesnesi daha sonra 10 sn'ye indirilirse kompozisyondaki sunum bilgisi geçersiz olur. Kompozisyonlarda kullanılan nesnelere yapılan değişiklikler, kompozisyonlardaki kullanımlarına göre kontrol edilmelidirler.

Çokluortam kompozisyonda kullanılan her ortam nesnesi için bir sunum nesnesi hem veritabanındaki nesneye referansı, hem de sunum için gerekli bilgiyi içerir. Her sunum nesnesinin tuttuğu bilgi bir ortamın sunumu için yeterlidir. Bir çokluortam doküman ise sunum nesnelerinin yerleştirilmesi ile ilgili bilgileri içerir. Bir çokluortam kompozisyon yaratılırken, sunum nesnelere zaman ekseninde dizilebilir ve/veya başka sunum nesnelere zamanda veya yerde bağlı olabilirler. Bunun yanında etkileşimli sunum nesnelere tuşlar, kayak çubuklar veya metin giriş alanları ile etkileşen ortam nesnelerini temsil ederler. Bir veya daha fazla sunum nesnesine zaman veya yer boyutlarında bağlı olabildikleri gibi, buna ek olarak, kullanıcının etkileşimi sunucunda uygulanacak bir eylem (action) tanımları da içerirler.

1.4 Zamana Bağlı İlişkiler

Sunum nesnelerinin bir çokluortam sunumda algılanabilmeleri için hem sürekli hem de ayrık ortamlar bir süre için sunulmalıdır. Sürekli ortamlar zamanla değer değiştirirken, ayrık ortamların değerleri sabit kalır. Sunum nesnelere zaman eksenine dizilirler. Her sunum nesnesi için en azından bir başlangıç belki bir de son belirlenmelidir. Sonuç olarak, bir sunumun zamanda ilerlemesinin uygun bir modeli seçilmeli ve çokluortam dokümanında saklanmalıdır. Bunun için iki temel yaklaşım önerilmiştir:

- Zamanın nokta bazlı modellenmesi: Nokta bazlı modelde zaman sürekli zaman içinde noktalarla temsil edilir. Zamandaki bir nokta sıfır uzunluktadır ve zaman ekseninde bir noktaya karşılık gelir. Zamandaki noktalar tamamen

sıralıdır. Zamanda herhangi iki nokta için önce ($<$), sonra ($>$) veya eşit ($=$) ilişkilerinden biri muhakkak doğrudur. Bu yöntem zamanın az sayıda zamansal ilişki ile basit bir modelidir.

- Zamanın aralık bazında modellenmesi: Zaman aralıkları zamansal modellemenin temelini oluştururlar. Bu model herhangi iki zaman aralığı için, iki aralık çakışmalar, iki aralık aynı zamanda başlar gibi 13 temel ilişki sunar.

Nokta bazlı model ve aralık bazlı model, her $[a; b]$ zaman aralığı zamanda iki noktaya eşit olduğu (yani başlangıç noktası, a ve bitiş noktası, b) sürece birbirlerine denk modellerdir. Bir çokluortam sunumun her bir ortamın sunum sürelerini temsil eden zaman aralıkları ile modellenmesi en uygun olarak görülmektedir.

Yarı açık zaman aralıkları ve belirsiz aralık bağıntıları sözkonusu olduğunda ise aralık bazlı modelleme de yetmeyebilir. Sunum nesnelere örneğin etkileşim için sunum başlangıç zamanı belirleyip bitiş zamanı belirlememiş olabilirler. İki sürekli sunum nesnesi aynı anda başlayıp birbirlerinin sonlarından habersiz olabilirler. Sonuç olarak yarı açık veya belirsiz zaman aralıklarını da içeren ileri zamansal modeller önerilmiştir (Galchuk et al., 1995 and Rothermel and Wahl 1994).

Çokluortam kompozisyonu yaratılırken sunum sırasında kullanıcı etkileşimi ile veya otomatik olarak sunum görüntüsünün değişmesi istenebilir. Bu durumda zamansal sıranın modeli zamanda bir noktada bir tek ortam sunumunun nasıl değişeceğinin belirlenmesine izin vermelidir. Bunun yanında, bir sunum nesnesinde sesin alçalıp yükselmesi gibi devamlı değişimler varsa zaman aralığı ve değişimin miktarı da belirlenmelidir. Kısaca, bir ortamın sunumunu değiştirecek parametrelere işlemlerin modellenmesi gerekmektedir.

1.5 Yersel İlişkiler

Görsel sunum nesnelere buldukları uzayda yerlerini belirleyen ek bilgiye ihtiyaç duyarlar. Sunum nesnelere arasında yersel ilişkiler de tanımlanabilir. Eğer birden fazla sunum nesnesi ekranda görüntüleniyorsa, sunum nesnelere üst üste gelebilirler. Görsel sunum nesnelere diğere sunum nesnelereinin sağında, solunda, üstünde, altında vs. olabilirler. Bu ilişki aynı zamanda çakışan bölgede XOR işlemi de olabilir.

1.6 Etkileşimli Sunum Nesnelere

Çokluortam sunumları kullanıcı ile etkileşime izin vermelidirler. Bu yolla, kullanıcı sunumu diğerek yollardan etkileyen diğerek sunum nesnelereini kullanmak isteyebilir. Aşağıda bir kullanıcının çokluortam sunum sistemi ile diğerek etkileşim grupları verilmiştir.

- Boyutlama işlemlere bir çokluortam sunumun görünüşünü diğektirirler. Örneğinin video penceresinin boyunun diğektirilmesi veya bir resmin renk tablosunun diğektirilmesi.
- Sinema işlemlere veya standart işlemlere bir videonun kumanda panelindeki işlemlere karşılık gelirler. Standart işlemlere arasında başlat, dur, durakla, devam et, ileri sar, geri sar vs. sayılabilir. Bunlar çokluortam sunumun hızı ve yönü ile ilişkili işlemlerdir.
- Karar işlemlere çokluortam sunumun yönüne karar verirler. Bu işlemlere sunumun belirli noktalarında, kullanıcının sunumun nasıl devam edeceğine karar vermesini sağlayan etkileşimli sunum nesnelere ile sağlanırlar.

Boyutlama işlemlereinin çokluortam dokümanlarında uygulanması zorunlu diğektir. Zaten tek bir sunum nesnesi üzerinde tanımlandıklarından, sunum nesnelere veritabanındaki ilgili ortamın modellenmesine bağılı olarak belirli boyutlama işlemlereine izin verebilirler.

Standart işlemlere çokluortam sunumun tüm zamansal gidişini üzerinde etkili olurlar. Özel olarak çokluortam doküman içinde saklanmalarına gerek yoktur.

Daha çok, çokluortam kompozisyonların veri tipleri ile birlikte modellenmesi gereken işlemlerdir.

Karar işlemlerinin ise çokluortam dokümanlarda belirlenmesi zorunludur. Bir çokluortam sunumun tasarımcısı veya kullanıcısı tarafından tanımlanırlar. Bir etkileşimli sunum nesnesinin hangi diğer sunum nesnelere referans ettiği ve etkileşimin çokluortam sunum üzerinde nasıl bir etkisi olacağı belirlenmelidir. Karar işlemleri bir grup sunum nesnesinin yanısıra tüm sunumu etkileyen işlemlerle ilgili olarak da düşünülebilirler. Bu ilgi isteyerek modellenmiş veya etkileşime bağlı işlemde gizli olabilir.

Çokluortam kompozisyonda etkileşimli sunum nesnelерinin zamansal düzenlenmesi özel ilgi gerektirir. Etkileşimli sunum nesnelерinin aktif oldukları zaman aralığı her zaman belli değildir. Bu durumda her zaman kapalı zaman aralıkları ile ifade edilemezler. Kapalı zaman aralıkları etkileşimin sadece bir süre için kullanıcının insiyatifinde olduğu durumlarda yeterlidirler. Fakat, etkileşimli sunum nesneleri kullanıcı onları seçene kadar da sunum içinde olabilirler.

1.7 Eşzamanlama İlişkileri

Eşzamanlama ilişkileri, sunumun zamansal ilerlemesi, sunum nesnelерinin yersel düzenlenmesi veya ilgili ortam nesnelерinin içeriği ile ilgili olabilirler. Sentetik eşzamanlama ilişkileri çokluortam sunumdan önce çokluortam dokümanda tanımlanmışlardır.

Bir diğer seçenek ise genellikle kişiden kişiye iletişim servislerinde olan canlı eşzamanlamadır. Bu tip bir servislere bir örnek video konferanstır.

- Zaman eşzamanlaması: Sentetik zaman eşzamanlaması ortam içi eşzamanlama ve ortamlararası eşzamanlama olarak ikiye ayrılabilir.
 - ◆ Ortam içi eşzamanlama: Her sürekli sunum nesnesi için, sunum verisi saniyedeki kare sayısı gibi istenen oynatma hızı bilgisini içerir.

Eşzamanlamayı sağlayan bir sunum elemanı belirlenen hızı ve devamlılığı sağlamak zorundadır.

- ◆ **Ortamlararası eşzamanlama:** Bir çokluortam kompozisyonun zamansal modeli, sunum nesnelere arasındaki zamansal ilişkileri de içermelidir. Bu ilişkiler aynı zamanda ayrık sunum nesnelere için ortamlararası eşzamanlamayı sağlamak için de kullanılabilirler (örneğin iki sunumu paralel başlatmak veya bitirmek için). Sürekli ortamlararası eşzamanlama sadece zamana bağlı sunum nesnelere için geçerlidir. Bu eşzamanlama ile sunu hızındaki sapmalar sürekli olarak kontrol edilip düzeltilir.
- **Yersel eşzamanlama:** Yersel ilişkiler zamansal düzenlemelerle birleştirilebilirler. Örneğin, bir müzik çalarken bir resim sürekli olarak sol üst köşeden sağ alta doğru hareket edebilir.

1.8 Servis Kalitesi Parametreleri

Bir çokluortam kompozisyon için servis kalitesi (Quality of Service - QOS) parametreleri tanımlanabilir. Çokluortam sunumları sunum kalitesindeki bozuklukları bir dereceye kadar tolare edebilirler. İnsan duyuları sürekli ortamın oynatılmasında hafif bozuklukları algılayamazlar. Bu yüzden ki çokluortam sunumları esnek gerçek zaman gerekliliklerine sahiptirler. Sentetik kompozisyonun gerçek sunum kalitesinin, tanımlanan sunum kalitesinden izin verilen sapma miktarları QOS parametreleri ile belirlenir. Değişik parametreler uygulamaların, çokluortam sistemin değişik bölümlerinden beklentilerini yansıtırlar. Aynı zamanda QOS parametreleri sistemin etkinliğini de tanımlarlar. Burada, sadece bireysel sunum nesnelere ve sunum nesnelere arasındaki ilişkiler ile ilgili QOS parametreleri ele alınmıştır. İşletim sistemi ve iletişim gibi görevlerle ilgilenmediğimiz için, QOS parametrelerini sadece çokluortam uygulamaları açısından görmekteyiz. Uygulama seviyesindeki QOS

parametreleri, daha alt sistem seviyelerindeki QOS modelleri yardımı ile bir sunum nesnesi tarafından sağlanmalıdır.

Tek sunum nesnelere için zaman bağımsız QOS parametreleri bir resmin veya bir videonun çözünürlüğü veya renk derinliği gibi sunumun kalitesi ile ilgilidirler. Sürekli sunum nesnelere için fazladan bir izin verilen eşzamanlama sapması (jitter) parametresi, verilen bir çalma veya oynatma hızı için belirlenen ortamlararası eşzamanlama kısıtlarının sağlanmasına imkan verir. Sunum nesnelere arası ilişkiler için olan QOS parametreleri genellikle nesnelere arasındaki zamansal ilişkiler ile ilgilidirler. Kayma parametresi belirlenen ortamlararası eşzamanlama ile karşılaştırıldığında, hızdaki birikmiş farkı temsil eder.



2. ÖNCEKİ ÇALIŞMALAR

Son yıllarda, çokluortam verinin yönetimi ile ilgili çok ve yoğun çalışmalar yapılmıştır. Bu çalışmalar hedefleri itibarı ile iki grupta toplanabilirler.

İlk grup çalışmalarda çokluortam sistemler varolan sistemlerin uzantıları olarak geliştirilmişlerdir. Bu çalışmalarda araştırmacılar uygulamalarında fizibiliteye önem vermişlerdir. İkinci grupta ise her bileşeni yeni amaca yönelik olarak yeniden tasarlanmış olarak tamamen yeni geliştirilen sistemler vardır.

Nesne tabanlı kurallar ve fikirler çokluortam sistemler için doğal bir uygulama içerirler. Entegre bir sistem geliştirilebilmesi için verinin dayanıklı depolanması da nesne tabanlı olmalıdır. Bu fikir, nesne tabanlı veritabanlarının geliştirilmesine ve çokluortam sistemlerin bu veritabanları üzerinde uygulanmasına yol açmıştır. (IBM, 1993) ve (Lau 1994)'de IBM SOM'daki (System Object Model) nesne depolama ve kuralları anlatılmaktadır. (Sessions 1996) konuyu daha da ileri götürüp, sonraki nesil nesne tabanlı veritabanları olan dayanıklı nesne depolarından (persistent object stores) bahseder. (Schöne, 1996)'da ise çokluortam veritabanı için sosyal tip sistemi geliştirilmiştir.

Buna rağmen, nesne tabanlı veritabanları üzerinde yapılan çalışmalar ilişkisel teorinin güçlü bir matematiksel temeli, geniş kullanım alanı ve geniş uygulama tecrübesi olduğu gerçeğini değiştirmemektedir. Sonuç olarak, çokluortam sistemler üzerine yapılan bazı çalışmalar da nesne tabanlı verinin depolanması için ilişkisel veritabanlarının kullanılmasını öngörmüşlerdir. (Agarwal et al., 1994) ve (Agarwal et al., 1995)'da nesne tabanlı programlama ile ilişkisel veritabanlarının kullanılması ele alınır. (Gottemukkala et al., 1996) ise ilişkisel veritabanlarında nesnelerin saklanması sorununa değinilir. (Ramanathan and Koduri 1996)'da açıklanan araştırma ilişkisel veritabanı sisteminin nesne tabanlı bir sisteme çevrimi ile ilgilidir. Bu çevrim işlemi, aslında ilişkisel sistemin kendisine karşılık gelen nesne tabanlı sisteme bire bir dönüştürülmesini içerir. (Hodges and Ramanathan, 1996); varolan verinin aktarımı için ilişkisel sisteme nesne tabanlı erişimin sağlanmasından

(Ramanathan, 1996) ve nesne tabanlı sistemin prototipinin hazırlanmasından (Ramanathan and Koduri 1996) ibarettir.

Video ve ses gibi zamana bağılı verinin eşzamanlanması çokluortam sistemin en hayati özelliklerinden biridir. (Galchuk et al. 1995) çokluortam verinin zaman yönetimi için yeni bir zamansal model önerir. (Rothermel and Wahl 1994) ise çokluortam sistemlerinde sürekli zamanın ayrıklaştırılarak kullanılması konusuna değinir. (Zhang) çokluortam sistemlerde eşzamanlı sunum yönetimi ile ilgilenir.

Çokluortam veri genellikle çok yoğun olduğundan, çokluortam dokümanların organizasyonu performans ve kullanım kolaylığını da etkileyen önemli bir faktördür. (Rakow and Böhm, 1994) çokluortam veri için meta-verinin özelliklerini inceler. (Wei, 1994) Çokluortam verinin varolan sistemlere entegrasyonu için çokluortam meta-veriyi varolan meta-veri standartları içine yerleştirir. (Jagadish and Story, 1993) de ise çokluortam dokümanlar için yeni bir veri modeli geliştirilmiştir. Yine, (Recker et al., 1995)'da çokluortam veri için diğer ortam tipleri üzerinde durulmuştur.

Öte yandan, içerik tanıma (content recognition) da çokluortam veri erişiminin bir diğer önemli konusudur. (Docherty, 1993)'de otomatik içerik erişimi için yeni bir sistem tasarlanmıştır.

Çokluortam veri erişimi, özel geliştirilmiş ortam sunucuların ziyadesiyle faydalanır. (Goyal et al., 1994)'da çokluortam sunucuların geliştirilmesi tartışılır. Depolama sunucularına ilişkin özel bir araştırma (Gemmell et al., 1994)'da yayınlanmıştır. (Rao et al., 1995) ise çokluortam sunucular için sunucu-itimli (server-push) ve istemci-çekimli (client-pull) teknolojileri karşılaştırılır. Yeni teknolojilerin çokluortam sistemlere uyarlanması, performans açısından (Gettys et al., 1997)'da incelenmiştir. (Mogul et al., 1997) ise HTTP'deki (Hypertext Transfer Protocol) gecikme (latency) problemi üzerinde durur.

Bir haberler veritabanını uygulayan örnek bir çokluortam veritabanı hala Alberta Üniversitesi (Kanada)'da geliştirilmektedir. Geliştirme süreci mantıksal tasarım (ElMadani et al., 1994) ve doküman girişi desteği (ElMadani, 1996) gibi aşamaları içermektedir. (Chakravarthy et al., 1993) tıp araştırmaları için bir çokluortam veritabanı sisteminin uygulanmasını anlatır. Yine bir diğer çokluortam veritabanı örneği (Gollapudi and Zhang a) ve (Gollapudi and Zhang b)'de verilmiştir.

(Löhr et al. 1995) ve (Jajodia and Subrahmanian, 1996) çokluortam sistemlerde genel eğilimler ve gelişmelerle ilgili referans bilgisi verirler. (Berra et al., 1996) çokluortam sistemler için tasarım ve uygulama stratejilerini tartışır. Çokluortam veri erişimi (DeVries, 1995)'de anlatılmıştır. Çokluortam ve resim veritabanları (Baker and Khoshafian, 1996)'da ele alınmıştır. (Boll et al., 1996) ise çokluortam verinin depolanması ve sunumu sorunlarını tartışır..

Bu çalışma aynı zamanda bilgisayar haberleşmesi ile ilgili uluslararası standartları da kullanır. (RFC1866) HTML (Hypertext Mark-up Language) 2.0 standartını açıklar. (RFC1867) HTML'de form-tabanlı geri yüklemeyi (upload) tartışır. (RFC1945) ve (RFC2068) HTTP 1.0 ve HTTP 1.1 iletişim kuralları standartlarıdır. (RFC2045), (RFC2046), (RFC2047), (RFC2048) ve (RFC2049)'un tümü MIME (Multipurpose Internet Mail Extensions) standardını oluştururlar. Hemen tüm diğer internet standartları için temel oluşturan (RFC822) ise internet metin mesajlarının biçimini belirler.

3. AMAÇ

Bu proje bir haberler veritabanı geliştirme amacındadır. Tasarlanan sistemin temel özellikleri arasında

- çoklu ortam verinin depolanması ve düzenlenmesi,
- haber bilgisinin doküman bazında yönetimi,
- haber tanımının yayınlanan her tür bilgiyi içerecek şekilde geliştirilmesi,
- bilginin internet ve/veya internetten yayınlanması,
- yeniden konfigürasyonla esnek kullanıma izin vermesi

sayılabilir.

Tasarım sürecinin anahtar noktaları ise,

- nesne tabanlı tasarım ve uygulama programı,
- tasarlanan sistemin ilişkisel bir veritabanı sistemi ile kullanımı için adaptasyonu,
- nesne belirteçleri (OID) ile nesne tanımlanması,
- doküman tabanlı organizasyon,
- sistemde yapılacak değişikliklerle yeni özelliklerin kazandırılmasının kolaylaştırılması amacı ile uygulanmayan kısımlar için de destek içeren tasarım,

olarak belirlenmişlerdir.

3.1 Çokluortam Haberler Veritabanı Sisteminin Temel Fonksiyonları

3.1.1 Sorgulama

Çokluortam Haberler Sistemi standart WWW (World Wide Web) gezginleri kullanılarak erişilebilir. Bir *Bilgi Yayıncı* istenen bilgiyi veritabanından alıp uygun içerik ve biçim bilgisini HTTP 1.0 iletişim kuralları kullanılarak kullanıcıya gönderen bir ara modül olarak çalışır. Kullanıcının gözünde veritabanı sadece dokümanlar ve çokluortam nesnelere içerir. Kullanıcı sadece dokümanlara ve tek tek çokluortam nesnelere erişebilir.

Çokluortam Haberler, kullanıcıların dokümanları veya çokluortam nesnelere açıklama bilgileri ile sorgulamasına, doküman sayfalarında kelime aranmasına ve çokluortam dokümanlara bir dosya sistemindeki klasör ve dosyalar gibi hiyerarşik bir biçimde erişilmesine imkan tanır.

3.1.2 Veri Girişi

Veritabanına veri girişi ayrı bir program kullanılarak yapılır. Veri Girişi programı kullanıcıların yeni bir çokluortam dokümanı yaratmalarına ve resim, video, metin gibi çokluortam nesnelere ile bu dokümanları süslemelerine izin verir. Sistem dokümanları ve çokluortam nesnelere bağımsız yönetmesi ile aynı çokluortam nesnenin birden fazla dokümanda kullanımına izin verir. Böylece veri fazlalığı da önlenmiş olur. Ancak, çokluortam nesnelere atomik olarak tasarlanmışlardır. Yani bir dokümanda çokluortam nesnenin sadece bir bölümünün kullanılmasına izin verilmez. Bu özellik çokluortam sistemin nesnelere biçimlerinden bağımsızlığını sağlar.

Veri girişi sırasında doküman metni yazılabilir veya dışarıdan yüklenebilir. Daha sonra kullanıcı uygun sayfalara çokluortam nesnelere ekler, uygunsa bu nesnelere özelliklerini düzenler ve dokümanı Çokluortam Haberler Veritabanı'nda saklar.

Sistem çokluortam nesnelere için 3 tip depolamaya izin verir.

- Nesnelerin veritabanındaki büyük alanlarda (BLOB) saklanması.
- Nesnelerin işletim sisteminde saklanması. (Bu seçenek disk, CD-ROM ve jukebox saklama şekillerini de içerir)
- Nesnelerin çevrim dışı saklanması (örneğin arşivde).



4. TASARIM

4.1 Nesne Tabanlı Tasarım

Çokluortam Haberler Veritabanı Sistemi nesne tabanlı olarak tasarlanmıştır. Bu yolla sistemdeki her bağımsız veri tipi bir nesne tanımının içine konulmuştur. Sonuç olarak nesne tabanlı sistem şunları içerir.

- bir grup nesne,
- bu nesnelerin dizileri,
- nesneler arası bağlar veya ilişkiler.

Nesne tabanlı tasarım daha sonra “Nesne Düzleme” (Object Flattening) adı verilen bir teknikle ilişki karşılığın çevrilir.

4.2 Sınıf Açıklamaları

Sınıf Adı	Açıklama Adı	Tanım
Object	OID	Nesne belirteci (Object Identification Number – OID). Bu numara tipinde bağımsız olarak sistemdeki tüm nesneler tarafından miras alınır..
	Name	Nesnenin adı
Folder	Parent Folder	Klasörün bağlı olduğu üst klasörün OID'i. Belgeler ve klasörler dosya sistemindeki dizin ve dosyalar gibi organize edilmişlerdir.
Document	Header	Belge başlığı

Sınıf Adı	Açıklama Adı	Tanım
	Copyright	Telif hakkını elinde bulunduran kişi. Bu açıklama alanı Person nesnesinin OID'ini içerir.
	Source	Haberin kaynağı. Haber kaynağı ulusal veya uluslararası bir haber ajansı, veya bir muhabir olabilir. Bu açıklama alanı Person nesnesinin OID'ini içerir.
	Where appeared	Haberin yayınlandığı yer. Bu açıklama alanı Person nesnesinin OID'ini içerir.
	When appeared	Yayınlanma tarihi.
	Start of news	Haberin başlangıcı. TV veya radyo haberleri için bu alan saat, dakika, saniye biçiminde zaman bilgisini, gazete haberleri için ise sayfa, satır, sütun bilgilerini içerir.
	End of news	Haberin sonu. TV veya radyo haberleri için bu alan saat, dakika, saniye biçiminde zaman bilgisini, gazete haberleri için ise sayfa, satır, sütun bilgilerini içerir.
	Type of News	Haber tipi (bkz. Ekler – Tanım Listeleri – Haber Tipleri). Bu açıklama alanı “News Type” nesnesinin OID'ini içerir.

Sınıf Adı	Açıklama Adı	Tanım
	News Subject	Haber olan olayın konusu. (bkz. Ekler – Tanım Listeleri - Haber Konuları). Bu açıklama alanı “News Subject” nesnesinin OID’ini içerir.
	Importance	Haberin önem seviyesi (bkz. Ekler – Tanım Listeleri – Önem Seviyeleri). Bu açıklama alanı Importance nesnesinin OID’ini içerir.
	Abstract	Haberin kısa özeti
	Keywords	Haberi tanımlayan anahtar kelimeler.
	Location	Haberin geçtiği veya ilgili olduğu coğrafi yer. Bu bir coğrafi yer nesnesinin OID bilgisidir.
	Storage	Orjinal haberin arşiv bilgisi. Bu bir arşiv nesnesi OID bilgisidir.
	Related News	Bu belgenin (haber) mantıksal olarak ilişkili olduğu diğer haberlerin listesi. Aslında sıfır veya daha fazla belge nesnesinin oluşturduğu dizidir.
	Subjects	İlgili kişilerin ve haberdeki rollerinin bulunduğu özne nesnelerinin OID’lerinin dizisidir.

Sınıf Adı	Açıklama Adı	Tanım
	Pages	Belgenin içerdiği sayfaların OID'lerinin dizisidir.
Page	Page number	Sayfanın doküman içindeki numarası.
	Owner	Sayfanın ait olduğu belge.
	Text	Sayfa metnini içeren Metin nesnesinin OID'i.
Collection of Multimedia Objects	OID	Çokluortam nesnesinin OID'i.
Collection of Visible Objects	Offset	Nesnenin hangi karakterden sonra yerleştirileceğini gösteren sayı.
Collection of Temporal Objects		(Bu soyut sınıf kendine ait bir özellik içermez)
Collection of Videos		(Bu soyut sınıf kendine ait bir özellik içermez)
Collection of Audios		(Bu soyut sınıf kendine ait bir özellik içermez)
Collection of Images		(Bu soyut sınıf kendine ait bir özellik içermez)
Storage	Media Type	Ortam nesnesinin içerdiği orjinal ortamın depolanma tipi. (bkz. Ekler – Tanım Listeleri – Saklama Türleri)

Sınıf Adı	Açıklama Adı	Tanım
	Room	Orjinal dokümanın arşiv adresinin ilk kırılımı.
	Section	Orjinal dokümanın arşiv adresinin ikinci kırılımı.
	Shelf	Orjinal dokümanın arşiv adresinin üçüncü kırılımı.
	Row	Orjinal dokümanın arşiv adresinin dördüncü kırılımı.
	Column	Orjinal dokümanın arşiv adresinin beşinci kırılımı.
	Order Number	Orjinal dokümanın arşiv adresinin altıncı kırılımı.
Content Based	Content	Veriyi saklayan içerik nesnesinin OID'i..
Multimedia Object	Keywords	İçeriği tanımlayan anahtar kelimeler.
	Multimedia Object Annotation	
Multimedia Object Annotation		(Bu soyut sınıf kendine ait bir özellik içermez)
Visible	Width	Görsel nesnenin genişliği.
	Height	Görsel nesnenin boyu
Temporal	Length	Ortam nesnesinin süresi.

Sınıf Adı	Açıklama Adı	Tanım
Text	Attributes	<p>Sayfadaki değişik ortam nesnesi referanslarının dizisi. Bu referanslar şunlar olabilir.</p> <ul style="list-style-type: none"> • HTML etiketi içeren bir Stil. • Video, Resim, Ses, İçerik gibi bir çokluortam nesnesi. • Bir Java applet veya yazı içi resim için dosya yolu. • WWW bağlantısı için bir Web Bağlantısı nesnesi.
	Text	Biçimsiz metin.
Image	Feature vector	Resmin karakteristik vektörü. Bu alan sistem için gelecekte uygulanabilecek bir uzantı olarak görülen İçerik Tabanlı Erişim (CBR - Content Based Retrieval) için kullanılabilir..
Audio	Collection of feature vectors	Ses içinde değişik zaman aralıklarına ait karakteristik vektörler dizisi. Bu alan sistem için gelecekte uygulanabilecek bir uzantı olarak görülen İçerik Tabanlı Erişim (CBR – Content Based Retrieval) için kullanılabilir.

Sınıf Adı	Açıklama Adı	Tanım
Video	Collection of feature vectors	Ses içinde değişik zaman aralıklarına ait karakteristik vektörler dizisi. Bu alan sistemde gelecekte uygulanabilecek bir uzantı olarak görülen İçerik Tabanlı Erişim (CBR - Content Based Retrieval) için kullanılabilir.
In-line	File Name	Görelî yol ismi ile birlikte yazı içi nesnenin dosya adı. Bu dosya belge sayfası ile birlikte gönderilecektir . Dosya içeriği bir resim veya bir ikili Java applet (*.class) bilgisi olabilir.
Content	Storage	Orjinal ortamın arşiv bilgisi. Bu bir arşiv nesnesi OID bilgisidir.
	File path.	Dosya sisteminde saklanan nesne bilgisinin tam yolu.
	BLOB	Veritabanında nesnenin saklandığı alan.
	MIME Type	Bu bir MIME Tipi nesnesi OID bilgisidir ve nesne HTTP ile istemciye gönderilirken HTTP başlık bilgisi "Content Type" alanına yazılır. MIME tipi bilgisi "<type>/<sub-type>" biçiminde olmalıdır.

Sınıf Adı	Açıklama Adı	Tanım
Geographical Location	Domestic/Foreign	Coğrafi yerin ülkesi yurtiçi mi yoksa yurt dışı mı olduğunu belirler. Eğer yurtiçi ise "Country" alanına gerek yoktur.
	Country	Coğrafi yerin ülkesi
	State	Coğrafi yerin ili
	District	Coğrafi yerin ilçesi
	Region	Coğrafi yerin bölgesi. Bölge bilgisi coğrafi yer ülke, il, ilçe üçlemesiyle tanımlanamıyorsa kullanılır.
Role		Gerçek ve tüzel kişilerin haber içindeki rollerinin tanımlandığı nesnedir.
Who		Gerçek ve tüzel kişiler ile şirketler için taban teşkil eden sosyal nesnedir.
Person		Gerçek ve tüzel kişi tanımları saklayan nesnedir.
Company		Şirket tanımları saklayan nesnedir.
Text Attribute	Start offset	Metin içinde açıklamanın başladığı yer.
	Length	Açıklamanın boyu.
	In-line Object	Yazı içi nesnenin görelî yolu. Sadece açıklama bir yazı içi nesneye bağ ise kullanılır.

Sınıf Adı	Açıklama Adı	Tanım
	Linked Object	Bağlanmış ortam nesnesinin OID'i. Sadece açıklama veritabanındaki bir ortam nesnesine veya bir Web Bağlantısı nesnesine bağ ise kullanılır.
	Style	Bağlanmış Stil nesnesinin OID'i. Sadece açıklama bir Stil nesnesine bağ ise kullanılır.
Related Persons	Subject	Rol alan veya ilgili Kişi nesnesinin OID'i.
	Role	Alınan rolü belirten Rol nesnesinin OID'i. (bkz. Ekler – Tanım Listeleri – Rol Türleri).
Feature Vector	Start Time	Karakteristik vektörü çıkarılan zaman aralığının başlangıcı.
	End Time	Karakteristik vektörü çıkarılan zaman aralığının sonu.
	Feature Vector	Karakteristik vektör. Bu alan için önceden tanımlı bir fonksiyon olmadığından istenildiği gibi girilebilir.
Audio Feature Vector		
Video Feature Vector	Frame Number	Karakteristik vektörü çıkarılan video çerçevesinin numarası.

Sınıf Adı	Açıklama Adı	Tanım
Web Link	URL	Nesnenin gösterdiği kaynağın numarası.
Style	Font Name	Sayfa Düzenleyicisi'nde Stili gösterirken kullanılacak font.
	Font Size	Sayfa Düzenleyicisi'nde Stili gösterirken kullanılacak fontun boyu.
	Font Bold	Sayfa Düzenleyicisi'nde Stili gösterirken kullanılacak fontun koyu olup olmayacağı.
	Font Italic	Sayfa Düzenleyicisi'nde Stili gösterirken kullanılacak fontun italik olup olmayacağı.
	Font Underline	Sayfa Düzenleyicisi'nde Stili gösterirken kullanılacak fontun altı çizili olup olmayacağı.
	Font Strikethrough	Sayfa Düzenleyicisi'nde Stili gösterirken kullanılacak fontun üstü çizili olup olmayacağı.
	Font Colour	Sayfa Düzenleyicisi'nde Stili gösterirken kullanılacak fontun rengi.
	Start Tag	HTML'de Stili gösterirken kullanılacak başlangıç etiketi.
	End Tag	HTML'de Stili gösterirken kullanılacak bitiş etiketi.

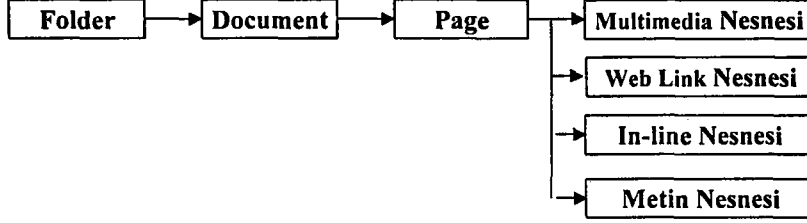
Sınıf Adı	Açıklama Adı	Tanım
News Type		Haber tiplerinin tanımlandığı nesne. Örneğin “televizyon haberi” veya “gazete ilanı”
News Subject		Haber konularının tanımlandığı nesne. Örneğin “ekonomi” veya “suç”.
Importance		Haberlerin önem derecelerinin tanımlandığı nesne. Örneğin “önemli” veya “flaş”.
MIME Type	Type	<type>/<sub-type> biçiminde MIME tipi
Media Type		Arşiv nesnelerinin saklandıkları ortam türlerinin tanımlandığı nesne. Örneğin “kağıt”, “manyetik ortam” veya “optik ortam”.

Tablo 4-1 : Nesneler ve alanları

4.3 Nesne Mimarisi

4.3.1 Kap hiyerarşisi

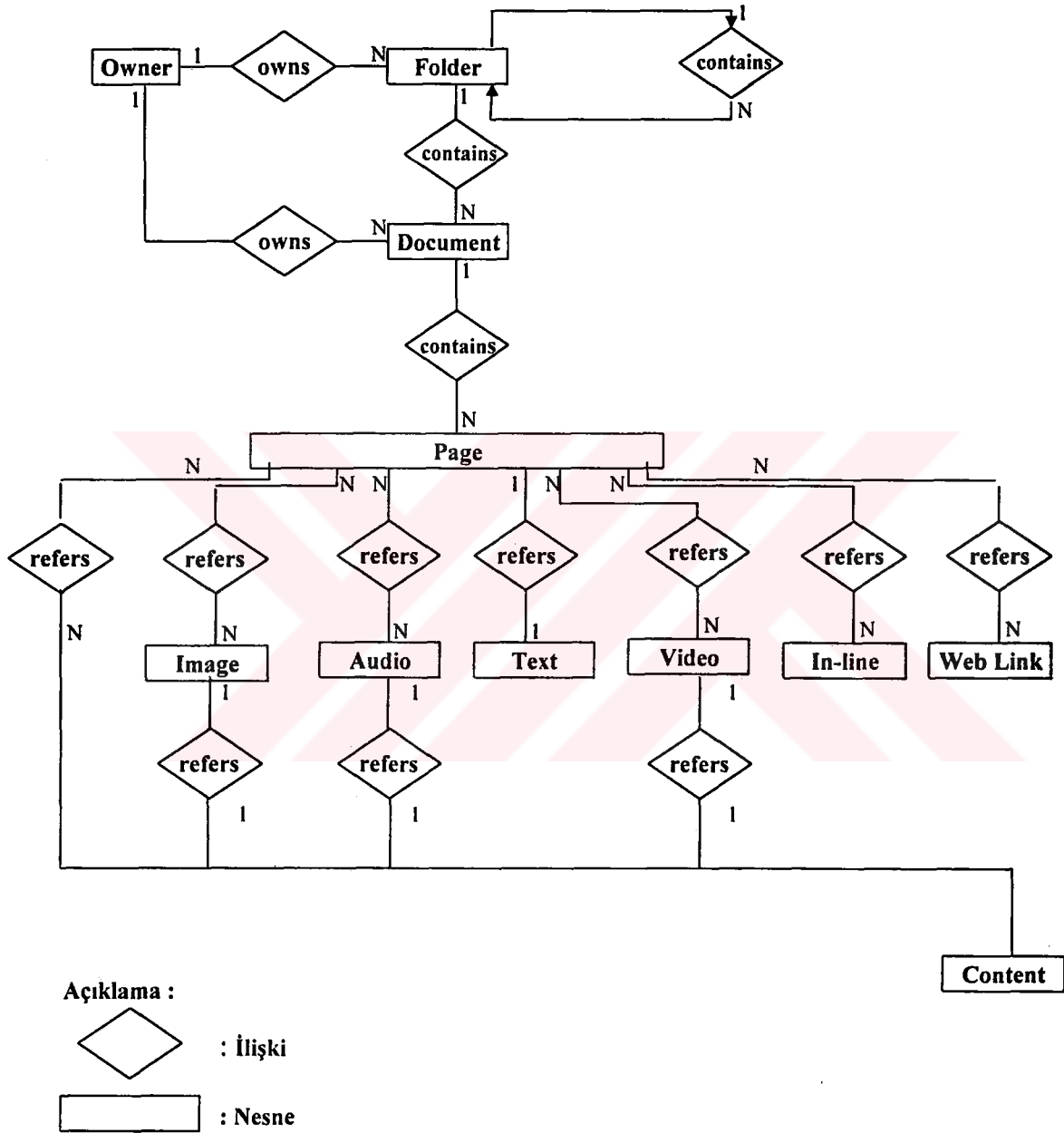
Sistem kap hiyerarşisi aşağıdaki şekilde açıklanmıştır..



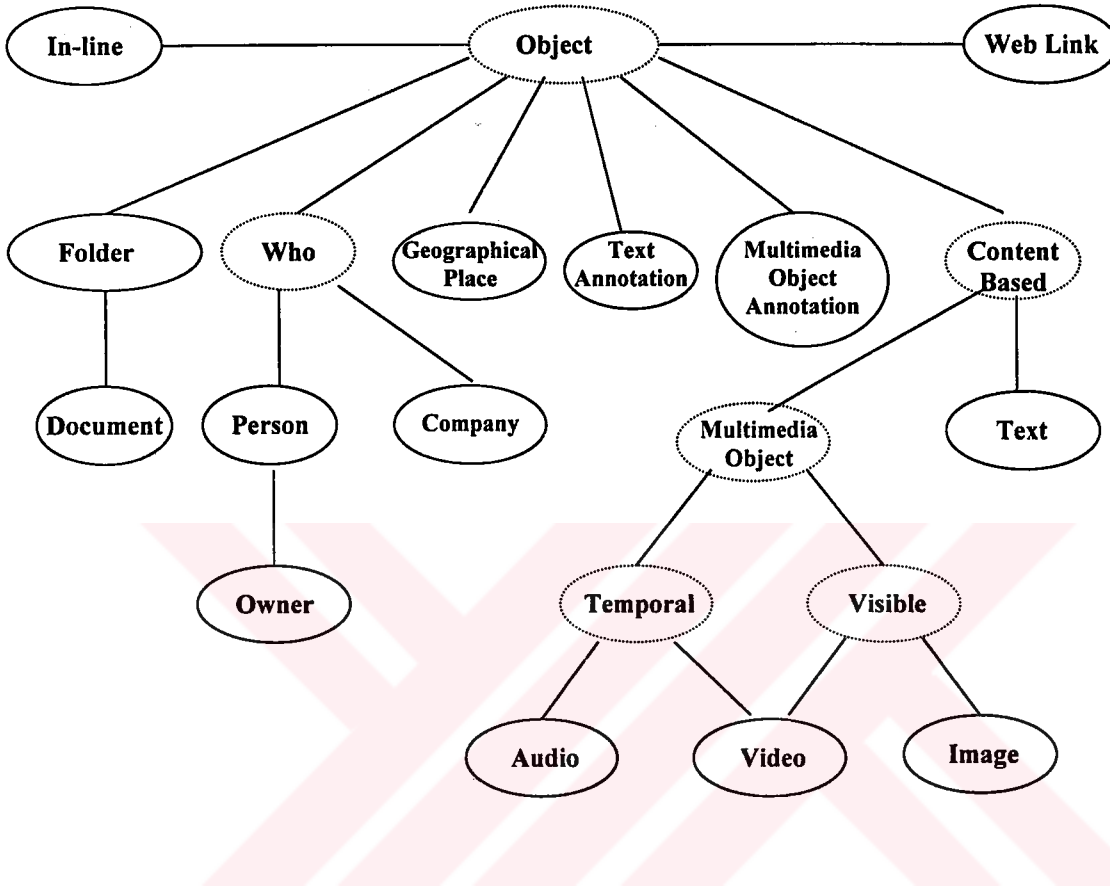
Şekil 4-1 : Kap hiyerarşisi

Yukarıdaki şekilde görüldüğü üzere klasörler belgeleri, belgeler sayfaları ve sayfalar da Metin, In-line, Web Bağlantısı ve Multimedia nesnelerini içerirler.

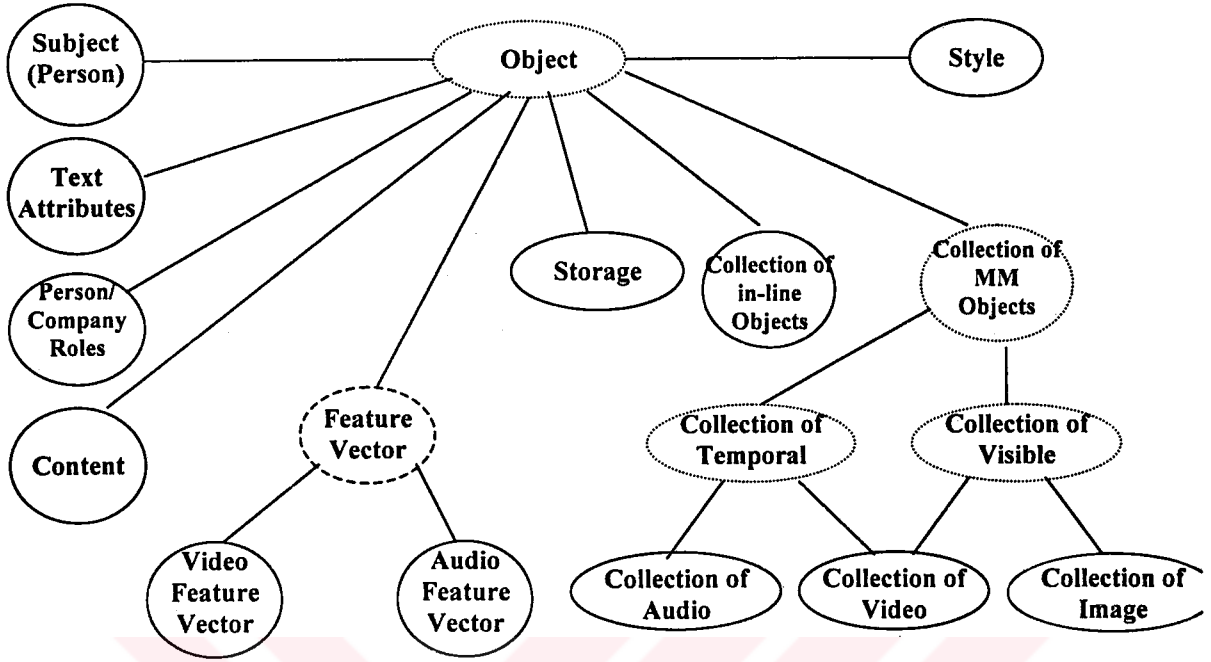
4.3.2 Nesne hiyerarşisi



Şekil 4-2 : Nesneler arasındaki ilişkiler



Şekil 4-3 : Nesne Hiyerarşisi



Açıklama :

○ : Somut nesne sınıfı

○ : Soyut nesne sınıfı

Şekil 4-4 : Nesne hiyerarşisi (devam)

Sistemdeki tüm nesnelere ortak OID (Object Identifier) ve Name alanlarını içeren Object nesnesinden türetilmişlerdir. Sonuç olarak sistemdeki tüm sürekli nesnelere ve bu alanları miras alırlar.

Document nesnesi, *ParentFolder* ortak alanını içeren *Folder* nesnesinden türetilmiştir.

Page nesnesi, bazı fazladan alanlarla birlikte mantıksal bir belge sayfasını içerir. *Document* 'a ilişkin bazı alanlarla birlikte *Text* nesnesine bir referans da yine *Page* nesnesi içinde yer alır.

Who nesnesi, *Person* ve *Company* nesnelere için bir baz oluşturur. *Object* nesnesinin *Name* alanı *Company* nesnelere için şirket adını *Person* nesnelere için kişi adını içerir. *Person* nesnesinden türetilen *Owner* nesnesinin miras aldığı alanlar dışında kendine ait hiçbir alanı yoktur. Sadece anlaşılabilirlik için hiyerarşiye eklenmiştir.

Style nesnelere mantıksal HTML etiketlerinin zengin metin karakter biçimleri ile eşleştirilmesini sağlarlar. Böylece, Veri Giriş Programı'nın Sayfa Düzenleyicisi'nde HTML etiketlerinin nasıl biçimlendiğine, ilgili *Style* nesnesi karar verir.

Text Attributes dizisi *Text* nesnelere için açıklamaları içerir. Bu açıklamalar, tamamen biçimsiz metnin tüm biçimleme özelliklerini içerirler.

Storage nesnesi Çokluortam Haberler Sistemi'ne nesne olarak girilen kaynakların orjinal arşiv bilgilerini içerir. *Document* ve *Content* nesnelere için arşiv bilgileri ve dolayısı ile *Storage* nesnelere tanımlanabilir.

Web Link nesnesi URL'si (Unified Resource Locator) ile bulunabilen bir İnternet kaynağını temsil eder. *Web Link* nesnelere *Page* nesnelere içindeki *Metin*'lerin değişik bölümlerini dış dünyaya bağlamak için kullanılırlar.

In-line nesnelere HTML destekli, yazı içi kaynaklarla ilgili –ki bunlar yazı içi resim ve Java applet'lerdir- URL'leri temsil ederler.

Hiyerarşinin en önemli bölümü *Content Based* nesne hiyerarşisidir. *Content Based* de *Object* 'ten türetilmiştir ve tüm çokluortam nesnelere için bir baz teşkil eder. *Content Based* nesnesi, nesnenin içeriğini temsil eden *Content* alanını içerir ve daha sonra, *Text*, *In-line* ve *Multimedia Object* nesnelere özelleşir. *Text* nesnesi kendine ait (yani miras alınmamış) *Text Attributes* alanını tanımlar ki bu alan bir *Text Attribute* nesne dizisidir ve metne ait tüm biçim bilgisini içerir. Öte yandan, *In-line* nesnesi, *Page* nesnesi ile tanımlanan sayfa ile birlikte dağıtılacak Java applet veya yazı içi resim dosyasının adını içerir.

Metnin biçimden tamamen ayrılması ile yapılan bu düzenleme, ayrıca, metin verisinin anlaşılmasını gerektiren Full Text Retrieval (FTR) gibi uygulamaları da kolaylaştırır.

Content Based nesnesinden türetilen ikinci nesne ise *Multimedia* nesnesidir. *Multimedia* nesnesi *Multimedia Object Annotation*, *Description* ve *Keywords* alanlarını içerir ve *Temporal* ve *Visible* alt nesnelere için bir baz teşkil eder. *Temporal* nesnesi, daha sonra, *Audio* ve *Video* nesnelere türetmek için kullanılacağından, bu nesnelere ortak alanı olan *Length* (süre) alanını içerir. *Visible* nesnesi, öte yandan, ekranda görünebilen çokluortam nesnelere (yani *Video* ve *Image*) için baz nesne olarak kullanılır. *Visible* nesnesi, *Width* ve *Height* alanlarını miras bırakır.

Hiyerarşinin sonunda, *Audio*, *Video* ve *Image* nesnelere gelir. Önceden belirtildiği gibi, bu nesnelere her biri CBR'a (Content Based Retrieval) imkan sağlamak için gerekli bilgiyi, yani karakteristik vektörleri içerirler. *Image* nesnesi tek bir karakteristik vektör içerir. Öte yandan, *Video* nesnesi seçilen çerçevelere veya zaman aralıklarına ait, *Audio* nesnesi ise sadece seçilen zaman aralıklarına ait karakteristik vektörler dizisi içerir.

Nesne veri tipi sistemi, ayrıca, referans nesnelere adı verilen nesnelere de içerirler. Bu nesnelere *Collection of Multimedia Objects*, kökünden türetiler ve hangi yolla ya da hangi özel alanlarla bir *Page* 'in bir *Content Based* nesnesine referans verdiğini belirlerler. Bu nesnelere önceden açıklanan sunum nesnelere

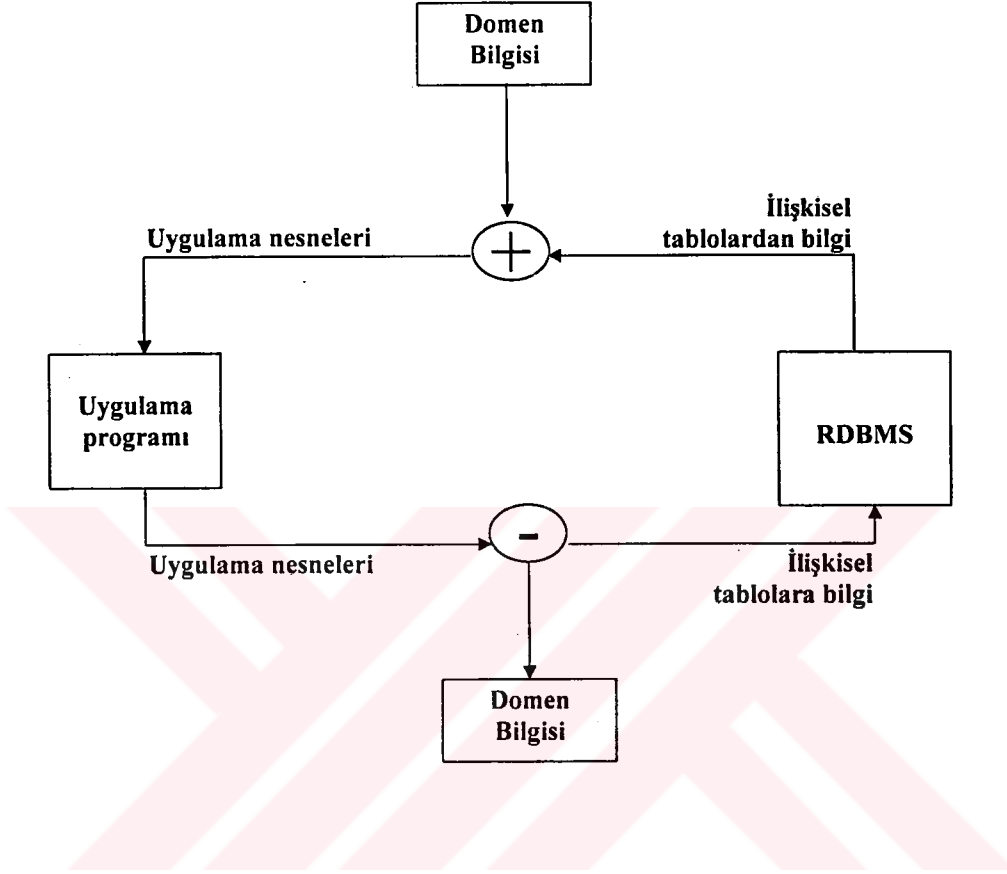
olarak ta tanımlanabilirler. Bir *Page* 'in bir *Video* veya bir *Audio* 'ya referans verdiği durumda referans nesnesi *Collection Of Temporal* soyut sınıfından türer. Bir *Page* 'in bir *Video* veya bir *Image* 'a referans verdiği durumda ise referans nesnesi *Collection of Visible* soyut sınıfında tanımlanan ve nesnenin sayfa metni içindeki yerini belirleyen *Offset Within Text* alanını içerirler. Hiyerarşinin en alt dalları kendilerine ait alanları olmayan ancak nesnelere yaratılabilen (yani somut nesnelere) *Collection Of Video*, *Collection Of Audio* ve *Collection Of Image* sınıflarını içerirler..

4.4 Nesne Tabanlı Yapıdan İlişkisel Yapıya Geçiş

Nesne tabanlı yapının ilişkisel yapıya çevrimi, nesne tabanlı şemadan domen bilgisinin çıkarılıp düzleştirilerek ilişkisel yapıya geçişi gerektirir. Bu süreçte, aksi belirtilmediği sürece aşağıdaki kurallara bağlı kalınmıştır.

1. Her somut sınıf bir tabloya karşılık gelir.
2. Her yaratılmış nesne bir tablo satırına karşılık gelir.
3. Her veri alanı (nesne özelliği) bir tablo alanına karşılık gelir.
4. Nesne dizileri alanlardan birinde sahip nesnenin OID'ini yabancı anahtar olarak saklayan bir tabloya karşılık gelir.
5. Miras alınan alanlar (özellikler), miras alınmayanlar gibi, tablolara alanlar olarak geçirilirler. Sonuç olarak, çevrimden sonra miras bırakma kaybolmuştur.
6. Soyut sınıfların hiçbiri, zaten diğer sınıflara özelliklerini miras bırakmak için kullanıldıklarından, ilişkisel şemaya çevrilmezler.
7. Nesne dizilerinin içindeki her eleman için kullanılan OID'ler kaldırılmışlardır. Çünkü zaten verinin tablo satırları halinde saklandığı ilişkisel tablolarda, her kaydın bir kayıt ID'si vardır.

Aşağıdaki şekilde ileri (nesne tabanlıdan ilişkisel'e) ve geri (ilişkisel'den nesne tabanlıya) çevrim süreci görülmektedir.



Şekil 4-5 : Nesne tabanlı'dan ilişkisel'e çevrim süreci

Çevrim sürecinde, ayrıca, bazı nesnelerin ve/veya alanların birleştirilip sıkıştırılması uygulamanın kolaylaştırılması açısından gerekli görülmüştür. Bu sebeple, sonuçta ortaya çıkan tabloların sayısı, önceden tasarlanan nesnelerin sayısından azdır.

Bu biçimde yapılan birleştirmeler arasında zaten tasarımı daha anlaşılır kılmak için kullanılan *Multimedia Object Annotation* nesnesinin kaldırılması ve iki alanı da bir çoka-çok ilişkinin ara tablosunun alanları olarak uygulanan *Subject* nesnesinin kaldırılması gösterilebilir.

4.4.1 Nesne düzleme

Nesne Düzleme nesne tabanlı'dan ilişkisel'e çevrim kurallarına bağlı kalınarak uyarlanmış bir tekniktir ve Çokluortam Haberler Veritabanı Sistemi tasarımı, ilişkisel karşılığına çevirmekte kullanılmıştır.

Nesne düzleme için sistemdeki nesnelere birincil ve ikincil nesnelere olarak ikiye ayrılmışlardır. Nesne tabanlı'dan ilişkisel'e çevrim sürecinde, birincil ve ikincil nesnelere için değişik çevrim teknikleri kullanılır.

Birincil nesnelere kendi başına bir anlam ifade eden bilgiyi içeren nesnelere dir. Birincil nesnelere de kendi içlerinde iki tip halinde incelenirler. İlk tip birincil nesnelere kendi içinde bir anlam ifade eden ve sistemde anlamsal olarak asıl saklanması gereken bilgiyi ihtiva eden, gerçek dünya varlıklarını temsil ederler. *Folder, Document, Page, Video, Audio, Image, Content, Text, Web Link* ve *Storage* nesnelere bunlardandır. Bu tip nesnelere, ayrıca, sistemde ayrı ayrı ve tek başlarına erişilebilmelidirler. İkinci tipte ise, tanımlamalar için kullanılan nesnelere gelir. Tanımlama nesnelere bir sınıf alanı için, sınırlı bir domen tanımlarlar. *Document* nesnesinin "News Type" alanı, örneğin, sadece önceden tanımlı listeden değerler içerebilir. Sonuç olarak, *News Type* nesnelere bir "OID" ve bir "Name" alanına sahip olacak şekilde tasarlanmıştır ve *Document* nesnesi "News Type" alanında bir *News Type* nesnesine referans saklar.

Öte yandan, ikincil nesnelere sadece ilgili bir nesne ile birlikte anlam ifade eden bilgi içerirler. Böylece bu nesnelere ayrı olarak sorgulanmalarına gerek yoktur. *Video Feature Vectors* sistemdeki bu tür nesnelere bir örnektir. *Audio Feature Vectors, Subject ve Related Documents* sistemdeki diğer ikincil nesnelere de.

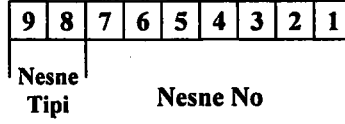
Veritabanı sistemi de nesnelere OID'leri verilerek bulunmasına izin verecek yapıda tasarlanmıştır.

Nesne tabanlı'dan ilişkisel'e çevrim işlemi aşağıdaki yolla gerçekleştirilmektedir,

1. Birincil nesnelere, OID'leri birincil anahtar kabul eden tablolara dönüştürülmüştür. Örnek olarak, OID alanı birincil anahtar olarak *DOCUMENTS* tablosuna dönüşen *Document* nesnesi gösterilebilir.
2. İkincil nesnelere, bağlı oldukları ana nesnelere OID'leri ile kendilerine ait bir veya daha çok veri alanını birincil anahtar kabul eden tablolara dönüştürülmüştür. Kendileri ile bağlı oldukları ana nesne arasındaki bire-çok ilişkisinin, çok tarafını oluştururlar.
3. Soyut sınıflardan miras kalan tüm alanlar tablo tanımlamalarının içine gömülmüştür.
4. *Content* nesnesinden türetilen *Video, Audio ve Image* nesnelere sadece kendi tanımladıkları alanları içeren ve miras kalan alanları içermeyen tablolara dönüştürülmüştür. *Content* nesnesinin kendisi OID alanlı bir tabloya dönüştürüldüğünden, *Video, Audio ve Image* nesnelere *Content* alanlarında *Content* nesnesine referans verirler. Böylece bu nesnelere *Content* nesnesine bire-bir-veya-sıfır ilişki kurulmuş olur. Bu yaklaşımda, *Content* nesnesinin somut bir sınıf olması ve çevrimden sonra bir tabloya dönüştüğü gerçeği kullanılmıştır.
5. Bir nesne tüm alanları ile (miras kalan veya tanımlanan) tabloya dönüştürüldüğünden soyut nesnelere ilişkisel yapıya çevrilmezler.

4.5 Nesne Belirteçlerinin Yapısı

Nesne belirteçleri (Object Identity - OID) ilgili tabloların (nesnelerin) birincil anahtarları olarak kullanılırlar. Bir nesne belirteci, en anlamlı iki rakamı tip bilgisini içeren, 9 rakamlı sayılar olarak tasarlanmışlardır. Aşağıda nesne belirtecinin yapısı görülmektedir.



Şekil 4-6 : Nesne Belirtecinin (OID) Yapısı

Nesne No'lar (OID) tüm sistem çapında, aynı tipte nesnelere arasında, yani tablonun satırları arasında, tektirler. Nesne belirteçleri ise, sistemdeki tüm nesnelere arasında tektirler. Yani, sistemdeki herhangi bir birincil nesne, sadece nesne belirteci ile belirlenebilir.

Nesne tipleri ve karşılık gelen OID örnekleri aşağıdaki tabloda verilmiştir.

Tablo 4-2 : Nesne tipleri ve OID örnekleri

Nesne	Nesne Tipi Öneki
Folder	1
Document	2
Page	3
Video	4
Audio	5

Nesne	Nesne Tipi Öneki
Image	6
Reserved	7
Text	8
Content	9
Location	10
Role	11
Person	12
Reserved	13
Storage	14
News Type	15
News Subject	16
Importance	17
Text Annotation	18
Style	19
Media Type	20
MIME Type	21
Storage	22
Reserved	23
Reserved	24
Web Link	25

Çokluortam Haberler Sistemi veritabanı yaratılan her nesne için, nesne tipine göre ve otomatik olarak bir nesne belirteci üretir.



4.6 İlişkisel Şema

Çevrim sürecinin bir sonucu olarak, sınıflar ve alanlar ilişkisel karşılıklarına çevrilmişlerdir. Sonuçta oluşturulan ilişkisel şema, tablo ve alan tanımları ile aşağıdaki listede görülmektedir.

Tablo 4-3 : İlişkisel Şema

Tablo Adı	Alan Adı	Açıklama
Folder	OID	Nesne Belirteci (Object Identification Number – OID).
	Name	Klasörün ismi
	ParentFolder	Bulunulan klasör. Bu tablonun “OID” alanından alınmış yabancı anahtardır.
Document	OID	Nesne Belirteci (Object Identification Number – OID).
	Name	Belgenin ismi
	ParentFolder	Bulunulan klasör. Bu <i>FOLDER</i> tablosunun “OID” alanından alınmış yabancı anahtardır. Belgeler ve klasörler dosya sistemindeki dizin ve dosyalar gibi organize edilmişlerdir.
	Header	Belge başlığı

Tablo Adı	Alan Adı	Açıklama
	Abstract	Belgenin kısa özeti
	Keywords	Belgedeki olayı tanımlayan anahtar kelimeler.
	Storage	Orjinal haberin arşiv bilgisi. Bu alan <i>STORAGE</i> tablosunun OID' alanını yabancı anahtar olarak içerir.
	Copyright	Telif hakkını elinde bulunduran kişi veya kuruluş. Bu alan <i>PERSON</i> tablosunun OID' alanını yabancı anahtar olarak içerir.
	Source	Haberin kaynağı. Haber kaynağı ulusal veya uluslararası bir haber ajansı, veya bir muhabir olabilir. Bu alan <i>PERSON</i> tablosunun OID' alanını yabancı anahtar olarak içerir.
	WhereAppeared	Haberin yayınlandığı yer. Bu alan <i>PERSON</i> tablosunun OID' alanını yabancı anahtar olarak içerir.
	WhenAppeared	Yayınlanma tarihi.

Tablo Adı	Alan Adı	Açıklama
	StartOfNews	Haberin başlangıcı. TV veya radyo haberleri için bu alan saat, dakika, saniye biçiminde zaman bilgisini, gazete haberleri için ise sayfa, satır, sütun bilgilerini içerir.
	EndOfNews	Haberin sonu. TV veya radyo haberleri için bu alan saat, dakika, saniye biçiminde zaman bilgisini, gazete haberleri için ise sayfa, satır, sütun bilgilerini içerir.
	NewsType	Haber tipi (bkz. Ekler – Tanım Listeleri – Haber Tipleri). Bu alan <i>NEWS TYPE</i> tablosunun OID alanını yabancı anahtar olarak içerir.
	Location	Haberin geçtiği veya ilgili olduğu coğrafi yer. Bu alan <i>GEOGRAPHICAL LOCATION</i> (coğrafi yer) tablosunun OID' alanını yabancı anahtar olarak içerir..

Tablo Adı	Alan Adı	Açıklama
	NewsSubject	Haber olan olayın konusu. (bkz. Ekler – Tanım Listeleri - Haber Konuları). Bu alan <i>NEWS SUBJECT</i> tablosunun OID alanını yabancı anahtar olarak içerir.
	Importance	Haberin önem seviyesi (bkz. Ekler – Tanım Listeleri – Önem Seviyeleri). Bu alan <i>IMPORTANCE</i> tablosunun OID alanını yabancı anahtar olarak içerir.
Page	OID	Nesne Belirteci (Object Identification Number – OID).
	Name	Sayfanın adı
	PageNo	Sayfanın belge içindeki sayfa numarası.
	Owner	Sayfanın ait olduğu belge. Bu alan <i>DOCUMENT</i> tablosunun OID alanını yabancı anahtar olarak içerir.
	Text	Sayfanın metnini tutan nesneye referans. Bu alan <i>TEXT</i> tablosunun OID alanını yabancı anahtar olarak içerir.
Text	OID	Nesne Belirteci (Object Identification Number – OID).
	Name	Nesnenin ismi
	Text Data	Biçimlenmemiş sayfa metni. Bu alan bir BLOB alanıdır.

Tablo Adı	Alan Adı	Açıklama
Text Attributes	ObjectOID	Bağlanmış <i>IMAGE</i> , <i>VIDEO</i> , <i>AUDIO</i> , <i>CONTENT</i> veya <i>WEB LINK</i> tablosunun OID'i. Sadece, açıklama veritabanındaki bir ortam nesnesine veya bir <i>WEB LINK</i> (Web Bağlantısı) nesnesine bağ ise kullanılır.
	Style	Bağlanmış <i>STIL</i> tablosunun OID'i. Sadece, açıklama bir <i>Stil</i> nesnesine bağ ise kullanılır.
	Path	Yazı içi nesnenin görelî yolu. Sadece, açıklama bir yazı içi nesneye bağ ise kullanılır.
	Start	Metin içinde açıklamanın başladığı yer.
	Length	Açıklamanın boyu
Storage	OID	Nesne Belirteci (Object Identification Number – OID).
	Room	Orjinal dokümanın arşiv adresinin ilk kırılımı.
	Section	Orjinal dokümanın arşiv adresinin ikinci kırılımı.
	Shelf	Orjinal dokümanın arşiv adresinin üçüncü kırılımı.

Tablo Adı	Alan Adı	Açıklama
	Row	Orjinal dokümanın arşiv adresinin dördüncü kırılımı.
	Column	Orjinal dokümanın arşiv adresinin beşinci kırılımı.
	OrderNo	Orjinal dokümanın arşiv adresinin altıncı kırılımı.
	MediaType	Ortam nesnesini içeren orjinal haberin depolanma tipi. (bkz. Ekler – Tanım Listeleri – Saklama Türleri). Bu alan <i>MEDIA TYPES</i> tablosunun OID alanını yabancı anahtar olarak içerir.
Image	OID	Nesne Belirteci (Object Identification Number – OID).
	Content	İlgili içerik nesnesi. Resmin kendisi de içerik nesnesinin içinde saklanır. Bu alan <i>CONTENT</i> tablosunun OID alanını yabancı anahtar olarak içerir.
	Name	Resmin adı
	Keywords	Resmi tanımlayan anahtar kelimeler
	Width	Resmin genişliği
	Height	Resmin boyu

Tablo Adı	Alan Adı	Açıklama
Audio	OID	Nesne Belirteci (Object Identification Number – OID).
	Content	İlgili içerik nesnesi. Sesin kendisi de içerik nesnesinin içinde saklanır. Bu alan <i>CONTENT</i> tablosunun OID alanını yabancı anahtar olarak içerir.
	Name	Sesin adı
	Keywords	Sesi tanımlayan anahtar kelimeler
	Length	Sesin süresi
Audio Feature Vector	Audio OID	Karakteristik vektörü çıkartılan ses nesnesi. Bu alan <i>AUDIO</i> tablosunun OID alanını yabancı anahtar olarak içerir.
	Start	Karakteristik vektörü çıkarılan zaman aralığının başlangıcı.
	End	Karakteristik vektörü çıkarılan zaman aralığının sonu.
	FeatureVector	Karakteristik vektör. Bu alan için önceden tanımlı bir fonksiyon olmadığından istenildiği gibi girilebilir.

Tablo Adı	Alan Adı	Açıklama
Video	OID	Nesne Belirteci (Object Identification Number – OID).
	Content	İlgili içerik nesnesi. Videonun kendisi de içerik nesnesinin içinde saklanır. Bu alan <i>CONTENT</i> tablosunun OID alanını yabancı anahtar olarak içerir.
	Name	Videonun adı
	Keywords	Videoyu tanımlayan anahtar kelimeler.
	Width	Videonun genişliği
	Height	Videonun boyu
	Length	Videonun süresi
Video Feature Vector	Video OID	Karakteristik vektörü çıkartılan video nesnesi. Bu alan <i>VIDEO</i> tablosunun OID alanını yabancı anahtar olarak içerir.
	Frame	Karakteristik vektörü alınan video çerçevesinin numarası.
	Start	Karakteristik vektörü çıkarılan zaman aralığının başlangıcı.

Tablo Adı	Alan Adı	Açıklama
	End	Karakteristik vektörü çıkarılan zaman aralığının sonu.
	FeatureVector	Karakteristik vektör. Bu alan için önceden tanımlı bir fonksiyon olmadığından istenildiği gibi girilebilir.
Content	OID	Nesne Belirteci (Object Identification Number – OID).
	Name	İçeriğin adı
	Description	Nesnenin kısa tanımı.
	Storage	Orjinal içeriğin (ortamın) arşiv bilgisi. Bu alan <i>STORAGE</i> tablosunun OID alanını yabancı anahtar olarak içerir.
	File path.	Eğer ortam dosya sisteminde saklanıyorsa görelî dosya yolu.
	BLOB	Eğer ortam veritabanında saklanıyorsa, bu alana konur.
Geographical Location	OID	Nesne Belirteci (Object Identification Number – OID).

Tablo Adı	Alan Adı	Açıklama
	Name	Coğrafi yerin adı
	Domestic/Foreign	Coğrafi yerin ülkesinin bulunulan ülke mi yoksa bir başka ülke mi (yurtiçi mi yoksa yurt dışı mı) olduğunu belirler. Eğer yurtiçi ise "Country" alanına gerek yoktur.
	Country	Coğrafi yerin ülkesi
	State	Coğrafi yerin ili
	District	Coğrafi yerin ilçesi
	Region	Coğrafi yerin bölgesi. Bölge bilgisi coğrafi yer, eğer, ülke, il, ilçe üçlemesiyle tanımlanamıyorsa kullanılır.
Role	OID	Nesne Belirteci (Object Identification Number – OID).
	Name	Rol adı (bkz. Ekler – Tanım Listeleri – Rol Tipleri)
Person	OID	Nesne Belirteci (Object Identification Number – OID).
	Name	Gerçek/tüzel kişi adı
News Type	OID	Nesne Belirteci (Object Identification Number – OID).

Tablo Adı	Alan Adı	Açıklama
	Name	Haber türü. (bkz. Ekler – Tanım Listeleri – Haber Türleri)
News Subject	OID	Nesne Belirteci (Object Identification Number – OID).
	Name	Haber konusu. (bkz. Ekler – Tanım Listeleri – Haber Konuları)
Importance	OID	Nesne Belirteci (Object Identification Number – OID).
	Name	Önem seviyesi. (bkz. Ekler – Tanım Listeleri – Önem Seviyeleri)
MIME Type	OID	Nesne Belirteci (Object Identification Number – OID).
	Name	MIME tipi tanımı.
	Type	<type>/<sub-type> biçiminde MIME tipi.
Media Type	OID	Nesne Belirteci (Object Identification Number – OID).
	Name	Saklama türünün adı. (bkz. Ekler – Tanım Listeleri – Saklama Türleri)

Tablo Adı	Alan Adı	Açıklama
Related Persons ¹	RelatedDocument	İlgili belge. Bu alan <i>DOCUMENT</i> tablosunun OID' alanını yabancı anahtar olarak içerir.
	RelatedPerson	İlgili kişi. Bu alan <i>PERSON</i> tablosunun OID' alanını yabancı anahtar olarak içerir.
	Role	Haberdeki rolü. Bu alan <i>ROLE</i> tablosunun OID' alanını yabancı anahtar olarak içerir.
Related News ²	RelatedDocument1	İlgili belge. Bu alan <i>DOCUMENT</i> tablosunun OID' alanını yabancı anahtar olarak içerir.
	RelatedDocument2	İlgilendirilmiş belge. Bu alan <i>DOCUMENT</i> tablosunun OID' alanını yabancı anahtar olarak içerir.
Style	OID	Nesne Belirteci (Object Identification Number – OID).
	Name	Stile verilen tanımlayıcı isim.

¹ *Subject* nesnesi ve ona ait çok-a-çok ilişki tablosu bir tek *RELATED PERSONS* tablosu içinde birleştirilmişlerdir.

² Şunu hatırlatmak gerekir ki belgeler arası ilişkiler çift yönlü değildir. Yani, eğer "A belgesi B belgesi ile ilgilidir" denildiğinde, bu B belgesinin de A belgesi ile ilgili olduğunu göstermez. Çift yönlü ilişkiler, iki ayrı ilişki tanımlanması ile sağlanabilir.

Tablo Adı	Alan Adı	Açıklama
	FontName	Sayfa Düzenleyicisi'nde Stili gösterirken kullanılacak fontun adı.
	FontSize	Sayfa Düzenleyicisi'nde Stili gösterirken kullanılacak fontun boyu.
	FontBold	Sayfa Düzenleyicisi'nde Stili gösterirken kullanılacak fontun koyu olup olmayacağı.
	FontItalic	Sayfa Düzenleyicisi'nde Stili gösterirken kullanılacak fontun italik olup olmayacağı.
	FontUnderline	Sayfa Düzenleyicisi'nde Stili gösterirken kullanılacak fontun altı çizili olup olmayacağı.
	FontStrikethrough	Sayfa Düzenleyicisi'nde Stili gösterirken kullanılacak fontun üstü çizili olup olmayacağı.
	FontColour	Sayfa Düzenleyicisi'nde Stili gösterirken kullanılacak fontun rengi.
	StartTag	HTML'de Stili gösterirken kullanılacak başlangıç etiketi.
	EndTag	HTML'de Stili gösterirken kullanılacak bitiş etiketi.

Tablo Adı	Alan Adı	Açıklama
Web Link	OID	Nesne Belirteci (Object Identification Number – OID).
	Name	Link ismi
	URL	Nesnenin gösterdiği Internet kaynağına ait URL (Unified Resource Locator).



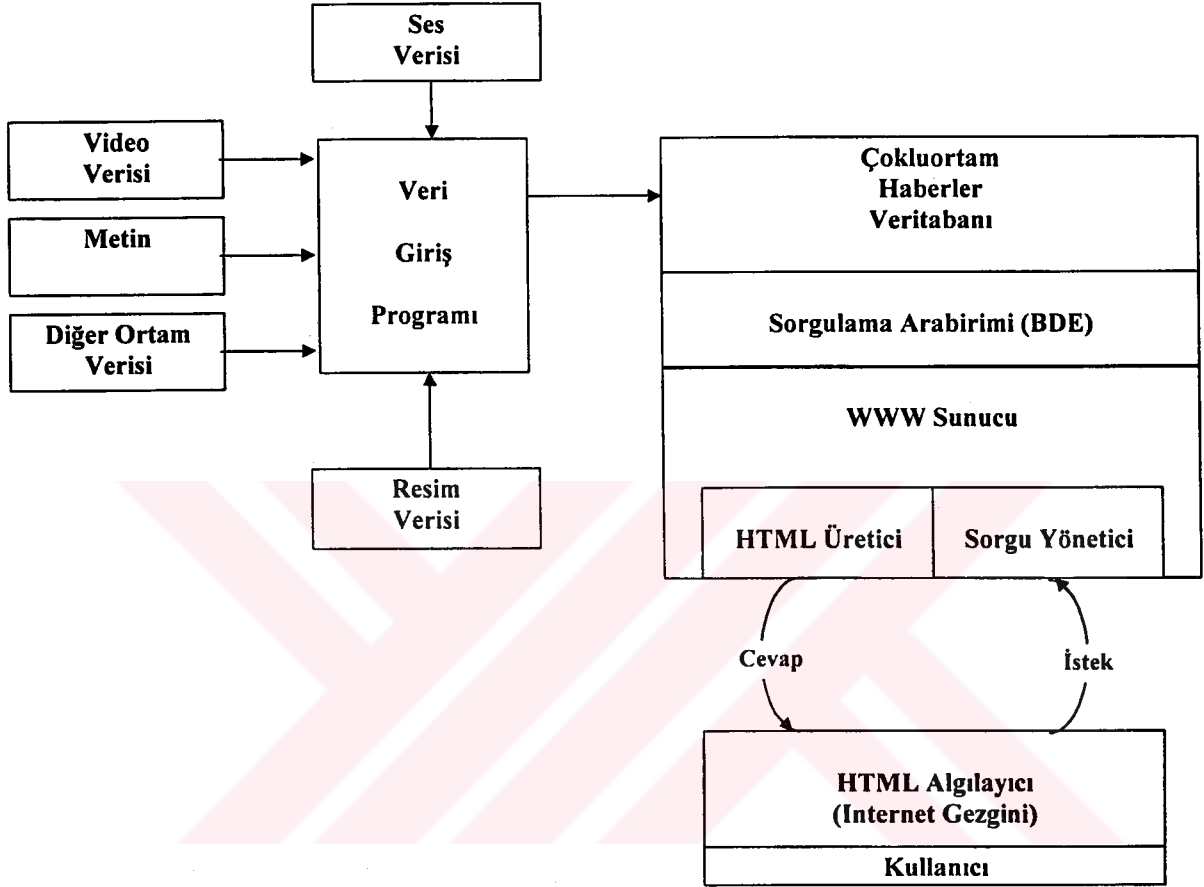
5. UYGULAMA

Sistemin nesne tabanlı tasarımı bitince, sıra uygulamanın nasıl yapılacağına gelmiştir. Önerilen sistemin uygulanmasında iki farklı kısım göze çarpar,

- Veritabanının uygulanması.
- Uygulama program(lar)ının geliştirilmesi,

Bu ayırım, veritabanının sistemin genel performansı etkileyecek en büyük faktör olacağı gerçeğine dayanmaktadır. Ayrıca, şuna da dikkat çekmek gerekir ki, çokluortam sistemin performansı üzerinde en etkili faktörlerden biri olan ağ bant genişliği hiç hesaba katılmamıştır. Bunun sebebi sistemin HTTP gibi yüksek seviyeli bir protokolün üstünde geliştirilecek olması ve Internet'te herhangi bir kabüle imkan tanımayacak kadar çeşitli ağ performanslarının görülebilir olmasıdır.





Şekil 5-1 : Sistem mimarisi

5.1 Uygulama Programı

Uygulama programı nesne tabanlı yazılım geliştirme teknikleri kullanılarak, Borland® Delphi® programlama dili ile geliştirilmiştir. Nesne tabanlı programlama altyapısı sayesinde, nesne tabanlı tasarım uygulama programında doğrudan uygulanabilmiştir.

Sözü edilen uygulama programı, aslında “Veri Giriş Programı” ve “Veri Yayıncısı” adı altında iki temel modülden oluşur.

5.2 Veri Giriş Programı

Veri Giriş programı çokluortam belgelerinin kompozisyonu ve veritabanında saklanması için kullanılır. Bu uygulama, istemci/sunucu mimaride ve Windows 95, Windows NT 3.51 veya üst sürümlerinde de çalışmak üzere geliştirilmiştir.

Çokluortam belgelerinin kompozisyonu tek tek sayfaların eklenmesi; text, video, resim, ses gibi çokluortam nesnelerinin veya WWW bağlantısı gibi çokluortam olmayan nesnelerin eklenmesi ve düzenlenmesi; metnin bölümlerinin “Stil” adı verilen mantıksal HTML tipleri ile etiketlenmesi ile tasarımını gerektirir. Bu tiplere “mantıksal tipler” denilmesinin sebebi, metnin, font, renk vs. ile belirlenen şekilsel görünümünden çok anlamını belirtmelerindedir. Sonuç olarak aynı mantıksal tipler, kullanılan HTML algılayıcının (WWW gezgini) tipine göre, de değişik bilgisayarlarda değişik görüntülenebilirler.

Son olarak, kullanıcı kompozisyona ilişkin açıklayıcı ve/veya tarihsel bilgileri de girerek tasarımı bitirir.

Şimdi, adım adım çokluortam belge oluşturulması sürecine bir bakarsak,

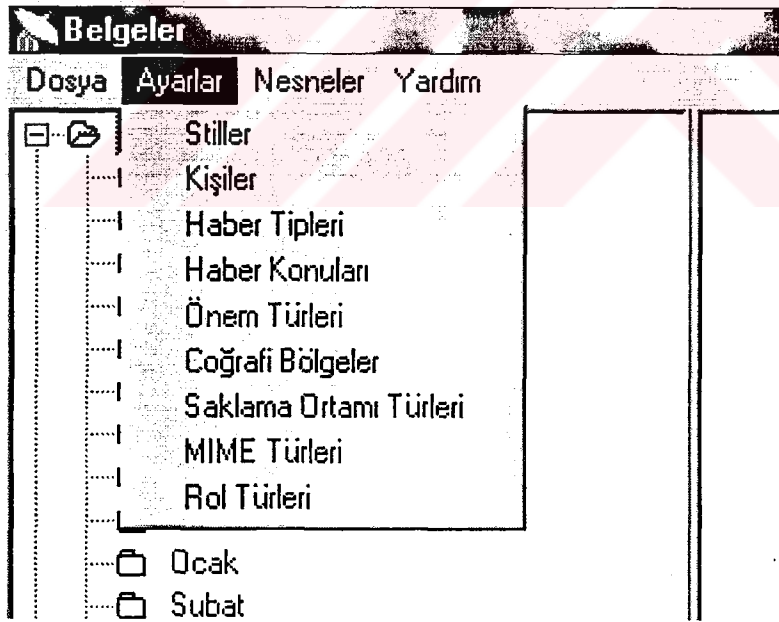
1. Kullanıcı bağlam menüsünden “Yeni Belge”yi seçerek başlar. Bu aşamada sistem otomatik olarak kompozisyona bir OID atar.
2. İlk yaratıldığında belgenin hiç sayfası yoktur. Kullanıcı, “Yeni Sayfa” seçeneği ile her seferinde yeni bir sayfa yaratabilir.

3. Yeni bir sayfa yaratıldığında, Metin ve Çokluortam nesnelere ilgili seçeneklerle sayfaya eklenirler. Her sayfa sıfır veya daha fazla sayıda çokluortam nesnesi ile sıfır veya bir Metin nesnesi içerebilir. Çokluortam nesnelere ancak Metin nesnesi varsa eklenebilirler.

Çokluortam nesnelere eklenmesinde, kullanıcı sistemde varolan nesnelere birini seçer veya yeni bir nesne yaratır.

4. Son olarak kullanıcı, açıklayıcı ve tarihsel bilgileri girip kompozisyonu bitirir. Bu bilgiler arşiv bilgisi, anahtar kelimeler, gruplama bilgileri vs.dir.

Veri Giriş programı, belgeleri bir dosya sistemindeki klasörler ve dosyalara benzer bir mantıkla yönetir. Windows 95 gezgini benzeri bir pencere, klasörlerin yaratılması, silinmesi ve adlandırılmalarına, belgelerin yaratılması, silinmesi ve düzenlenmelerine imkan tanır. Bundan başka, kullanıcılar "Ayarlar" menüsünden tanım nesnelere düzenleyebilir veya "Nesneler" menüsünden birincil nesnelere bağlamdan bağımsız olarak erişebilirler.



Şekil 5-2 : Ayarlar menüsü

Aşağıda "Ayarlar" menüsündeki işlemlerin açıklamaları verilmiştir.

5.2.1 Stiller menüsü

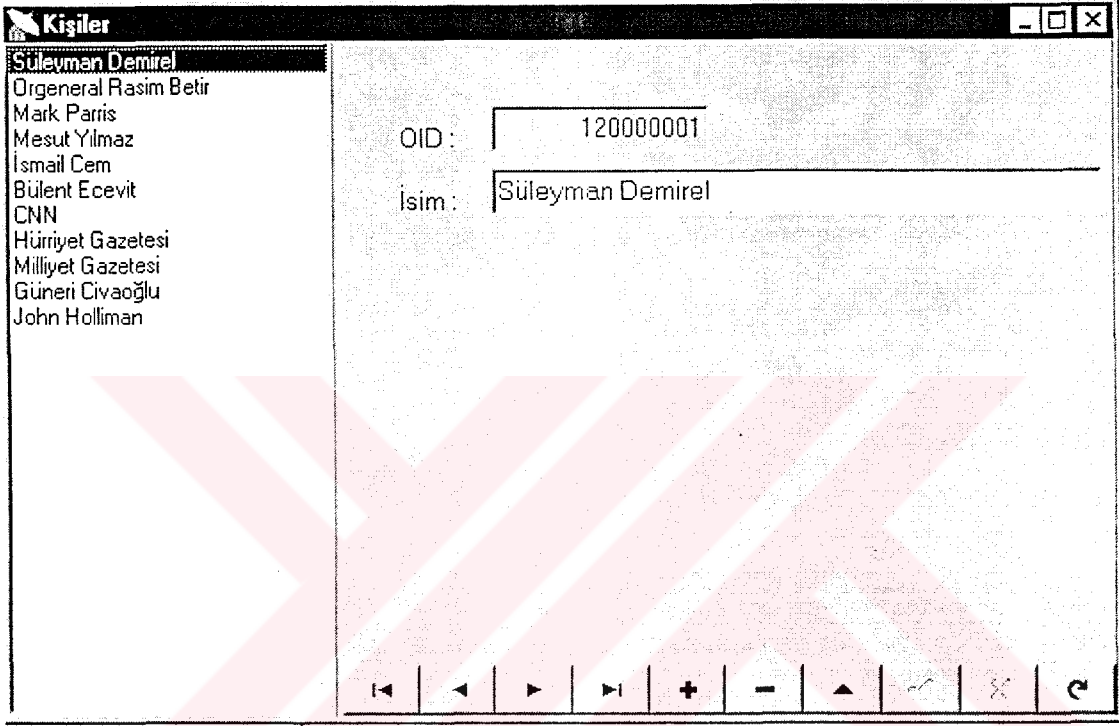
Stiller, sayfalarda kullanılan HTML stillerinin tanımlanabildiği bir pencere açar. Burada stiller, font, büyüklük, renk, koyuluk, italik, alt çizgi, üstü çizili gibi görünüş bilgilerinin yanısıra, HTML etiketleri ile birlikte tanımlanırlar. Tanımlanan HTML etiketleri, stilin kullanıldığı sayfa üretildiğinde, metin parçasının etrafını çevirirler. Görünüşe ilişkin diğer bilgiler ise stilin Sayfa Düzenleyicisi'nde görüntülenmesi için kullanılır.

Stil İsmi	Font	Boy	K / İ / Ü	Renk	HTML Etiketi Başlangıç	HTML Etiketi Son	Örnek
Başlık 1	Arial	14	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>		<H1>	</H1>	Örnek
Başlık 2	Arial	12	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>		<H2>	</H2>	Örnek
Başlık 3	Arial	12	<input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>		<H3>	</H3>	<i>Örnek</i>
Başlık 4	Arial	12	<input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>		<H4>	</H4>	<i>Örnek</i>
Koyu	Arial	10	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>				Örnek
Vurgulu	Arial	10	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>		<EMP>	</EMP>	<i>Örnek</i>

Şekil 5-3 : Stil tanımları

5.2.2 Kişiler menüsü

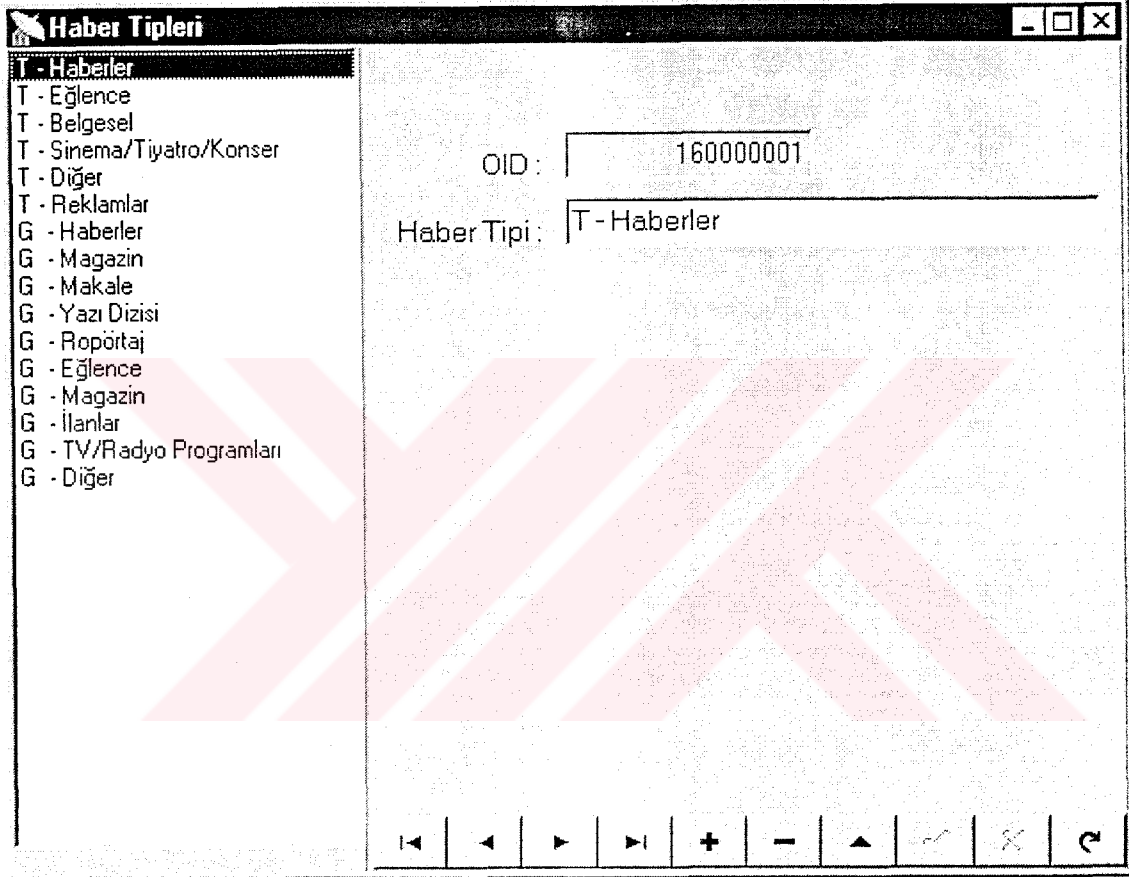
Kişiler menüsünden sistemde kullanılan kişi (*Person*) nesnelerinin tanımlamaları yapılır. Bu tanımlar daha sonra Belge (*Document*) nesnesinin Kaynak (*Source*), Yayın Yeri (*AppearedIn*), Telif Hakkı (*Copyright*) alanlarında ve İlgili Kişi nesnelerinin tanımlamalarında (*RelatedPersons*) kullanılırlar.



Şekil 5-4 : Kişi tanımları

5.2.3 Haber tipleri menüsü

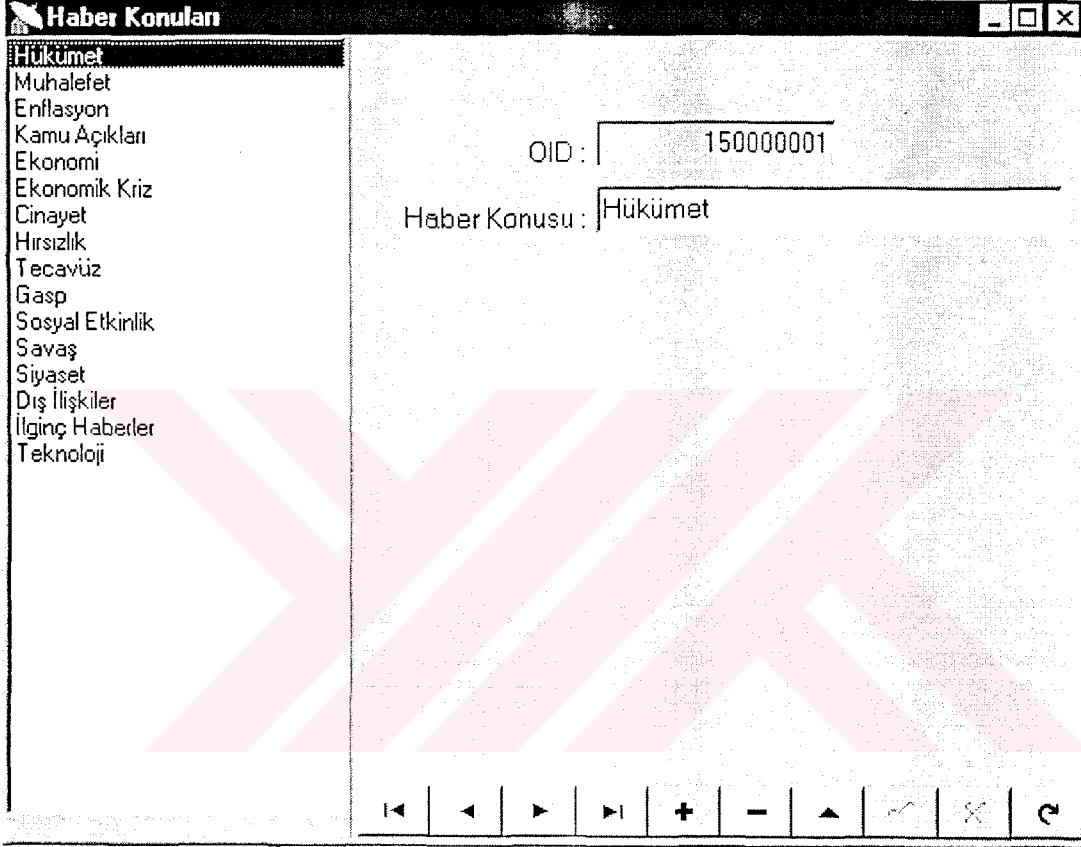
Belgelerde içeren haberin tipi bu ekranda tanımlanan değerlerden biri olmalıdır. Haber tipleri menüsü Haber Tipi (*News Style*) nesnelere için kullanılır.



Şekil 5-5 : Haber tipi tanımları

5.2.4 Haber konuları menüsü

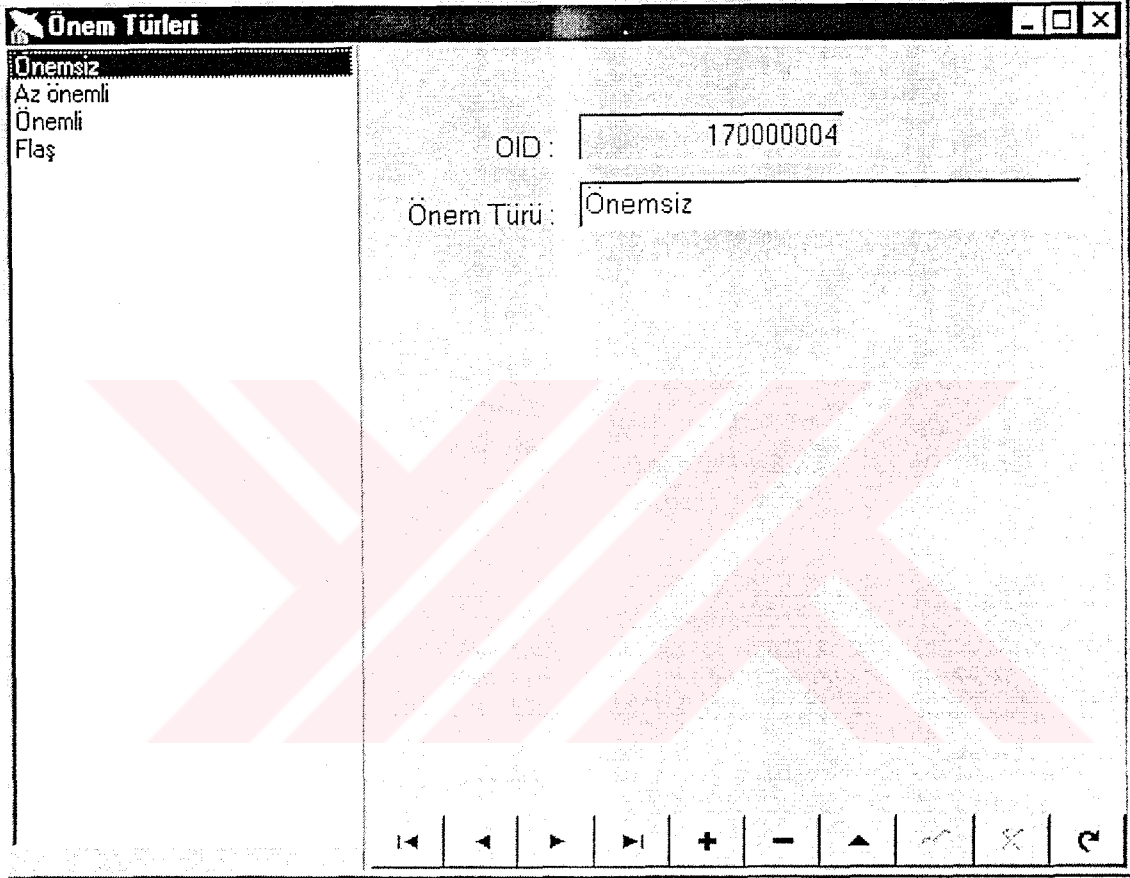
Haber konuları menüsü Belge nesnesinin Haber Konusu (*News Subject*) alanında kullanılacak değerlerin tanımlandığı menüdür.



Şekil 5-6 : Haber konusu tanımları

5.2.5 Önem seviyeleri menüsü

Her Belge (*Document*), ve dolayısı ile haber, için belirlenen önem seviyeleri bu ekrandan tanımlanır.



Şekil 5-7 : Önem seviyeleri tanımları

5.2.6 Coğrafi bölgeler menüsü

Burada tanımlanan Coğrafi Bölge (*Location*) nesneleri haberlerin ilgili oldukları bölgeleri tanımlamada kullanılırlar.

Coğrafi Bölgeler

İstanbul
Hakkari kırsalı
Güney kutbu
Fatih, İstanbul
Kuzey Irak sınırı
Silopi
Amerika
Ankara
EGLIN Hava Üssü
Clearwater, Florida

OID : 100000001

İsim : İstanbul

Mahal :

İl : İstanbul

Ülke :

Bölge :

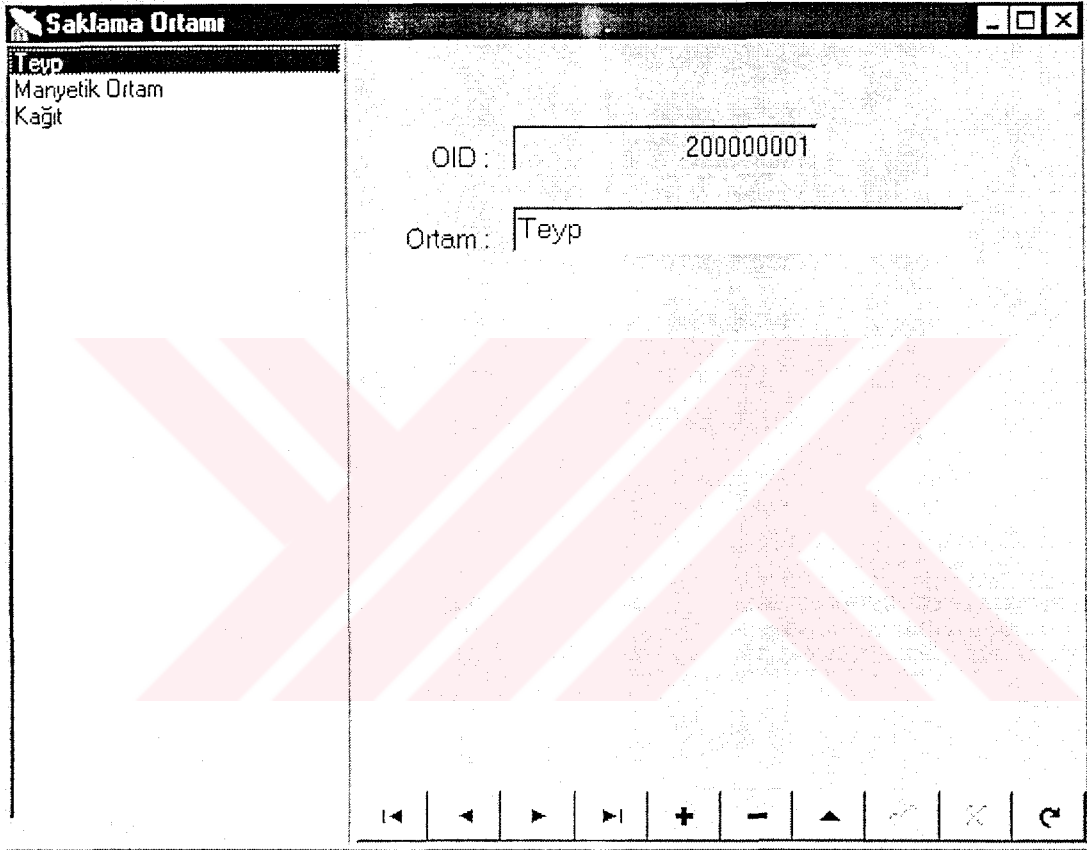
Yerel bölge tanımı ?

Navigation icons: back, forward, zoom in, zoom out, etc.

Şekil 5-8 : Coğrafi bölge tanımları

5.2.7 Saklama ortamı türleri menüsü

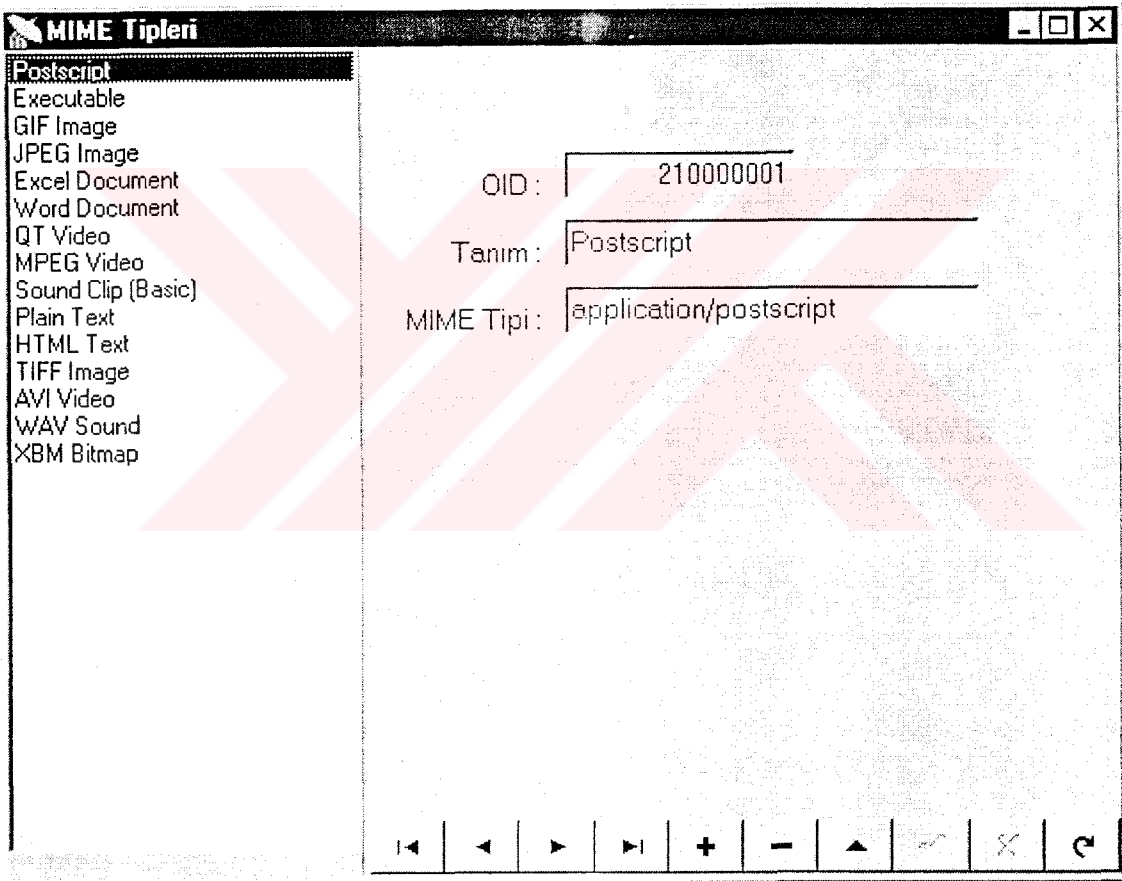
Sistemdeki her arşiv nesnesinin belirlediği ortamın saklanabileceği ortamların listesi bu ekrandan tanımlanır ve Arşiv (*Storage*) nesnelерinin tanımında kullanılır.



Şekil 5-9 : Saklama ortamı türü tanımları

5.2.8 MIME türleri menüsü

MIME türleri bir İçerik (*Content*) nesnesinin biçimini belirlemek için kullanılırlar. MIME tipi bilgisi HTTP başlığının "Content Type" alanına atanır. Kullanıcılar, İçerik (*Content*) nesnelere için MIME tipi'ni seçerken son derece dikkatli olmalıdırlar. Eğer MIME tipi doğru girilmez ve Internet gezgini tarafından anlaşılamazsa, gezgin yanıtıcı olarak "Cannot open site XXXX" mesajını verir.

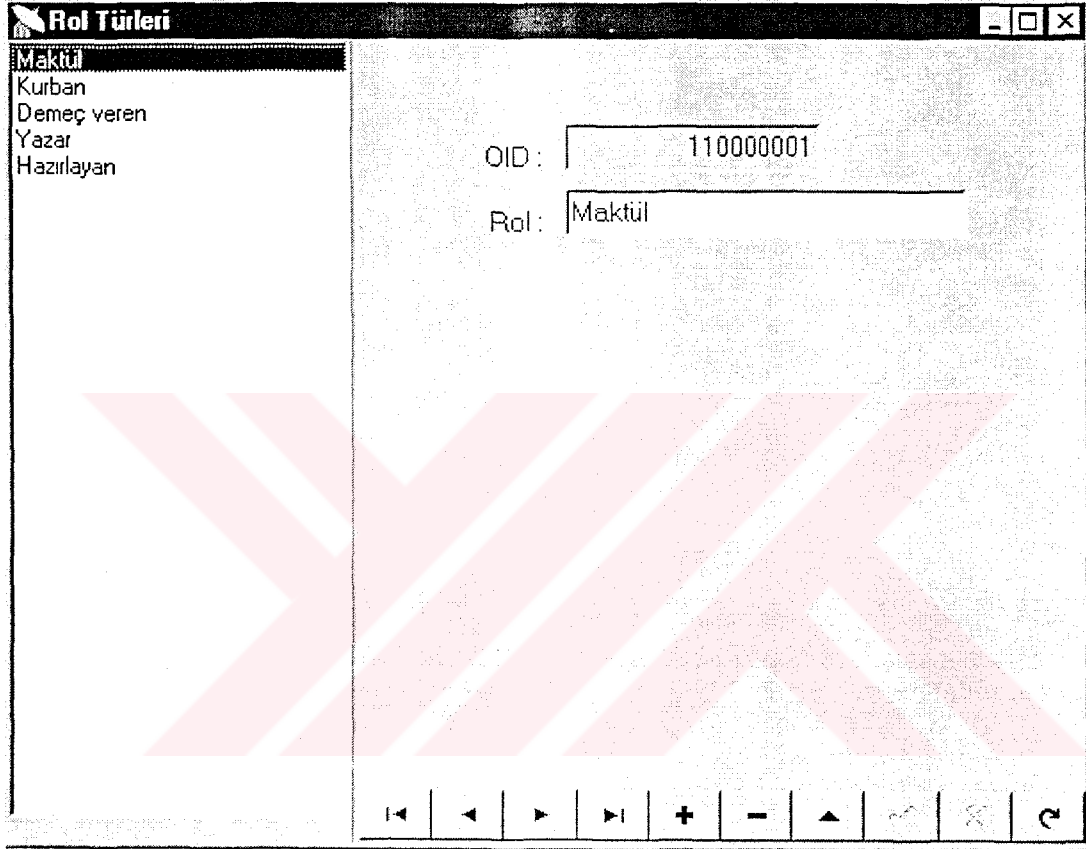


Şekil 5-10 : MIME tipi tanımları

5.2.9 Rol türleri menüsü

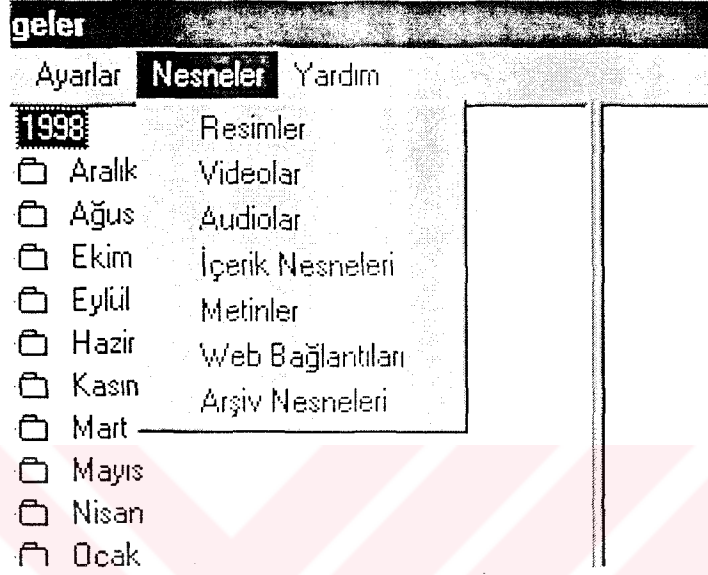
Buradan girilen Rol (*Role*) nesneleri belgelerle ilgili kişileri, Kişi'nin (*Person*)

Belge'deki (*Document*) rolü ile tanımlamak için kullanılırlar.



Şekil 5-11 : Rol tanımları

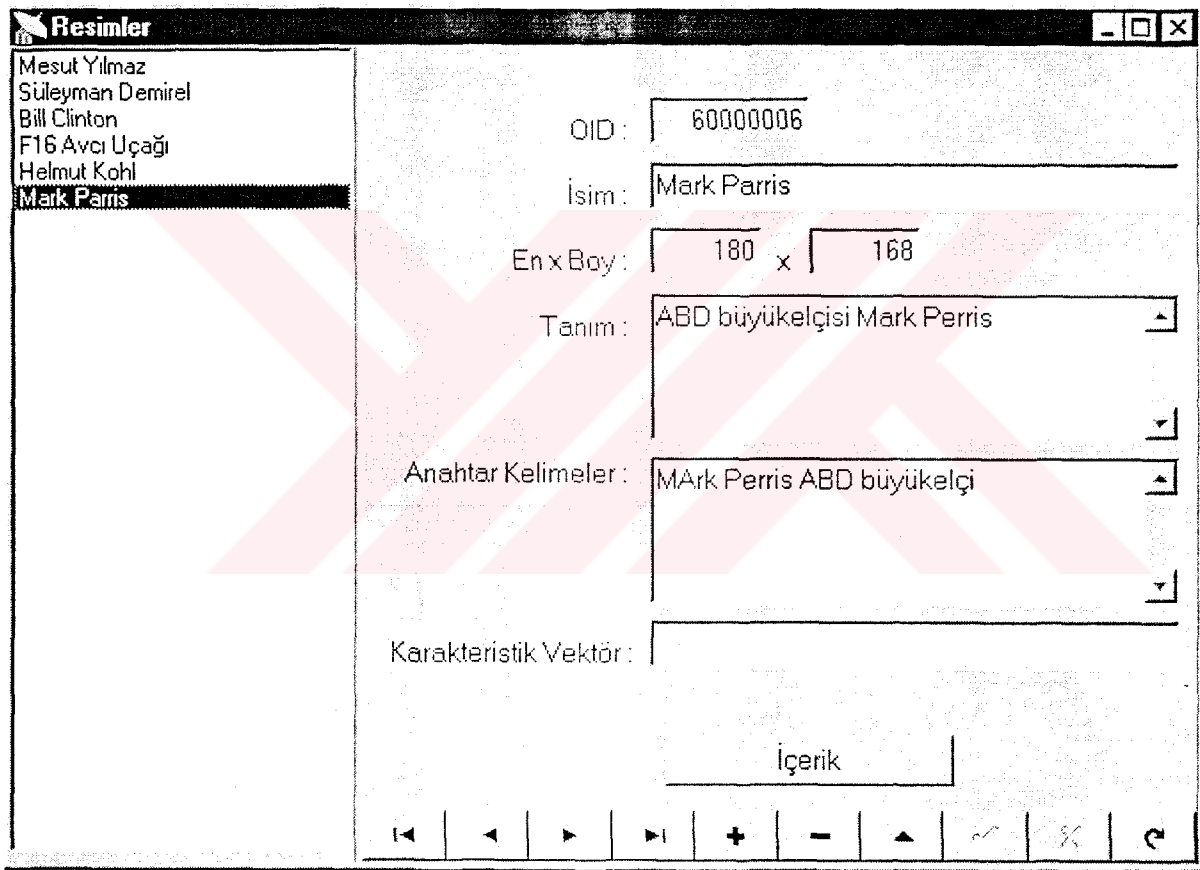
“Nesneler” menüsü sistemdeki ana nesnelerin birer birer gözlenmeleri ve güncellenmeleri için kullanılırlar.



Şekil 5-12 : Nesneler menüsü

5.2.10 Resimler menüsü

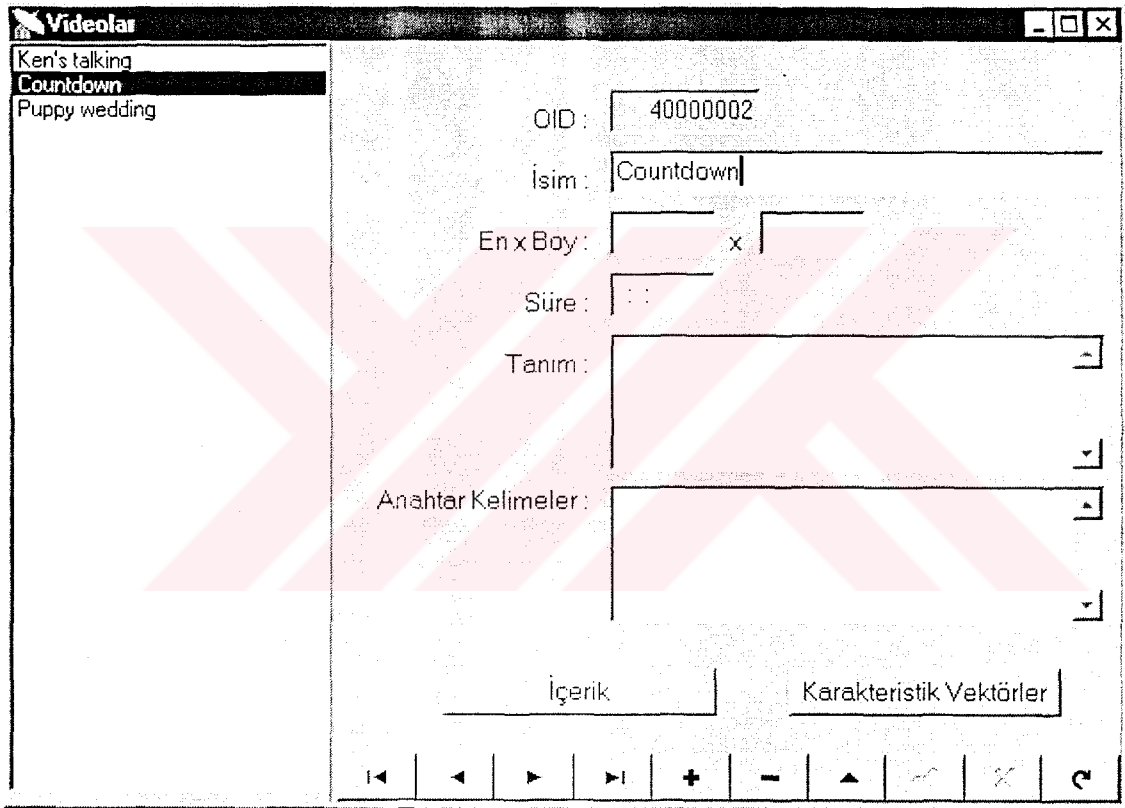
Resimler menüsü sistemdeki tüm Resim (*Image*) nesnelerinin belge Sayfalarındaki (*Page*) kullanımlarından ve diğer nesnelerle ilişkilerinden bağımsız olarak düzenlenmesine izin verir. Aynı durum Video (*Video*) ve Ses (*Audio*) nesneleri için de geçerlidir. Bu pencereden, ayrıca, görünen Resim nesnesi ile ilgili İçerik (*Content*) nesnesine de erişilebilir.



Şekil 5-13 : Resim nesnelere

5.2.11 Videolar menüsü

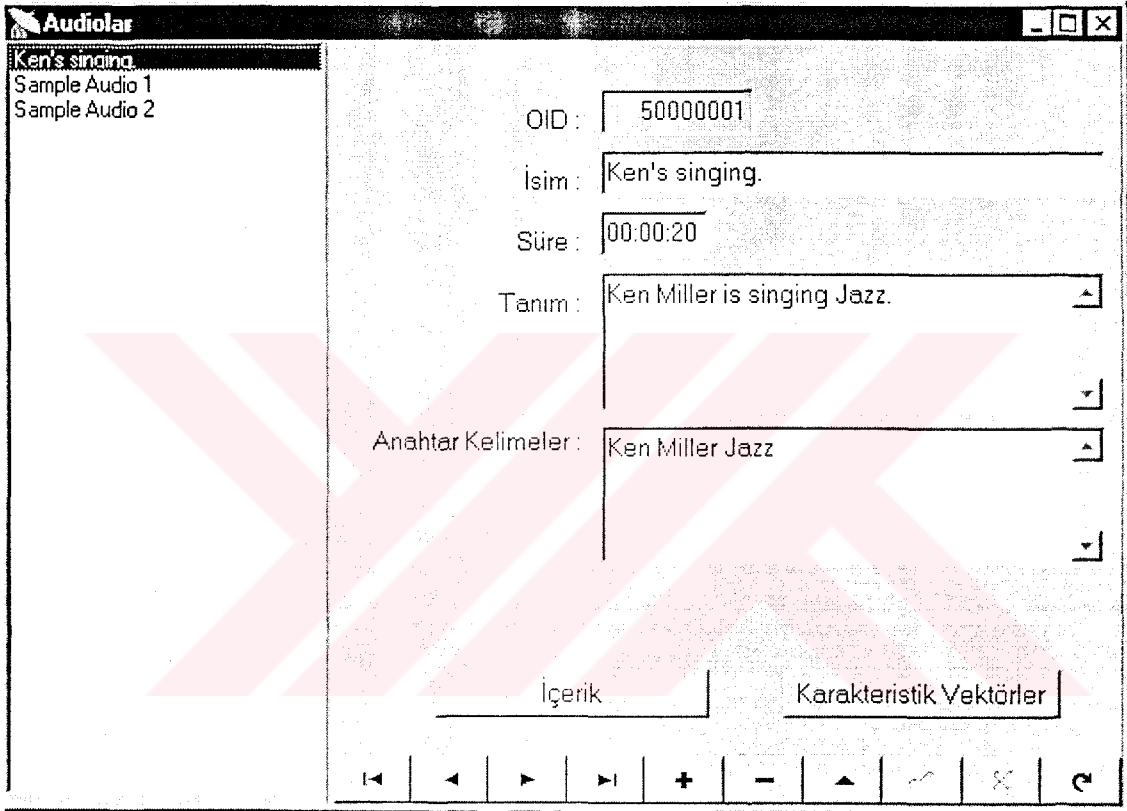
Buradan sistemdeki tüm Video (*Video*) nesneleri, Resim (*Image*) nesnelindeki kurallarla yönetilirler. Yine, Resim nesnelindeki gibi bu pencereden video ortamını içeren İçerik (*Content*) nesnesine erişim imkanı vardır. Bunu ötesinde, Video nesnesine ait Karakteristik Vektör'lere (*Feature Vector*) de bu formdan erişilebilir.



Şekil 5-14 : Video nesneleri

5.2.12 Audiolar menüsü

Bu menüden açılan form ile Ses (*Audio*) nesneleri Video (*Video*) nesnelere benzer şekilde düzenlenirler.



Şekil 5-15 : Audio nesneleri

5.2.13 İçerik nesneleri menüsü

Sistemdeki tüm İçerik (*Content*) nesneleri, bu menüden açılan for üzerinde diğer nesnelere ile olan baba-oğul ilişkilerinden bağımsız olarak, yani düz bir görünüşte, düzenlenebilirler.

Bir İçerik nesnesinin ortamı, dosya sisteminde bir dosya olarak veya veritabanındaki BLOB yapısında bir alanda saklanabilir. Eğer dosya sisteminde saklanıyorsa, sadece sanal dizin bazında URL belirteci girilir. Bu durumda, İçerik nesnesi WWW'de yayınlanmak istendiğinde, "İçeriği Göster" (Play Content) işlemi HTML'de "anchor" etiketi ile belirlenen bir dosya hiper-bağına denktir. Bu durumda Internet gezgini dosyayı uzantısı ile belirlediğinden, MIME tipi verilmesine gerek yoktur. Bu özellik kullanıcılara İçerik nesnesi yardımı ile istemciye dosya yükleme özelliği sağlar. Bundan başka, çok kullanılmayan bir MIME tipindeki veya hiçbir MIME tipi olmayan ortamlar da istemciye yüklenip sonradan bir başka biçimde kullanılabilir. Burada verilen URL, yapılandırma ayarları dosyasında (mmnews.ini) bulunan "ContentDir" dizinine göre, görece bir yol içermelidir.

Eğer, İçerik nesnesi veritabanı BLOB alanında saklanırsa, "İçeriği Göster" (Play Content) eylemi, HTML başlığındaki "Content Type" alanı verilen MIME tipi, ve mesaj gövdesi BLOB alanındaki biçimsiz veri olan HTTP cevabı hazırlanıp istemciye gönderilir.

Bundan başka, İçerik nesnesi eğer sayısal ortam dışında arşivlenen bir orijinal nesneyi temsil ediyorsa veya İçerik nesnesinde saklanan ortam sayısal ortam dışında da arşivleniyorsa, bu arşiv bilgisini saklamak için bir Arşiv (*Storage*) nesnesi yaratılıp arşiv bilgilerini saklar.

İçerik [Minimize] [Maximize] [Close]

OID : 90000037

İsim : Puppy wedding

Tanım : Wedding ceremony for two dogs

Saklama Biçimi

Dosya Sisteminde Sakla

Yol : _____

Veritabanında Sakla

MIME Tipi : video/quicktime

Veritabanında içerik var Dosyadan İçerik Yükle

İçerik Boyu : 809202 Byte

Arşivleme

Oda : VDO2

Bölüm : IR3

Raf : _____

Satır : 23

Sütun : _____

Sıra : _____

Ortam : Teyp

Sample

Logo

Mark Perris

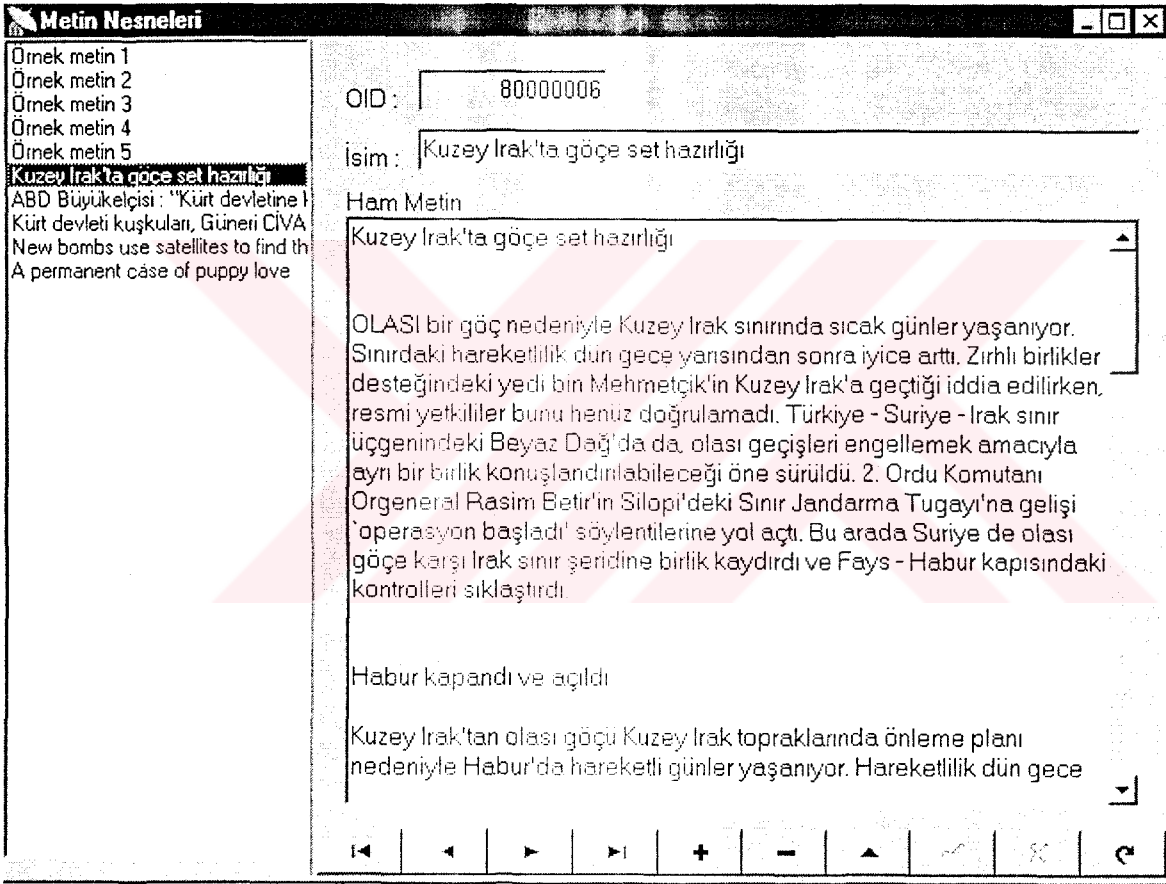
Puppy wedding

Navigation: [Home] [Left] [Right] [Next] [Add] [Remove] [Up] [Check] [Close] [Refresh]

Şekil 5-16 : İçerik nesneleri

5.2.14 Metinler menüsü

Sistemdeki her Metin (*Text*) nesnesi yalnız bir Sayfa (*Page*) nesnesi ile ilişkilidir. Sayfa nesnelere daha sonra Belge (*Document*) nesnelere bağlanırlar. Bu pencereden, sadece biçimsiz metinler görülebilir. Biçim bilgisini içeren Metin Açıklamaları (*Text Attributes*) sadece Sayfa Düzenleyicisi'nden görülebilir.



Şekil 5-17 : Metin nesnelere

5.2.15 Arşiv nesneleri menüsü

Siistemdeki tüm Arşiv (*Storage*) nesneleri bu menüden görülebilirler. Arşiv nesneleri Belge'leri (*Document*) oluşturan orjinal haberlerin veya çokluortam nesnelерinin dış ortamdaki arşivlerde saklanmasıна ait bilgileri tutarlar. Arşiv bilgisi, altı tane hiyerarşik adres alanını içerir. En genişinden en darına, bu alanlar : Oda, Bölüm, Raf, Satır, Sütun ve Sıra'dır. Arşiv bilgisi, ayrıca, arşivleme ortamını belirlemek için Saklama Ortamı Türleri'ne (*Media Type*) bir referans olan "Saklama Ortamı" alanını da içerir.

Arşiv Nesneleri	OID	Oda	Bölüm	Raf	Satır	Sütun	Sıra	Saklama Ortamı
220000001								
220000002								
220000003								
220000004								
220000005								
220000006								
220000007								
220000008								
220000009								
220000010								
220000011								
220000012	220000012	GZT	HAB	YURT	CC	3B	23	Kağıt
220000013								
220000014								

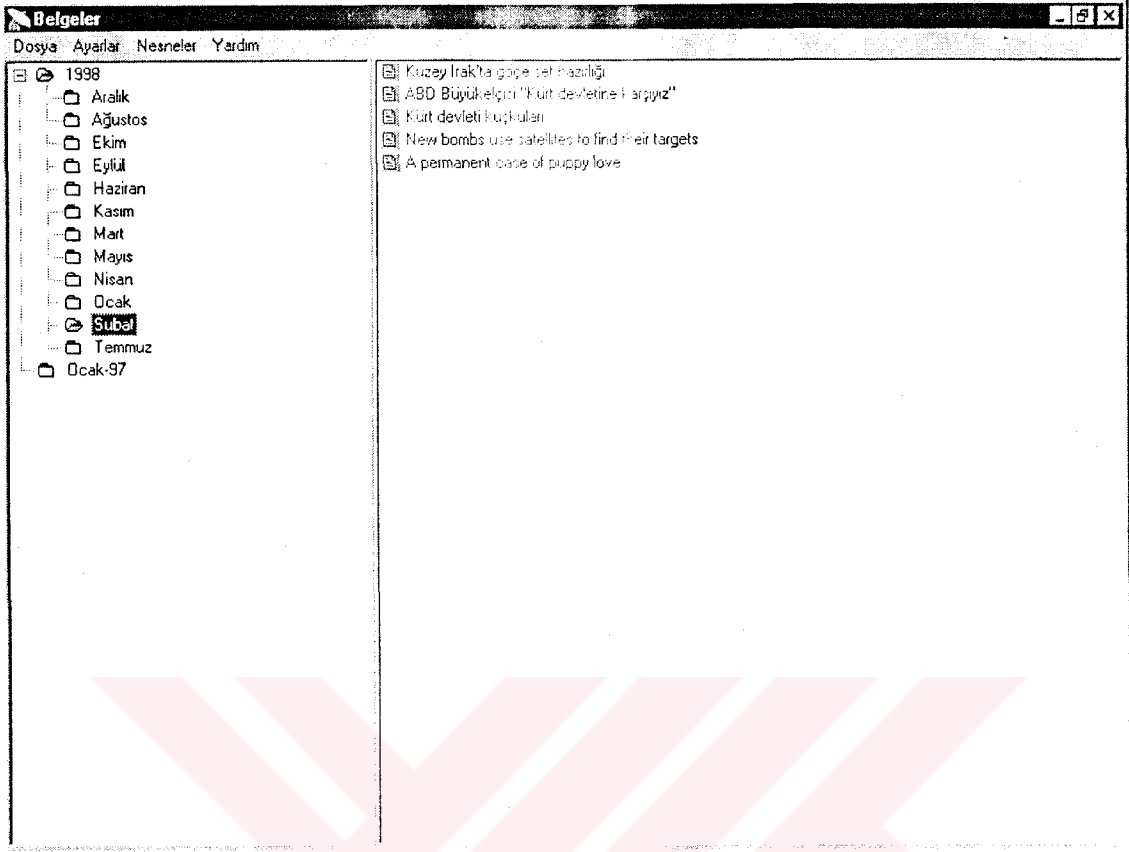
Şekil 5-18 : Arşiv nesneleri

5.2.16 Belge listesi

Belge Listesi Form'u Veri Giriş programının ana formudur. Sitemde tanımlı klasörler sol panoda, seçili klasörün içinde bulunan belgeler de sağ panoda görünürler. Bu iki pano, form ile birlikte boyut değiştirirler. Panoların genişlikleri panolar arasındaki sınır çizgisi tutulup çekilerek değiştirilebilir. Belgeler ve klasörler sağ fare tuşu ile görünen bağlam menüleri yardımı ile düzenlenirler. Klasör panosunun bağlam menüsü yeni klasör yaratma (eğer fare hiçbir dizin üstünde tıklanmadı ise "Yeni Kök Klasörü Yaratma" seçeneği gelir), klasörü silme ve adını değiştirme işlemlerini içerir.

Belge panosunun bağlam menüsü ise, Belgelerin eklenip silinmesi için gerekli komutları içerir. Her iki bağlam menüsündeki "Tazele" seçeneği Klasör veya Belge tanımlarının veritabanından yeniden okunup son durumun pencereye aktarılmasını sağlar. Böylece, son "Yenile" işleminde bu yana diğer kullanıcılar tarafından belge veya klasörler üzerinde yapılan adlandırma, silme ve ekleme işlemlerinin sonuçları da görülür.

Bir belge adının üzerine fare ile çift tıklayarak Belge Düzenleme Penceresi açılır ve bu pencereden belge ile ilgili her türlü değişiklik yapılır.



Şekil 5-19 : Belge listesi

5.2.17 Belge düzenleme ekranı

Belge Düzenleme Ekranı, kullanıcıların belge ile ilgili açıklama bilgilerini düzenlemesine, Belge ile ilgili diğer sistem nesnelere erişilmesine ve son olarak kompozisyonların düzenlenmesine imkan tanır.

Bu pencerede, Belgeye ilişkin bilgiler tablı bir kullanıcı arabirimi ile sunulur. Tablaların üstünde belgenin adı ve OID değerleri görünür.

“Genel” tabı, Belge ile ilgili genel bilgi içerir. “Başlangıç” ve “Son” alanları genel kullanım içindir. Televizyon haberleri için başlangıç ve bitiş zamanını, gazete haberleri için ise başlangıç ve bitiş sayfa/sütun numaralarını içerirler.

Belge No : 20000006 Belge Adı : ABD Büyükelçisi 'Kürt devletine karşıyız'

Genel | Önsöz / Anahtar | Arşivleme | Sayfalar | İlgili Kişiler | İlgili Belgeler | Diğer

Belge Başlığı : ABD Büyükelçisi "Kürt devletine karşıyız"

Belgeyi Yaratan : SYSDBA

Başlangıç : 1:6 Tarih : 10.02.1997

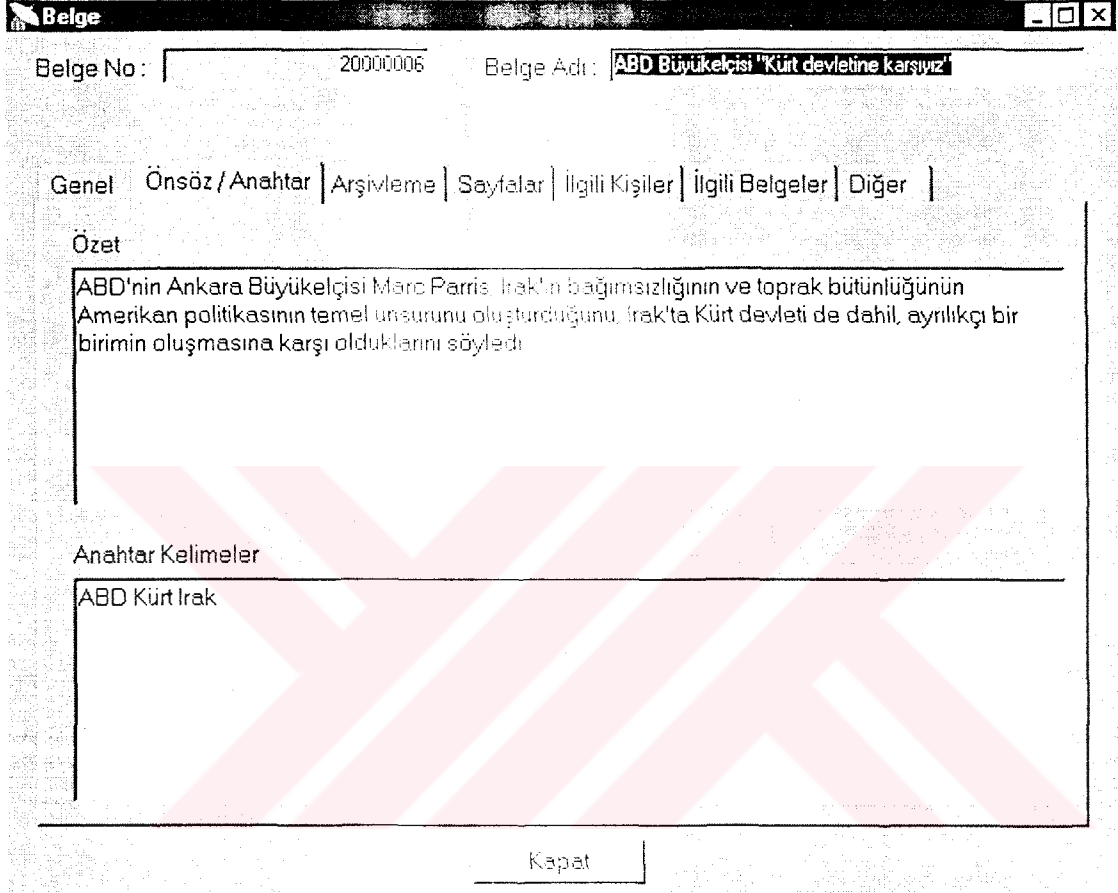
Son : 1:6 Yayın Tarihi : 10.02.1997

Açıklama
ABD büyükelçisinin demeci.

Kapat

Şekil 5-20 : Belge ekranı - genel

“Özet/Anahtar” tabında Belgenin kısa bir özetini ve anahtar kelimelerini içeren iki büyük metin alanı vardır.



Belge No : 20000006 Belge Adı: ABD Büyükelçisi "Kürt devletine karşı"

Genel Önsöz / Anahtar Arşivleme Sayfalar İlgili Kişiler İlgili Belgeler Diğer

Özet

ABD'nin Ankara Büyükelçisi Marc Parris, Irak'ın bağımsızlığının ve toprak bütünlüğünün Amerikan politikasının temel unsurunu oluşturduğunu, Irak'ta Kürt devleti de dahil, ayrılıkçı bir birimin oluşmasına karşı olduklarını söyledi.

Anahtar Kelimeler

ABD Kürt Irak

Kapat

Şekil 5-21 : Belge ekranı - özet/anahtar kelimeler

“Arşivleme” tabı, Belge’deki habere kaynak olan orjinal haberin, varsa, arşivleme bilgisini gösterir. Belgeyi oluşturan orjinal haberin, dış ortamdaki arşivlerde saklanmasına ait bilgilerine arşivleme bilgisi diyoruz. Eğer ilgili bir Arşiv nesnesi yok ise, yani Belge hiçbir arşiv nesnesine referans vermiyorsa, bu taba geçilmeye çalışıldığında. Arşiv nesnesinin bulunamadığını ve yeni Arşiv nesnesinin yaratılıp yaratılmaması gerektiğini soran bir mesaj kutusu görüntülenir.

The screenshot shows a window titled "Belge" with the following fields and values:

- Belge No: 20000006
- Belge Adı: ABD Büyükelçisi "Kürt devletine karşıyız"
- Navigation tabs: Genel | Önsöz / Anahtar | Arşivleme | Sayfalar | İlgili Kişiler | İlgili Belgeler | Diğer
- Oda: G2T
- Bölüm: YURT
- Raf: HAS
- Satır: 00
- Sütun: 24
- Sıra: 4
- Saklama Ortamı: K.01
- Buttons: Kapat

Şekil 5-22 : Belge ekranı - arşiv bilgisi

“**Sayfalar**” tabı kullanıcının Belge (*Document*) Sayfa’larını (*Page*) düzenlemesini sağlar. Sağdaki listede Sayfa numaraları ve isimleri listelenmiştir. Listedeki bir Sayfa bilgisinin üzerine gelindiğinde, sağdaki alanlara Sayfaya ait bilgiler doldurulur. Sayfa ile ilgili bilgiler OID, İsim, Sayfa No ve tabi ki Metin bilgileridir. “Metin Düzenle” tuşu ile combobox kontrolünde ismi yazılı sayfa Metni’nin (*Text*) düzenlenmesini sağlayan “Sayfa Düzenleyicisi” ekranı açılır. “Yeni Sayfa Aç” tuşu ise her seferinde Belge’ye yeni bir sayfa eklemeye yarar. Şunu da belirtmek gerekir ki, yeni sayfa yaratıldığında ilgili Metin nesnesi yaratılmaz. Kullanıcı, özel olarak Metin nesnesini de yaratmak zorundadır.

Yeni bir Sayfa yaratıldığında varsayılan bir isimle yaratılır. Kullanıcı isterse bu ismi değiştirebilir.



Belge No : 20000006 Belge Adı : ABD Büyükelçisi "Kürt devletine karşıyız"

Genel | Önsöz / Anahtar | Arşivleme | Sayfalar | İlgili Kişiler | İlgili Belgeler | Diğer

1. Sayfa 1

Sayfa OID : 30000008

Belge OID : 20000006

İsim : Sayfa 1

Sayfa No : 1

Sayfa Metni : ABD Büyükelçisi : ▾

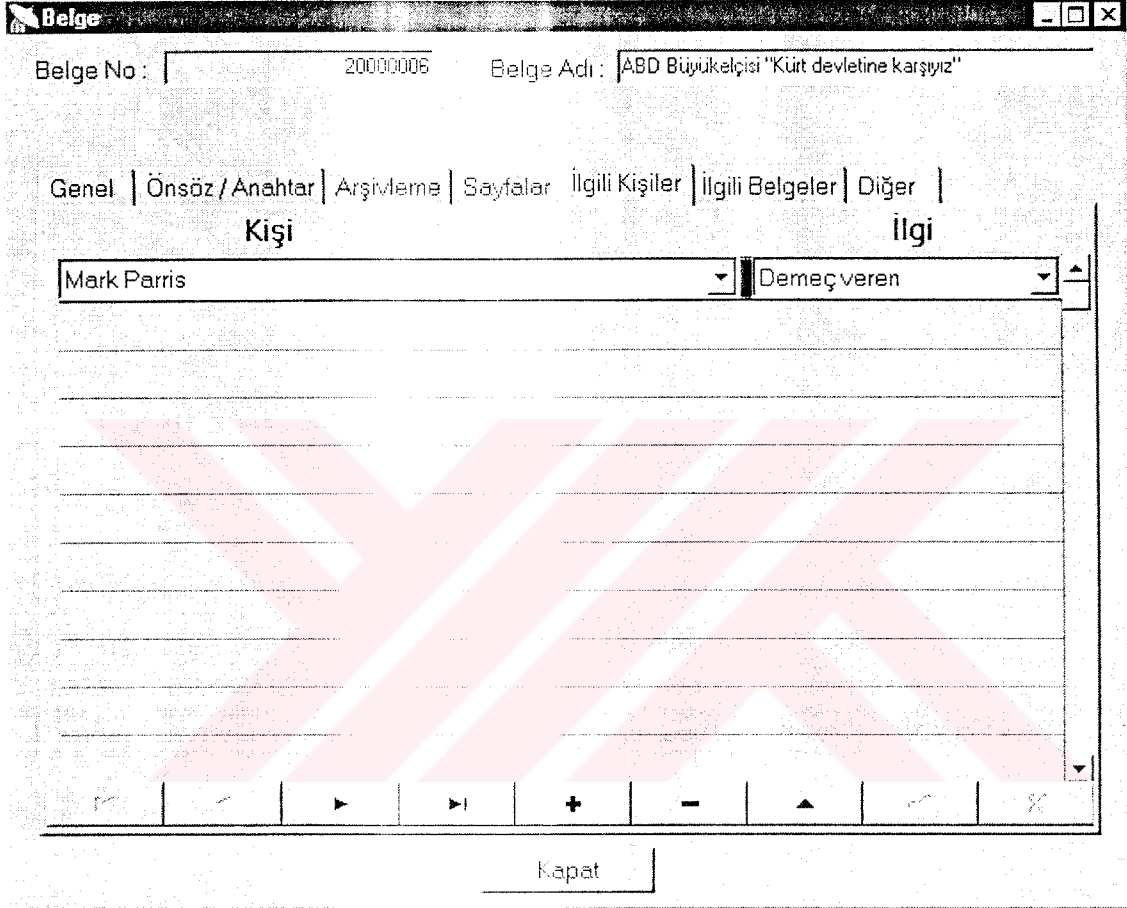
Sayfa Metni Düzenle

Yeni Sayfa Aç

Kapat

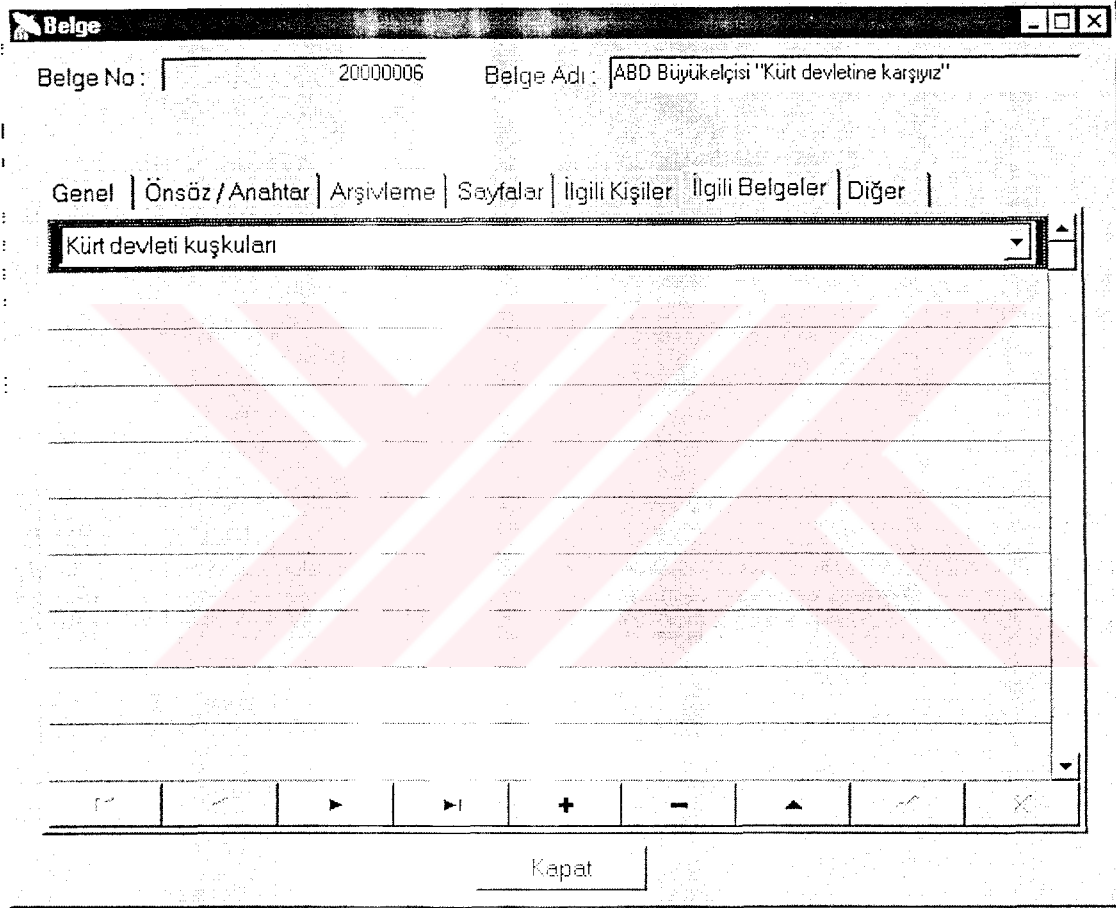
Şekil 5-23 : Belge ekranı - sayfalar

“İlgili Kişiler” tabında Belge (ve içerdiği haber) ile ilgili Kişi’lerin (*Person*) ve Rol’lerinin (*Role*) bir listesini görebiliyoruz. Kullanıcılar listenin altındaki tuşları kullanarak bu listedeki kayıtları düzenleyebilirler.



Şekil 5-24 : Belge ekranı – ilgili kişiler

“İlgili Belgeler” tabında ise o anda düzenlenen Belge ile ilgili diğer Belge’lerin bir listesi vardır. “İlgili Kişiler” tabındaki gibi listenin altında görünen tuşlarla, varolan kayıtlar düzenlenip yeni kayıtlar eklenebilir veya silinebilir. Bu listedeki Belge’lerin, bilgileri görünen Belge ile ilgilerine kullanıcı karar verir.



Şekil 5-25 : Belge ekranı - ilgili belgeler

“Diğer” tabında ise, Belgeye ilişkin diğer açıklama bilgileri görülmektedir. Tüm bu bilgiler, aslında, tanım nesnelere gelmektedirler. Ayrıntılı olarak bakarsak burada “Haber Kaynağı”, “Telif Hakkı”, “Yayımlandığı Yer” alanları birer Kişi (*Person*) nesnesi, “Haber Tipi” alanı bir Haber Tipi (*News Type*), “Haber Konusu” alanı bir Haber Konusu (*News Subject*) nesnesi, “Yer” alanı bir Coğrafi Bölge (*Location*) nesnesi, “Önem Derecesi” alanı bir Önem Derecesi (*Importance*) nesnesine referanslardır.

Belge No : 20000006 Belge Adı : ABD Büyükelçisi "Kürt devletine karşıyız"

Genel | Önsöz / Anahtar | Arşivleme | Sayfalar | İlgili Kişiler | İlgili Belgeler | Diğer

Telif Hakkı : Hürriyet Gazetesi

Yayımlandığı Yer : Hürriyet Gazetesi

Önemi : Önemsiz

İlgili Yer / Bölge : Ankara

Konu : Dış İlişkiler

Haber Kaynağı : Hürriyet Gazetesi

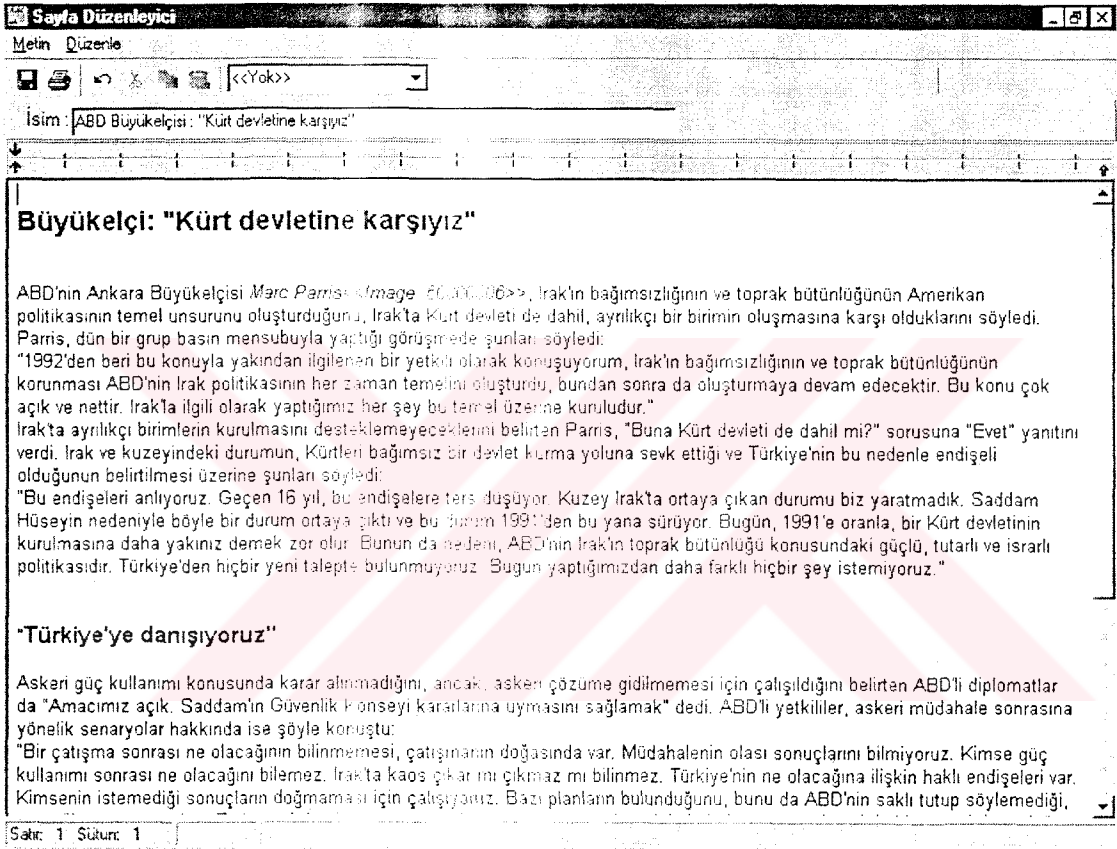
Haber Tipi : G - Haberler

Kapat

Şekil 5-26 : Belge ekranı - diğer alanlar

5.2.18 Sayfa Düzenleyicisi

Sayfa Düzenleyicisi ekranı açıldığında, önce biçimsiz metin yüklenir. Daha sonra, Metin (*Text*) nesnesinin Metin Açıklamaları'ndan (*Text Attributes*) yararlanılarak yüklenen metin biçimlendirilir.



Şekil 5-27 : Sayfa Düzenleyicisi

Sayfa Düzenleyicisi ile kullanıcılar, bir Sayfa'nın Metin (*Text*) nesnesini düzenlerler. Metin nesnesinin kendisi sadece biçimsiz metni saklar. Metnin biçimine ilişkin bilgiler Metin Açıklamaları (*Text Attributes*) nesne dizisinden (veya tablosundan) gelirler.

Bir Metin Açıklaması nesnesi açıklamanın nerede başlayıp bittiğinin yanısıra, açıklamaya özel bilgi içerir. Bir metin parçası için üç değişik açıklama tanımlanabilir :

- HTML etiket açıklaması, aslında, bir Stil (*Style*) nesnesi OID'ini saklar. Bu durumda, metin parçası Sayfa Düzenleyicisi'nde Stil nesnesinde belirlenen görünüm biçiminde gösterilirken, HTML'de yine Stil nesnesinde belirlenen etiketlerin arasında (yani bu etiketlerin belirlediği mantıksal tip ile) gösterilir.
- Yazı içi nesne açıklaması ise, bir yazı içi resim veya Java applet dosyası için gerekli URL'i içerir. Bu URL, yapılandırma ayarları dosyasında (mmnews.ini) bulunan "ImagesDir" veya "AppletsDir" dizinlerine göre, görelî bir yol içermelidir. Sayfa Düzenleyicisi'nde, yazı içi nesnelere özel olarak biçimlendirilirler. Kırmızı metin rengi ile ve italik olarak yazılırlar. Dosya isimleri "<<" ve ">>" işaretleri ile sınırlandırılmışlardır. HTML sayfalarında ise resimler "", Java applet'leri ise "<APPLET>" etiketleri ile gönderilirler. Resimler ve applet'ler dosya uzantıları ile ayırt edilirler. Örnek olarak, "cats.jpg" adındaki bir resim dosyası Sayfa Düzenleyicisi'nde aşağıdaki gibi görünür.

<<RESİM Cats.jpg>>

Bu ifadenin bir kısmı değiştirilemez, ancak tümü ile silinebilir ki bu durumda HTML sayfasındaki referansı da silinmiş olur.

- Bağ açıklaması, öte yandan, bir çokluortam nesneye (*Image, Video, Audio, Content*) veya bir Web Bağlantısı (*Web Link*) nesnesinin OID'ini içerir. Böylece, bağ sadece bir Resim, Video, Ses, İçerik veya Web Bağlantısı

nesnesini gösterebilir. Nesne tipi OID'in ilk iki alanındaki tip bilgisinden çıkartılır. Bağ açıklamaları da Sayfa Düzenleyicisi'nde özel bir biçimde görüntülenirler. Bağ metni, tümü ile mavi metin rengi ile ve italik olarak yazılır. Bağlanan metin parçasının sonuna, bağlanan nesnenin tipini ve OID'ini gösteren açıklayıcı bilgi, "<<<<" ve ">>>>" karakterleri ile sınırlandırılmış olarak, eklenir. Örneğin "oylanan oyunun afişi" metin parçasından "60000001" değerli OID'e sahip Resim (*Image*) nesnesine bir bağ kurulacak olursa, Sayfa Düzenleyicisi'nde

... oynanan oyunun afişi<<Image 60000001>>...

şeklinde gözüktür, veya "Borland International" metin parçasından Web Bağlantısı (*Web Link*) nesnesine bir bağ kurarsak biçimlendirme

... Borland International<<HyLink 25000001>>...

şeklinde olur. Değişik nesnelere için Sayfa Düzenleyicisi'nde görünen bağ etiketleri aşağıdaki tabloda verilmiştir.

Tablo 5-1 : Sayfa Düzenleyicisi tip etiketleri

Tip Etiketi	Nesne
Image	Image
Video	Video
Audio	Audio
MIME	Content
HyLink	Web Link

Yine dikkat edilmelidir ki, bađ bir kere kuruldu mu, ne metin parçasının kendisi ne de bađı belirleyen açıklama deđiştirilemezler. Ancak ikisi birden silinebilirler.

HTML sayfasında, eđer bađ bir çokluortam nesneyi gösteriyorsa, Veri Yayıncısı'dan nesneyi isteyen bir sorgulama isteđi olarak gerçekenir. Yukarıdaki ilk örnek HTML sayfasında ,

```
<A HREF=/mmnews.dll/Images?ImageOID=60000001> oynanan oyunun  
afiři<A>
```

olarak üretilecektir.

Eđer bađ, bir WWW adresini gösteren Web Bađlatısı (*Web Link*) nesnesine ise, bađ HTML "anchor" etiketi ile üretilir. (). Yukarıdaki ikinci örnek ise HTML sayfasında

```
<A HREF="www.borland.com">Borland International<A>
```

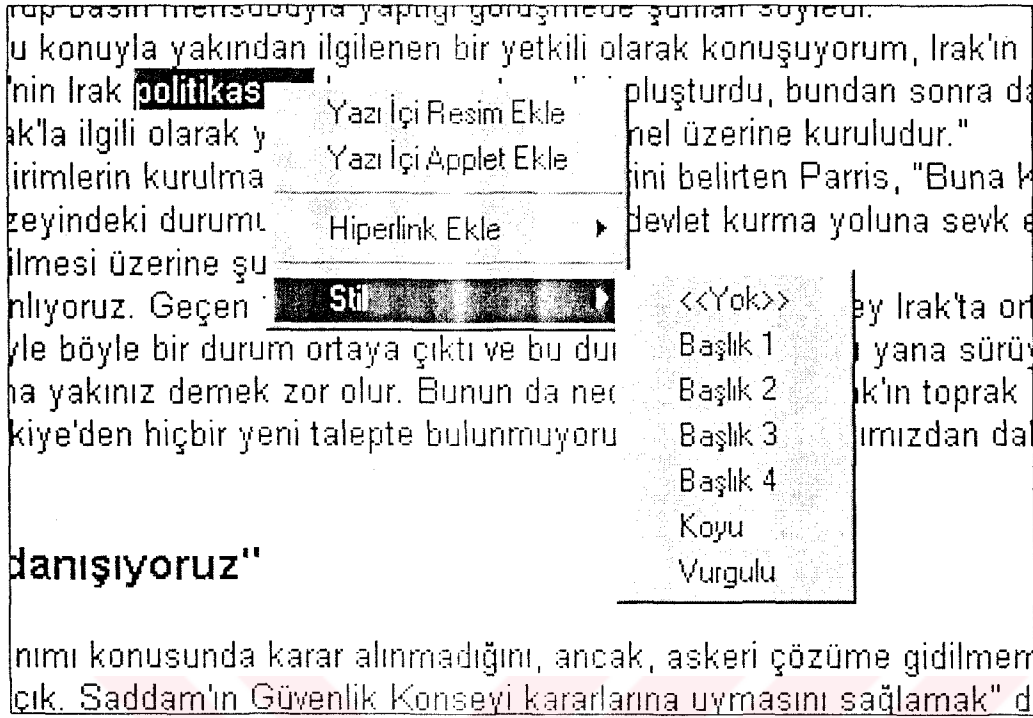
olarak üretilecektir.

Sayfa Düzenleyicisi, bir zengin metin editörü olarak tasarlanmıştır. Geri al, kes, yapıştır, kopyala, tümünü seç gibi standart metin düzenleme olanakları ile donatılmıştır. Ayrıca, yazılı metnin ASCII veya zengin metin biçimlerinde bir işletim sistemi dosyası olarak saklanmasına ve sayfanın yazdırılmasına da izin verir. Editör özellikleri menüden veya hemen altındaki araç çubuğundan kullanılabilirler.

Bir Stil açıklamasının atanması, ilgili metin parçasının seçilip istenen stilin ya araç çubuğundaki listeden veya sağ fare tuşu ile ortaya çıkan bađlam menüsünün "Stiller" alt menüsünden istenen stilin seçilmesi ile yapılır. Her iki listenin elemanları kullanıcı tarafından tanımlanan Stil (*Style*) nesnelere

tümünü, ve hepsinden önce “<<Yok>>” isimli bir satırı içerir. Stil uygulanmış bir metin parçası seçili iken, stiller listesinden bu satırın seçilmesi varolan stilin kaldırılmasına yol açar. Eğer seçili metine herhangi bir stil uygulanmamışsa hiçbir şey olmaz. Aynı sonuç “Düzen” menüsünden “Stili Temizle” satırını seçerek de alınabilir. Eğer bağlam menüsü çağrıldığında hiçbir metin seçili değilse, “Stiller” menüsü altındaki satırlar ve araç çubuğundaki stil listesi işlevsiz haldedirler.

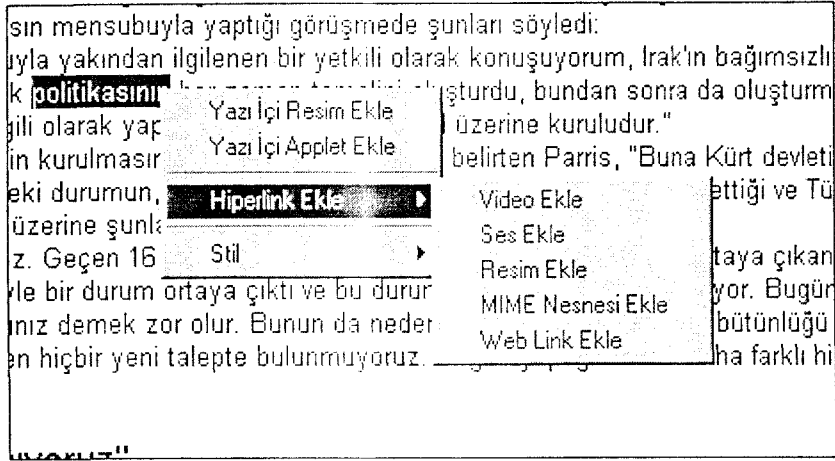




Şekil 5-28 : Sayfa Düzenleyicisi - stiller menüsü

Yazı içi nesnelere (applet'ler ve yazı içi resimler), imleci istenen yere götürüp bağlam menüsünü açıp, uygun menüyü seçerek eklenirler. Eğer bu işlem yapılmadan bir kısım metin seçili ise, nesne eklendiğinde seçili olan yazı silinir.

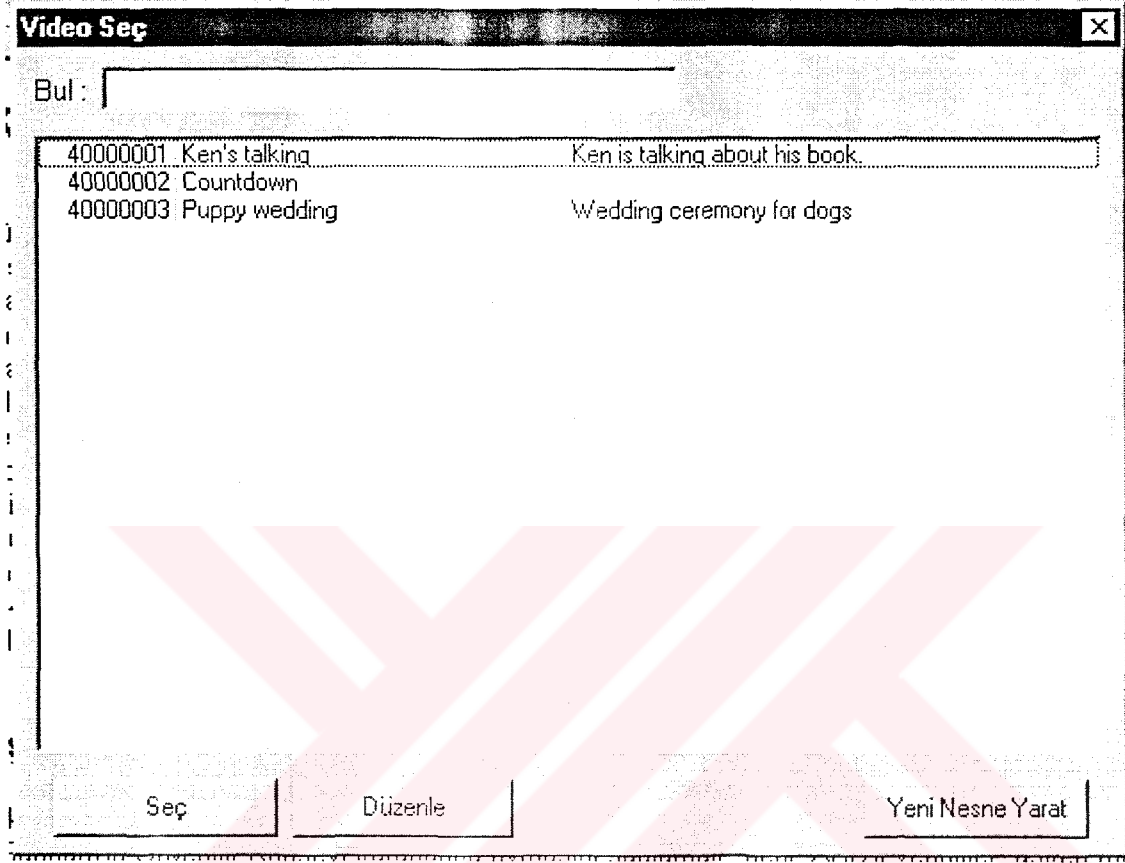
Veritabanındaki nesnelere bağlantılar, bağlam menüsünü açıp, "Hiperlink Ekle" menüsünden ilgili alt menü seçilerek yapılır. Ancak, bu işlem yapılmadan önce bir kısım metin seçili olmalıdır. Yoksa "Hiperlink Ekle" menüsünün alt menüleri işlevsiz durumda görünürler.



Şekil 5-29 : Sayfa Düzenleyicisi - hiperlink menüsü

Veritabanındaki bir nesneye bir hiperlink kurulmadan önce, kullanıcıdan bağlanılacak nesneyi belirlemesi beklenir. Bu amaçla, istenen türde (Resim, Video, Audio, İçerik, Web Bağlantısı) nesnelerin bir listesini içeren ilgili "Nesne Seçim Formu" açılır.

5.2.19 Nesne seçim formu



Şekil 5-30 : Nesne seçim formu – Video listesi

Yukarıdaki ekran Video nesneleri için açılan örnek bir ekrandır. Diğer nesnelere için de aynı ekran değişik liste elemanları ile kullanılır. Listenin üstündeki “İsim” alanı listede görünen elemanları sadece bu alana girilen isim ile başlayanlara sınırlamak için kullanılır. İstenilen nesne, ismi veya isminin bir bölümü bu alana girilerek bulunabilir.

“Seç” tuşu o anda listede seçili nesneyi bağlanılacak nesne olarak belirler. “Düzenle” ve “Yeni Nesne Yarat” tuşları kullanıcıya, seçili nesnenin düzenlenmesi ve yeni bir nesne yaratılıp ona bağlanması imkanını tanırlar.

Her iki tuş da aslında aynı formu açarlar. Ancak seçilen tuşa göre bu form, ya varolan bir kaydı getirir veya yeni, boş bir kayıt açar.

5.2.20 Video nesnesi formu

Video nesnesi formu, ilgili Video (*Video*) nesnesinin alanlarının düzenlenmesine imkan tanır.

OID :	<input type="text" value="400000001"/>	<input type="button" value="Kaydet"/>
İsim :	<input type="text" value="Ken's talking"/>	<input type="button" value="Vazgeç"/>
Genişlik :	<input type="text" value="200"/>	<input type="button" value="Karakteristik Vektörler"/>
Boy :	<input type="text" value="100"/>	<input type="button" value="İçerik"/>
Süre :	<input type="text" value="00:00:12"/>	
Tanım :	<input type="text" value="Ken is talking about his book."/>	
Anahtar Kelimeler :	<input type="text" value="Ken"/>	

Şekil 5-31 : Video nesnesi formu

“İçerik” tuşu, Video nesnesi ile ilgili İçerik nesnesinin düzenleneceği ekranı açar. Eğer, Video nesnesi ile ilişkilendirilmiş hiçbir İçerik nesnesi yoksa (nesne yeni yaratıldığında olduğu gibi), bu tuşun etiketi “Yeni İçerik” olarak değişir ve

bu durumda tuşa basılırsa, yeni bir içerik nesnesi yaratılıp Video nesnesi ile ilişkilendirilir.

Video nesnelere, Ses (*Audio*) nesnelere gibi, birden fazla karakteristik vektöre (*Feature Vector*) sahip olabilirler. Böylece, bu karakteristik vektörlerin düzenlenmeleri ayrı bir formdan yapılır. “Karakteristik Vektörler” tuşu “Video karakteristik vektörleri formu”nu açıp bunların düzenlenmesini sağlar.

5.2.21 Video karakteristik vektörleri formu

Video karakteristik vektörleri formu ile, bir Video nesnesi için tanımlanan vektörler düzenlenirler. Karakteristik vektörler (*Feature Vector*), Çokluortam Haberler Veritabanı sistemi içinde önceden tanımlanmış özel bir kullanıma sahip değildirler. Buna rağmen, ileride sisteme dahil edilebilecek CBR (Content Based Retrieval) özelliğinin kullanımı için uygulanmışlardır.

Karakteristik vektörlerin değerleri için bir kısıtlama olmadığından, hala, Video nesnesinin değişik bölümlerini tanımlamada kullanılabilirler. Örneğin, karakteristik vektör alanına, video ortamının değişik sahneleri ile ilgili tanımlayıcı anahtar kelimeler girilip bunlar üzerinden sorgulama yapılabilir. Böyle bir kullanımda ortamlar, en azından bir dereceye kadar içeriklerine bağlı olarak sorgulanmış olur.

Her karakteristik vektör dört alan içerir. Çerçeve numarası alanı karakteristik vektörün çıkarıldığı çerçeveyi gösterir. Karakteristik vektörler bir tek video çerçevesi için çıkarılabileceği gibi, bir video aralığı için de çıkartılabilirler. Bu durumda, çerçeve numarası yerine, karakteristik vektörün alındığı aralığın başlangıç ve bitiş noktaları, zaman olarak girilir.

Audio Ekle

OID : 50000001

İsim : Ken's singing.

Süre : 00:00:20

Tanım : Ken Miller is singing Jazz.

Anahtar Kelimeler : Ken Miller Jazz

Kaydet

Vazgeç

Karakteristik Vektörler

İçerik

Şekil 5-33 : Ses nesnesi formu

“İçerik” tuşu, Ses (*Audio*) nesnesi ile ilgili İçerik nesnesinin düzenleneceği ekranı açar. Eğer, Ses nesnesi ile ilişkilendirilmiş hiçbir İçerik nesnesi yoksa (nesne yeni yaratıldığında olduğu gibi), bu tuşun etiketi “Yeni İçerik” olarak değişir ve bu durumda tuşa basılırsa yeni bir içerik nesnesi yaratılıp Ses nesnesi ile ilişkilendirilir.

“Karakteristik Vektörler” tuşu “Ses karakteristik vektörleri formu”nu açıp bunların düzenlenmesini sağlar.

5.2.24 Resim nesnesi formu

Resim nesnelerinin açıklama alanları, Resim Nesnesi Formu ile düzenlenirler.

Resim Ekle

OID : 60000001

İsim : Mesut Yılmaz

Genişlik : 320

Boy : 200

Karakteristik Vektör :

Tanım : Snapshot of a sample app

Anahtar Kelimeler : resim

Kaydet

Vazgeç

İçerik

Şekil 5-35 : Resim nesnesi formu

Video ve Ses nesneleri gibi, Resim (*Image*) nesnesi de ortamın saklandığı bir İçerik (*Content*) nesnesine referans verir. “İçerik” tuşu yine ilgili İçerik nesnesine erişim için kullanılır ve davranışı Video ve Ses nesneleri için benzer formlarda anlatıldığı gibidir.

Resim verisi zamanla deęişmedięinden, veritabanında her Resim nesnesi için bir karakteristik vektör tutulur.

Video, Ses ve Resim nesnelerinin İçerik nesnelere ile bağlantılarında, şunu hatırlatmak gerekir ki böyle bir ilişki bir kez kurulduğunda bozulamaz. Yani Video, Ses veya Resmin referans ettiği İçerik nesnesi deęiştirilebilmesine rağmen referans deęiştirilemez veya silinemez.

5.2.25 İçerik nesnesi formu

İçerik nesnesi formu, sistemdeki bir İçerik (*Content*) nesnesinin açıklama ve ortam bilgisinin düzenlenmesine izin verir.



İçerik	
OID :	900000037
İsim :	Puppy wedding
Tanım :	Wedding ceremony for two dogs
Saklama Biçimi	
<input type="radio"/>	Dosya Sisteminde Sakla
Yol : <input type="text"/>	
<input checked="" type="radio"/>	Veritabanında Sakla
MIME Tipi :	video/quicktime
Veritabanında içerik var <input type="checkbox"/>	<input type="button" value="Dosyadan Yeni İçerik Yükle"/>
İçerik Boyu :	609202 Byte
Arşivleme	
Oda :	VD02
Bölüm :	IR3
Raf :	<input type="text"/>
Satır :	23
Sütun :	<input type="text"/>
Sıra :	<input type="text"/>
Ortam :	Teyp

Şekil 5-36 : İçerik nesnesi formu

İçerik nesnesinin sahip olduğu ortam bilgisi iki şekilde saklanabilir,

- bir işletim sistemi dosyası olarak veya
- veritabanında bir BLOB alanı içinde.

Eğer bilgi bir dosyada saklanacaksa, “Dosya Sisteminde Sakla” seçeneği seçilmelidir. Bu durumda, dosya ismi “Yol” alanında verilir. Burada girilen yol, bir sanal dizin yoludur ve yapılandırma ayarları dosyasında (mmnews.ini) “[Publishing]” seçeneği altında “ContentDir” anahtarı ile verilen yola göre görelidir. Bu depolama tiplerinin, içeriğin istemciye yollanma şeklini nasıl etkilediğini görmek için “Sayfa Düzenleyicisi” ve “Veri Yayıncısı” bölümlerine bakınız.

Diğer bir depolama seçeneği, kullanıcılara ortam verisini BLOB alanında saklama imkanı verir. Bunun için, “Veritabanında Sakla” seçeneği seçilir. Ortam verisi BLOB alanında saklandığında, verinin MIME tipi de girilmek zorundadır. Bu tip bilgisi, daha sonra oluşturulan HTTP cevabının başlığındaki “Content Type” alanında kullanılır. Bu alan aslında bir MIME Tipi (*MIME Type*) nesnesine referanstır.

BLOB alanındaki veri, “Dosyadan Yeni İçerik Yükle” tuşuna basıp listeden bir dosya seçerek, bu dosyanın içeriği ile günlenebilir. “Veritabanında içerik var” etiketli checkbox kontrolü, veritabanındaki BLOB alanında veri olup olmadığını gösterir. “İçerik Boyu” ise, BLOB alanındaki verinin sekizli sayısı cinsinden boyunu gösterir.

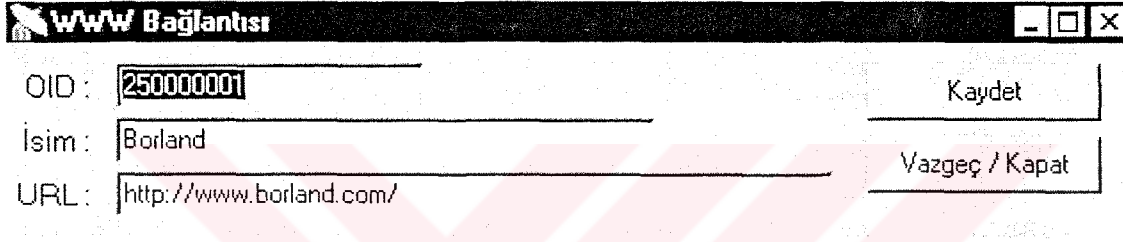
Şuna dikkat edilmelidir ki “Saklama Biçimi” grubundaki seçeneklerden biri seçildiğinde, diğer seçeneğe ilişkin tüm alanlardaki bilgiler sıfırlanır. Yani, “Veritabanında Sakla” seçeneği seçildiğinde, “Yol” alanı, “Dosya Sisteminde Sakla” seçeneği seçildiğinde ise “MIME Tipi” ve ortamın saklandığı BLOB alanları sıfırlanır.

İçeriğin temsil ettiği ortamın sistem dışı saklanması ile ilgili bilgi içeren “Arşivleme” grubu, aslında İçerik nesnesinden referans verilen Arşiv

nesnesinin alanlarını gösterir. Eğer referans verilen bir Arşiv nesnesi yoksa, bu grubun yerine yeni Arşiv nesnesi yaratılmasını söyleyen bir tuş belirir. İçerik nesnelere verilen Arşiv nesnesi referansları da değiştirilemez ve silinemezler.

5.2.26 Web bağlantısı nesne formu

Web Bağlantısı (*Web Link*) nesnelere bir URL'yi tanımlarlar. Bu URL'ler ise, WWW üzerinde herhangi bir kaynağa erişim için kullanılırlar. Web bağlantısı nesne formu, bir Web Bağlantısı'nın özelliklerinin düzenlenebilmesi için kullanılır.



OID :	<input type="text" value="250000001"/>	Kaydet
İsim :	<input type="text" value="Borland"/>	Vazgeç / Kapat
URL :	<input type="text" value="http://www.borland.com/"/>	

Şekil 5-37 : Web bağlantısı nesne formu

5.3 Veri Yayıncısı

Uygulama programının ikinci kısmı, standart WWW gezgini ile Çokluortam Haberler sitesini ziyaret eden kullanıcı ile, veritabanında depolanan bilgi arasında bir arabirim olarak çalışan, arabirim programı olarak tasarlanmıştır. Kullanıcıdan gelen sorgulama istekleri, bu sistem tarafından algılanır ve sonuçlar, yani oluşturulan çokluortam belgeleri ve/veya çokluortam nesneler, HTML (HyperText Markup Language) sayfaları halinde kullanıcıya gönderilir. Böylece, Çokluortam Haberler Veritabanı, WWW üzerinde yayınlanmış olur. Kullanıcılar, Çokluortam Haberler sitesine bağlanıp buradaki bilgiye erişebilirler.

Veri Yayıncısı da aslında, her biri önemli bir fonksiyonu gerçekleştiren iki bölümden oluşur : *Sorgu Yönetici* WWW üzerinden gelen sorgu isteklerini algılayıp bunları veritabanına gönderir ve *HTML Üretici* sorgu yöneticinin veritabanına gönderdiği sorguların sonucu olarak, veritabanından gelen bilgi ile HTML sayfalarını üretir.

Veri Yayıncısı programı, "mmnews.dll" adında bir süreç-içi dinamik bağlama kütüphanesidir (Dynamic Link Library - DLL). Süreç-içi servisler, WWW sunucu servisinin adres uzayında, her bağlantı için ayrı sicimler (thread) halinde çalışırlar. Bu nedenle, hatalara karşı daha hassastırlar ve hata ayıklamaları daha zordur. Böylece, geliştirme için kullanılan WWW sunucunun da süreç-içi kütüphaneleri desteklemesi gerekmektedir.

Bundan başka, geliştirme süreci, diğer geliştirme araçları ile birlikte aynı makinada çalışabilen bir WWW sunucu seçilmesi ile büyük oranda kolaylaştırılabilecektir.

Yukarıda sayılan sebeplerle, Çokluortam Haberler Sisteminin geliştirilmesinde Microsoft® Internet Information Server® kullanılmıştır.

Genel uygulamalar için, HTTP 1.0 veya üstünü destekleyen ve süreç-içi kütüphanelerin kullanımına izin veren herhangi WWW sunucu kullanılabilir.

Şu da hatırlatılmalıdır ki, “Öneriler” bölümünde önerilen bazı özellikler HTTP 1.1 desteği gerektirmektedirler.

5.3.1 Mimari

Bu bölümde, Veri Yayıncısı programının geliştirilmesinde kullanılan dil bileşenleri ve işlevleri anlatılmıştır.

5.3.1.1 Web modülü

Veri Yayıncısı programı, birçok görsel olmayan (non-visual) bileşen içeren bir “Web Modülü” nesnesi içerir. Web Modülü, HTTP mesajlarını sockets arabiriminden almak ve yollamak için bir “Web Dispatcher” içerir. “Web Dispatcher”, HTTP istek mesajlarını, uygun eylem işleyicilere gönderip bir cevap oluşturulmasını sağlar. Eylem işleyiciler, cevap verdikleri HTTP mesaj tiplerine (POST, GET vs.) ve yol bilgisine göre tanımlanırlar. Web Dispatcher, mesaj tipi ve yol bilgisi tanımları gelen mesajınkine eşit olan eylem işleyicilerini tetikler. Her eylem işleyici, daha sonra cevap mesajının içeriğine katkısını yapar. Eğer, mesaja uygun eylem işleyici bulunamazsa, varsayılan işleyici cevabı oluşturmak üzere çağırılır. Eğer eylem işleyicilerinin taranması sonunda cevap gövdesi (içeriği) oluşturulmamış ise, istemciye hiçbir şey gönderilmez.

5.3.1.2 Veritabanı

Veritabanı bileşeni de, diğerleri gibi, Web Modülü’nün içindedir. Bu bileşen veritabanına özel bilgileri tutar ve veritabanının BDE’de (Borland Database Engine).tanımlı bir veritabanı tanımı ile ilişkilendirilmesini sağlar.

5.3.1.3 Oturum

Oturum nesnesi, program sicimlerinin (thread) veritabanına yaptıkları bağlantıları yönetir. Yeni bir sicim yaratıldığında, Oturum nesnesi otomatik olarak bağlantı havuzundan bir bağlantıyı sicime atar. Eğer havuzda boş bağlantı yok ise, Oturum nesnesi yeni bağlantı yaratır. Oturum nesnesi, ayrıca,

havuzda boş veya kullanılan, belirli sayıda bağlantının bulunmasından da sorumludur.

5.3.1.4 Tablolar ve sorgular

Web Modülü içindeki tablo ve sorgu bileşenleri, veritabanındaki tablolara erişime imkan tanırlar. Kullanıcı sicimi yaratıldığında, tüm tablolar açılır ve aynı sicimin daha sonraki isteklerinde kullanılırlar. Bu yöntem, kaynakların etkin kullanımı ve hız arasında, hızdan yana bir tercihi belirler. Çünkü, tablo açıp kapamada geçen zaman büyük ölçüde bertaraf edilmiş, buna karşılık sicimin yaşam süresi boyunca, kullanılması bile, açık olan tabloların harcadıkları kaynaklar gözardı edilmiştir.

5.3.1.5 Sayfa üreticileri

Sayfa üreticileri, bir giriş şablonundan HTML komut katarları üretirler. Bir başka deyişle, HTML şablonunun, Internet gezgini gibi bir istemci uygulama tarafından algılanabilecek HTML komutları katarına çevirirler. HTML şablonu, HTML komutları ve “OnHTMLTag” olayı tarafından gerçek içerikle değiştirilen, HTML şeffaf etiketler içerirler. Bir sayfa üreticinin “Content” özelliği okunduğunda, şablon HTML dosya, “OnHTMLTag” olayında HTML-şeffaf sayfa üretici etiketleri, gerçek değerleri ile değiştirilmiş olarak döndürülür. Web Modülü, istemcilere gönderilen her çeşit sayfa için bir sayfa üretici içerir.

5.3.2 Çok-sicimli bağlantı yönetimi

Yukarıda değinildiği gibi, Veri Yayıncısı WWW sunucu sürecinin adres uzayı içinde bir süreç-içi sunucu olarak çalışır. Siteye yapılan her kullanıcı bağlantısı için yeni bir Veri Yayıncısı sicimi yaratılır. Daha sonra, her Veri Yayıncısı sicimi de veritabanına yeni bir bağlantı yaratır veya havuzdan bir bağlantı alır.

Bu işlev Oturum bileşeninin “AutoSession” özelliği kullanılarak sağlanır.

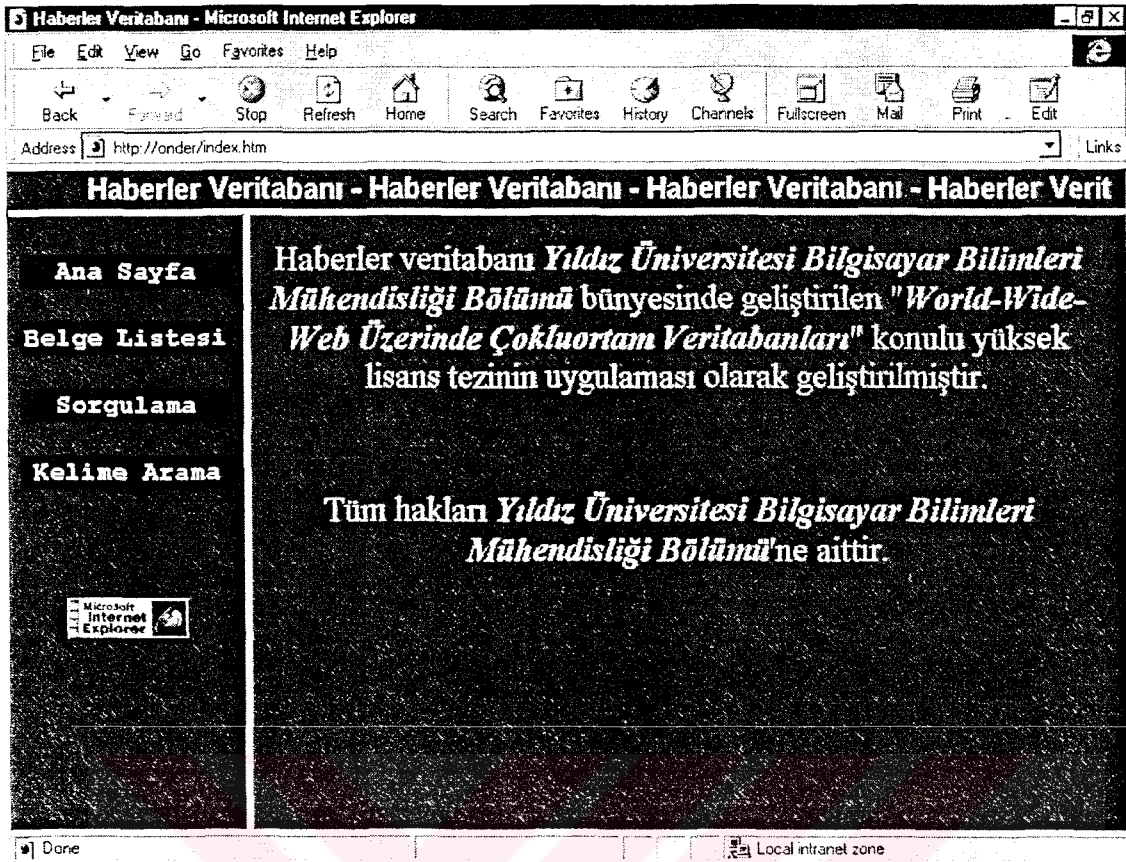
5.3.3 WWW sitesi yönetimi

Çokluortam Haberler sitesi, hemen hemen tamamen, Veri Yayıncısı programı tarafından yönetilir. Buna tek istisna sitenin ana sayfasıdır (index.htm). Sitenin ana sayfası üç çerçeveden oluşur. İlk çerçeve en üstteki başlık çerçevesidir. Burada uygulamanın ismini gösteren bir “marquee” ActiveX nesnesi kullanılmıştır.

Soldaki ikinci çerçeve ise, sitenin fonksiyonları için bir ana menü görevini görür. Bu çerçeve “Contents.htm” adlı HTML sayfasını içerir. Her ana fonksiyon, bir “hover button” nesnesine bağlanmıştır. Kullanıcı, bir menü tuşunu seçtiğinde Veri Yayıncısı tarafından yeni bir sayfa üretilip orta çerçeveye gönderilir.

Orta çerçeve, Veri Yayıncısı’na yapılan istekler üzerine HTML sayfaları halinde gönderilen sonuçlarının gösterilmesi için kullanılır.





Şekil 5-38 : Çokluortam haberler sitesi - ana sayfa

Çokluortam Haberler sitesi dört ana fonksiyon içerir,

- Ana Sayfa** : Site ve telif hakkı bilgilerinin bulunduğu bir sayfa üretir. Bu, aynı zamanda, Veri Yayıncısı için varsayılan eylemdir. Yani istemciden gelen bir istek anlaşılmazsa, Veri Yayıncısı ana sayfa istenmiş gibi davranır.
- Belge Listesi** : İçinde sistem kök dizinindeki belge ve klasörlerin listelendiği bir HTML sayfa üretir. Bu listedeki her eleman da, eğer klasör ise o klasörün içerdiği belge ve klasörlerin bir listesine; eğer belge ise o belgeye ait açıklama bilgilerinin listelendiği birer HTML sayfası için Veri Yayıncısı'ndan yapılacak istekleri tetiklerler. Bu istekler URL bağlantıları şeklindedir.

Sorgulama : Kullanıcıya, belge ve çokluortam nesnelərini, açıklama alanları ile sorgulayacağı bir HTML sayfa gönderir.

Kelime Arama : Kullanıcıya, belge sayfalarında bulunan kelimeler arasında arama yapabileceği bir HTML sayfa gönderir.

5.3.4 WWW sunucu etkileşimi

Programların WWW sunucu ile olan ilişkileri :temel olarak iki biçimde olabilir.

5.3.4.1 Süreç-dışı etkileşim

CGI (Common Gateway Interface)'deki gibi, süreç-dışı etkileşim kullanıldığında programın nesne kodu, kullanıcı isteği üzerine sunucu servisi tarafından kendi adres uzayına sahip bir süreç olarak çalıştırılır. Sunucu aynı zamanda bu sürece parametreler de gönderir.

Bu yöntemin iki önemli faydası vardır. Öncelikle, yeni süreç kendi adres uzayına sahip olduğundan, programdaki hatalar sunucunun işleyişini etkilemezler. İkinci olarak, WWW sunucunun tipi, işletim sistemi veya programlama tekniklerine bağımlılık yoktur. Aynı süreçler olarak çalışan iki program, standart giriş/çıkış'a (veya WinCGI'daki gibi belirli biçime sahip bir dosyaya) yazarak ve oradan okuyarak haberleşirler. Sonuç olarak, sunucu ile etkileşecek böyle bir programın yazımı gayet kolaydır. Bundan başka, geliştirilen program değişik işletim sistemi ve donanım platformlarına en az değişiklikle veya hiç değişiklik yapmadan taşınabilirler.

Öte yandan, süreç-dışı programları çok önemli bir dezavantajı vardır. Program kullanıcı isteği ile çalıştırıldığından, her yeni kullanıcı isteği için aynı program tekrar yeni bir süreç olarak başlatılır, çalıştırılır ve yok edilir. Böylece süreç yaratma ve yok etme işlemleri sisteme ağır bir yük getirirler. Üstelik, bir çok kullanıcının aynı anda siteye bağlanmaları durumunda, aynı programın birçok varlığı kaynakları ziyan eder ve bir süre sonra kaynak yetersizliğinden isteklere cevap veremez hale gelebilir.

5.3.4.2 Süreç -içi etkileşim

Süreç-dışı etkileşim yöntemine bir alternatif olarak süreç-içi etkileşim kullanılabilir. Süreç-içi etkileşim programcının, sunucunun adres uzayında çalışacak sicimler yaratmasına olanak tanır. Sicimler sunucu tarafından ve bazı API (Application Programming Interface) fonksiyonları kullanılarak yaratıldıklarından, bu metod, işletim sisteminin dinamik bağlanmaya (dynamic linking) izin vermesini gerektirir. Süreç-içi etkileşim, Microsoft®'un Internet Information Server®'ı (IIS) veya Netscape®'in Netscape Server®'ında kullanılabilir

Bu tekniğin, süreç-dışı etkileşime göre önemli bir üstünlüğü de sicimlerin sunucu tarafından yaratılıp yönetilmeleri ile ayrı süreçler oluşturulmaması ve kullanıcı isteklerine cevap verirken, sistemi büyük bir yükten kurtarmış olmasıdır.

Öte yandan, sicimler sunucu ile aynı adres uzayını kullandıklarından, programdaki küçük bir hata WWW sunucu sürecinin çökmesine sebep olabilir. Aynı sebepten, bu tip proramlarda hata ayıklama daha zordur. Son olarak, program özel bir API kullanılarak geliştirileceğinden bu API yi desteklemeyen sunuclarda çalışmaz.

5.3.5 Sorgu yönetici

Uygulamanın bu bölümü kullanıcıdan gelen istekleri algılayıp, istenen veriyi veritabanından alır.

Kullanıcılar, Sorgu Yönetici'den iki parametre ile bilgi isterler. İlk parametre getirilmesi istenen nesnenin veya yapılması istenen işlevin tipidir. Eğer istenilen bir nesne ise, ikinci parametre nesneyi sistemde tek olarak belirleyen nesne belirteci (OID)'dir.

Normal olarak, OID bir nesneyi sistem bazında zaten tanımlar. Bu yüzden, aslında, yukarıda sözü edilen tip bilgisine gerek yoktur. Ancak, tip bilgisinin

eklenmesi ile veritabanına yapılacak isteklerden beklenen bilgi, sadece istek cümlesine bakarak ta anlaşılabilir.

İstek tipi bilgisi Veri Yayıncısı'na, HTTP istek mesajının Yol Bilgisi (Path Information) alanında gönderilir. OID bilgisi ise istek parametreleri (Request Parameters) kısmındadır. Sonuç olarak, örneğin bir Belge için sorgulama isteği,

/mmnews.dll/Documents?DocOID=20000003

biçimindedir. Bu örnekte, "Documents" Yol Bilgisi alanında geçilen nesne tipi, "DocOID" istek parametresi adı ve "20000003" istek parametresi değeridir.

Nesne tipi etiketleri ve karşılık gelen parametre isimleri aşağıdaki tabloda tetikledikleri eylemlerle birlikte verilmişlerdir.

Tablo 5-2 : Veri Yayıncısı eylemleri

Nesne Tipi (Yol Bilgisi)	Parametre Adı	Eylem
/Folders	FolderOID	İçinde buldukları klasörün OID'i parametre değeri ile verilen tüm dosyaların ve klasörlerin listesini al. Parametre değeri verilmemiş veya "-1" verilmiş ise kök klasörünün içeriği getirilir.
/Documents	DocOID	OID'i parametre değeri ile verilen Belge (<i>Document</i>) nesnesinin bilgilerini getir.
/Pages	PageOID	OID'i parametre değeri ile verilen belge Sayfa'sını (<i>Page</i>) getir. .
/Videos	VideoOID	OID'i parametre değeri ile verilen Video (<i>Video</i>) nesnesinin bilgilerini getir.

Nesne Tipi (Yol Bilgisi)	Parametre Adı	Eylem
/Audios	AudioOID	OID'I parametre değeri ile verilen Ses (<i>Audio</i>) nesnesinin bilgilerini getir.
/Images	ImageOID	OID'i parametre değeri ile verilen Resim (<i>Image</i>) nesnesinin bilgilerini getir.
/Content	ContentOID	OID'i parametre değeri ile verilen İçerik (<i>Content</i>) nesnesinin bilgilerini getir.
/Play	ContentOID	OIDi parametre değeri ile verilen İçerik nesnesinin ortam bilgisini getir.
/QueryDoc		<p>Form bilgisinde verilen şartları sağlayan Belge nesnelerinin listesini getir. Bu istek sadece eylem tipi "POST" olan bir formdan yollanırsa geçerlidir. Aksi halde, sözü edilen formu da içeren sorgulama ekranı istemciye yollanır. Form değişkenleri ve manaları aşağıda verilmiştir.</p> <p>DocHdr : Belge başlığı</p> <p>DocName : Belge adı</p> <p>DocDate : Belge tarihi</p> <p>DocType : Belge'nin haber tipi</p> <p>DocSubj : Belge'nin haber konusu</p> <p>DocCpr : Telif hakkı sahibinin adı</p> <p>DocPub : Yayıncının adı</p> <p>DocPDate : Haberin yayınlanma tarihi</p> <p>DocStart : Belge'nin başlangıcı</p>

Nesne Tipi (Yol Bilgisi)	Parametre Adı	Eylem
		DocEnd : Belge'nin sonu DocImp : Önem seviyesi DocSum : Haberin özeti DocKw : Anahtar kelimeler DARRoom : Arşiv odası DARSEct : Arşiv bölümü DARShf : Arşiv rafı DARRow : Arşiv satırı DARCol : Arşiv sütunu DAROrd : Arşiv sırası DARMed : Arşiv saklama ortamı DRPPERName : İlgili kişi adı DRPRole : İlgili kişi rolü DLOName : İlgili yer adı DLODist : İlgili yer ilçesi DLOState : İlgili yer ili DLOReg : İlgili yer bölgesi DLOCnt : İlgili yer ülke
/QueryMedia		Form bilgisinde verilen şartları sağlayan çokluortam nesnelere listesini getir. Bu istek sadece eylem tipi "POST" olan bir formdan yollanırsa geçerlidir. Aksi halde, sözü edilen formu da içeren sorgulama

Nesne Tipi (Yol Bilgisi)	Parametre Adı	Eylem
		<p>ekranı istemciye yollanır. Form değişkenleri ve manaları aşağıda verilmiştir.</p> <p>MType : Sorgulanacak çokluortam nesnesi tip(ler)I</p> <p>MName : Nesne adı</p> <p>MDesc : Nesne tanımı</p> <p>MWid : Nesne genişliği</p> <p>MHgt : Nesne boyu</p> <p>MLen : Nesne süresi</p> <p>MFVec : Nesne'nin karakteristik vektörü</p> <p>MKWords :Nesne'nin anahtar kelimeleri</p> <p>MARRoom : Arşiv oda</p> <p>MARSect : Arşiv bölüm</p> <p>MARShf : Arşiv raf</p> <p>MARRow : Arşiv satır</p> <p>MARCol : Arşiv sütun</p> <p>MAROrd : Arşiv sıra</p> <p>MARMed : Arşiv saklam ortamı</p>
/QueryWord		Belge sayfalarında, form bilgisinde verilen kelimeleri ara ve bunları içeren belgeleri getir.

Nesne Tipi (Yol Bilgisi)	Parametre Adı	Eylem
		<p>Bu istek sadece eylem tipi "POST" olan bir formdan yollanırsa geçerlidir. Aksi halde, sözü edilen formu da içeren sorgulama ekranı istemciye yollanır. Form değişkenleri ve manaları aşağıda verilmiştir.</p> <p>Words : Belge sayfaları içinde aranacak kelimeler.</p>

5.3.6 HTML üretici

HTML Üretici, sonuçları HTTP 1.0 cevap mesajları içinde, ürettiği HTML biçiminde sayfalarla istemciye gönderir. Birçok mesaj, HTML sayfalar içermesine rağmen, ortam verisinin iletildiği durumlarda video ve ses gibi değişik MIME tipine sahip bilgi de gönderilir.

HTML Üretici, kullanıcı istekleri üzerine, Sorgu Yönetici'nin veritabanından aldığı veriyi biçimleyip cevap olarak istemciye geri gönderir. Üretilen sayfalar bir nesneye ait açıklama bilgilerini (Belge ekranı) veya bir nesne listesini (klasör listesi, Belge veya çokluortam nesne sorgulama sonuçları) içerirler veya bir giriş ekranıdır (sorgulama ekranları).

HTML Üretici istemciye gönderilebilen her tip sayfa için bir sayfa üretici kullanır. Sayfa üreticileri HTML-şeffaf etiketleri ile birlikte bir HTML şablonu içerirler. İstemciye gönderilmeden önce HTML-şeffaf etiketlerin yerine anlamlı bilgi yerleştirilir.

HTML Üretici tarafından üretilen sayfalar Microsoft® Internet Explorer® 4.x ile birlikte gelen dekorasyon nesnelere ve olanaklarını kullanırlar. Tüm sayfalarda standart bir görünüm elde etmek için "Microsoft Theme Control"

kullanılmıştır. “Hover button”lar içindekiler çerçevesinde ve “marquee control”de başlık çerçevesinde kullanılmışlardır.

Buna rağmen, istenen kontrollerin bulunmadığı sitelerde bile, internet gezgini kaskad stil sayfalarını (Cascading Style Sheets - CSS) desteklediği sürece site resim ve dekorasyon eksiklikleri ile, ama işlevsel olarak doğru kullanılabilir.

HTML sayfalarında buldukları tüm bağlantıları takip ettiklerinden, HTML Üretici, Internet robotların siteyi ziyaret etmesini önlemek için, üretilen her sayfaya “NO-ROBOTS” META açıklamasını koyarlar.

HTML Üretici tarafından oluşturulan sayfalar tamamen genel standartlara uygun olarak ve hiçbir ek özelliğe ihtiyaç duymayacak şekilde oluşturulurlar. Bir başka deyişle, “text/html” tipli sayfalar katıştırılmış özel nesnelere içermezler. Özel nesnelere gönderileceği zaman ayrı bir cevap mesajı ile ve ilgili MIME tipi atanarak gönderilirler.

Üretilen sayfalarda sadece MIME tipleri kullanılır. Ancak kullanıcılar Web Bağlantısı (*Web Link*) nesnelere ile veya dosya sisteminde saklama seçeneği kullanılan İçerik (*Content*) nesnelere ile istedikleri internet kaynağına erişimi sağlayabilirler.

Sonuç olarak, Çokluortam Haberler sitesinin kullanımı için özel bir bileşene ihtiyaç yoktur denilir.

Basitçe, Sorgu Yöneticisi kullanıcı isteği ile veriyi veritabanından alır, ve sonra HTML Üretici sayfa üreticindeki etiketleri, veritabanından alınan bilgi ile değiştirerek cevap sayfasını oluşturur. Bu durum, hepsi için olmasa bile, birçok sayfa için geçerlidir.

Eğer, Sorgu Yöneticisi istenen nesneyi bulamazsa, istemciye, kullanıcıyı hatadan haberdar eden bir hata sayfası gönderilir.

Belge (*Document*) Sayfa’larının (*Page*) HTML olarak oluşturulması, Metin Açıklamaları’nın (*Text Attributes*) HTML’e çevrilmesini gerektirdiğinden, yukarıdakilerden daha karmaşık işlemler içerir. Metin Açıklamaları beş değişik şekilde HTML’e çevrilirler.

1. Paragraf açıklamaları, HTML paragraf etiketlerine (<P>) çevrilirler.
2. Stil (*Style*) tanımlamalarında kullanılan HTML başlangıç ve son etiketleri, Stil'lerin HTML'e çevrilmesinde, stili içeren metin parçasının başına ve sonuna konulurlar.
3. Yazı-içi nesnelere, ya HTML resim () veya applet (<APPLET CODE="xxx">) etiketleri içinde gösterilirler. Kullanılacak HTML etiketine (IMG veya APPLET) dosyanın uzantısına bakılarak karar verilir. ".class" uzantılı dosyalar "APPLET" etiketi ile, tüm diğerleri ise IMG etiketi ile kullanılırlar.

Şu da belirtilmelidir ki, Java applet parametreleri desteklenmemektedir. Eğer parametrelili Java applet'leri kullanılacaksa, bunların her değişik parametre değeri için bir başka, parametresiz applet yazılmalı ve o applet'ler kullanılmalıdır.

4. Çokluortam nesnelere (Video, Ses, Resim ve İçerik) bağlantılar, Veri Yayınlayıcıdan, ilgili nesnelere isteyen sorgu istekleri olarak HTML'e katılırlar. Örnek olarak "Süleyman Demirel" metin parçasının Süleyman Demirel'in portresini içeren bir Resim nesnesine bağlandığını düşünelim. Bu durumda üretilen HTML sayfasındaki bağlantı şuna benzer,

```
<A HREF="/mmnews.dll/Images?ImageOID=60000001"> Süleyman  
Demirel</A>
```

Bu bağlantılar hakkında daha geniş bilgi için "Sorgu Yöneticisi".başlığına bakınız.

5. Metin içerisindeki Web Bağlantısı nesnelere referanslar, HTML "anchor" etiketleri (<A>) ile Web Bağlantısı nesnesinin temsil ettiği URL'ye bir bağlantı içerirler. Örneğin, "Borland International" metni alanları aşağıda verilen bir Web Bağlantısı nesnesine referans versin.

OID : 25000001

Name : Borland

URL : www.borland.com

Bu durumda sayfanın HTML görünümünde,

```
<A HREF=www.borland.com> Borland International </A>
```

ifadesi yer alır.

5.3.6.1 Ana sayfa

WWW sitesinin ana sayfası, üç çerçeveden oluşan bir çerçeve kümesi olarak tasarlanmıştır. Üstteki başlık çerçevesi uygulama adını gösterir, soldaki içerik çerçevesinde ise sitenin ana fonksiyonlarına erişen menü vardır. Ortadaki üçüncü çerçeve Veri Yayıncısı'ndan gelen cevapların görüldüğü yerdir. Ana sayfa (index.htm), içerik sayfası (contents.htm) ve başlık sayfası (banner.htm) sitede zaten dosya olarak vardır ve Veri Yayıncısı tarafından üretilmezler

5.3.6.2 Belge listesit

Belge listesi Veri Yayıncısı tarafından, klasör sorgulama isteklerine cevap olarak üretilir. OID'i istek mesajında verilen klasör içindeki klasörleri ve belgeleri arka arkaya listeler.

Belge listesindeki her klasör satırı, o klasörün içindeki belge ve klasörlerin istendiği sorguya bir istek; her belge satırı ise belgenin açıklama bilgilerinin getirildiği sorguya bir istek bağlantısı içerirler. Eğer listelenen klasör kök klasörü değilse, bu listenin en üstünde, bir üst klasöre çıkış için de bir bağlantı bulunur.

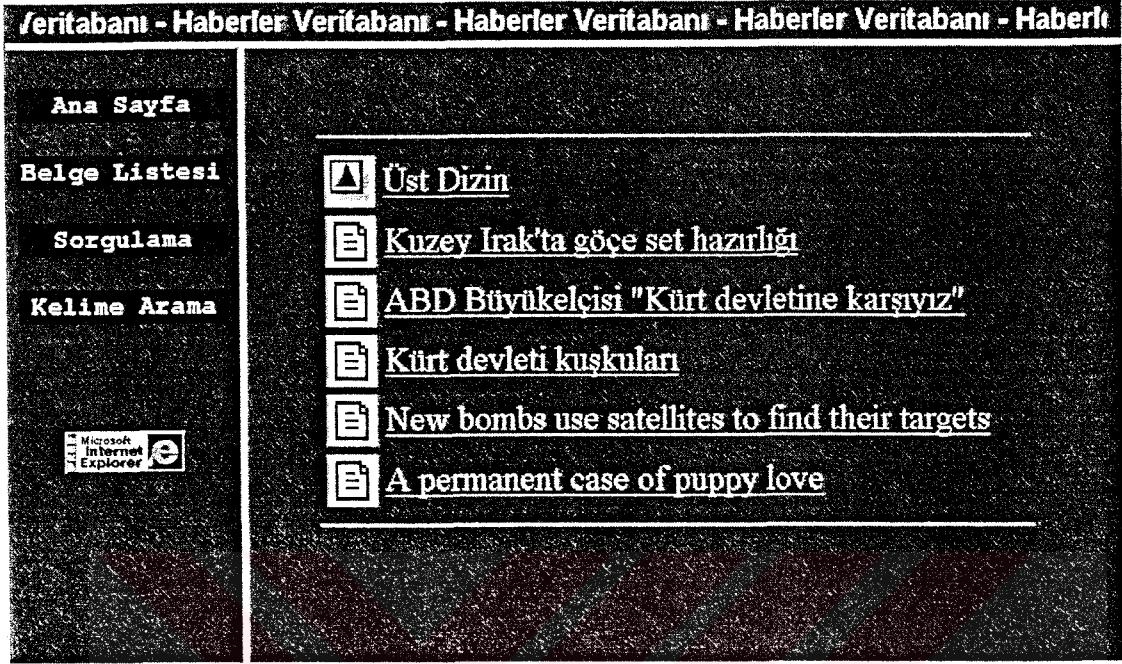
Örnek bir klasör sorgu cümlesi şöyle olabilir.

```
/mmnews.dll/Folders?FolderOID=10000001.
```

Bu istek üzerine Veri Yayıncısı, istemciye OID'i "10000001" olan klasör altındaki tüm klasörleri ve belgeleri içeren bir sayfa gönderir.

Eğer "FolderOID" parametresi verilmez veya "-1" verilirse, Veri Yayıncısı kök klasörünün içeriğini döndürür.

Eğer istenen OID'e sahip bir klasör nesnesi bulunamazsa, istemciye bir hata sayfası döndürülür.



Şekil 5-39 : Belge listesi

5.3.6.3 Belge açıklamaları

Belge açıklamaları sayfası belge sorgulamalarının sonucu olarak döndürülürler. Bu sayfada Belge'ye ilişkin açıklama bilgilerinin yanısıra, ilgili Belge'lere ve Belge Sayfa'larına da bağlantılar vardır.

Bir Belge sorgusu aşağıdaki biçimde olabilir,

/mmnews.dll/Documents?DocOID=20000001

Burada OID'i "20000001" olan Belge'nin bilgileri istenmektedir.

Bu sorgu sonucu gönderilen açıklama bilgileri şunlardır:

- Belge başlığı.

- Belge OID'i.
- Belge adı.
- Belge tarihi.
- Haber tipi.
- Haber konusu.
- Haberin kaynağı.
- Haberin telif hakkı sahibi.
- Haberin yayınlandığı yer.
- Haberin yayınlandığı tarih.
- Haberin başlangıcı.
- Haberin sonu.
- Haberin önem seviyesi.
- Haberin özeti.
- Anahtar kelimeler.
- Haberin arşivde saklama bilgisi. Bu bilgi arşiv adresinin Oda, Bölüm, Raf, Satır, Sütun ve Sıra belirteçleri ile Ortamının Tipi bilgisini içerir.
- Haber ile ilgili kişilerin , haberdeki rolleri ile birlikte, listesi.
- İlgili Belgelerin isimlerinin bir listesi. Bu liste aynı zamanda ilgili Belgelere bağlantıları da içermektedir.
- İlgili coğrafi bölgeye ait bilgiler. Bu bilgi de referans verilen Coğrafi Bölge nesnesinin OID, İsim, İlçe, İl, Ülke ve Bölge alanlarını içerir.

ifabarı - Haberler Veritabanı - Haberler Veritabanı - Haberler Veritabanı - Haberler V																									
Ana Sayfa	<h2>Kuzey Irak'ta göçe set hazırlığı</h2> <table border="1"> <tr> <td>Belge OID :</td> <td>20000005</td> </tr> <tr> <td>İsim :</td> <td>Kuzey Irak'ta göçe set hazırlığı</td> </tr> <tr> <td>Tarih :</td> <td></td> </tr> <tr> <td>Açıklama :</td> <td></td> </tr> <tr> <td>Kaynak :</td> <td>Hürriyet Gazetesi</td> </tr> <tr> <td>Telif :</td> <td>Hürriyet Gazetesi</td> </tr> <tr> <td>Yayımlandığı Yer :</td> <td>Hürriyet Gazetesi</td> </tr> <tr> <td>Yayın Tarihi :</td> <td></td> </tr> <tr> <td>Başlangıç :</td> <td></td> </tr> <tr> <td>Bitiş :</td> <td></td> </tr> <tr> <td>Tip :</td> <td>G - Haberler</td> </tr> <tr> <td>Konu :</td> <td>Dış İlişkiler</td> </tr> </table>	Belge OID :	20000005	İsim :	Kuzey Irak'ta göçe set hazırlığı	Tarih :		Açıklama :		Kaynak :	Hürriyet Gazetesi	Telif :	Hürriyet Gazetesi	Yayımlandığı Yer :	Hürriyet Gazetesi	Yayın Tarihi :		Başlangıç :		Bitiş :		Tip :	G - Haberler	Konu :	Dış İlişkiler
Belge OID :		20000005																							
İsim :		Kuzey Irak'ta göçe set hazırlığı																							
Tarih :																									
Açıklama :																									
Kaynak :		Hürriyet Gazetesi																							
Telif :		Hürriyet Gazetesi																							
Yayımlandığı Yer :		Hürriyet Gazetesi																							
Yayın Tarihi :																									
Başlangıç :																									
Bitiş :																									
Tip :	G - Haberler																								
Konu :	Dış İlişkiler																								
Belge Listesi																									
Sorgulama																									
Kelime Arama																									
Microsoft Internet Explorer																									

Şekil 5-40 : Belge açıklamaları

5.3.6.4 Sayfa

Sayfa'ların sorgulanabilmesi ancak Sayfa'ya ait OID bilgisinin verilmesi ile mümkündür. Sayfa'lar, mutlaka Belgelere bağlı olmadıklarından, yani kendi OID'lerine sahip olduklarından, Sayfa sorguları ait oldukları Belge'ye ilişkin bir bilgi de içermezler. Buna rağmen, programın uygulama şekli ile Sayfa'lara sadece ilgili Belge üzerinden erişilebilir.

Aşağıda "30000001" OID değerine sahip bir Sayfa'yı isteyen HTML bağlantısının URL kısmı görülmektedir.

</mmnews.dll/Pages?PageOID=30000001>

Bir Sayfa için istemciye şu bilgiler gönderilir.

- Sayfanın ait olduğu Belge ismi.

- Sayfa'nın Belge içindeki numarası.
- Sayfa OID'i.
- Sayfa adı.
- HTML haline çevrilmiş Sayfa metni. (bkz. HTML Üretici). Sayfa metninin başında ve sonunda birer HTML "horizontal ruler" elemanı bulunur.
- İlgili belgenin diğer Sayfa'larının sorgulanabileceği HTML bağlantıları.

erler Veritabanı - Haberler Veritabanı - Haberler Veritabanı - Haberler Veritabanı - H

Ana Sayfa

Belge Listesi

Sorgulama

Kelime Arama

Kuzey Irak'ta göçe set hazırlığı

OLASI bir göç nedeniyle Kuzey Irak sınırında sıcak günler yaşanıyor. Sınırdaki hareketlilik dün gece yansından sonra iyice arttı. Zirhli birlikler desteğindeki yedi bin Mehmetçik'in Kuzey Irak'a geçtiği iddia edilirken, resmi yetkililer bunu henüz doğrulamadı. Türkiye - Suriye - Irak sınır üçgenindeki Beyaz Dağ'da da, olası geçişleri engellemek amacıyla aynı bir birlik konuşlandırılacağı öne sürüldü. 2. Ordu Komutanı Orgeneral Rasim Betir'in Silopi'deki Sınır Jandarma Tugayı'na gelişini 'operasyon başladı' söylentilerine yol açtı. Bu arada Suriye de olası göçe karşı Irak sınır şeridinde birlik kaydırdı ve Fays - Habur kapısındaki kontrolleri sıklaştırdı.

Habur kapandı ve açıldı

Kuzey Irak'tan olası göçü Kuzey Irak topraklarında önleme planı nedeniyle Habur'da hareketli günler yaşanıyor. Hareketlilik dün gece yansından sonra iyice arttı. Saat 02.00 sularında Habur Sınır Kapısı kapatılırken, Silopi yakınlarındaki Hac Konaklama Tesisleri'nde konuşlanan tank birlikleri Habur Kapısı, Görümlü ve Ortaköy'de Hezil çayı üzerine kurulan iki köprü üzerinde hareketliliğiyle dikkat çekti. Sabah saatlerinde de Habur Sınır Kapısı yeniden araç trafiğine açıldı.

Betir Silopi'de

Körfezde sıcak gelişmeler yaşanırken, olası bir göç dalgasına karşı önlemler paketi uygulamasına geçiliyor. Şırnak'ın Silopi ilçesi Sınır Jandarma Tugayı'nda kriz merkezi oluşturuldu. Tüm gelişmeler

Şekil 5-41 : Belge sayfası

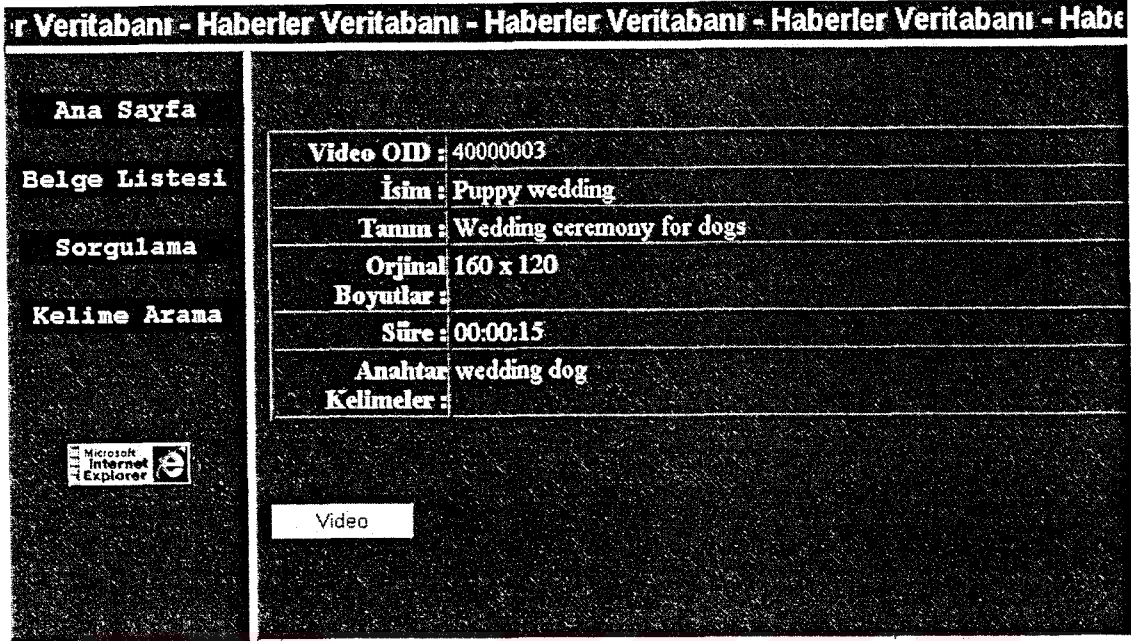
5.3.6.5 Video açıklamaları

Veri Yayıncısı, Video sorgularına, istenen Video'nun açıklama alanlarının bulunduğu bir HTML sayfa üreterek cevap verir. Video sorguları, aşağıda görüldüğü gibi, Yol Bilgisi olarak "Videos" içerirler ve "VideoOID" parametresinde de istenen Video nesnesinin OID bilgisini verirler.

/mmnews.dll/Videos?VideoOID=40000001

Video nesnesinin açıklamaları olarak aşağıdaki bilgiler gösterilmektedir.

- Video nesnesinin OID değeri.
- Video adı.
- Video tanımı.
- Video anahtar kelimeleri.
- Video boyutları (genişlik x boy).
- Video'nun süresi.
- Video karakteristik vektörleri listesi. Bu listedeki her satırda çerçeve numarası ve başlangıç ve bitiş zamanları ile karakteristik vektörün değeri verilir.
- İlgili İçerik nesnesini sorgulayacak bir HTML bağlantısı. Eğer böyle bir referans girilmemiş ise, bağlantı yerine kullanıcıya referans eksikliğini söyleyen bir not görülür.



Şekil 5-42 : Video açıklamaları

5.3.6.6 Ses açıklamaları

Ses açıklama alanlarının bulunduğu HTML sayfalar, Ses sorguları sonucunda üretilirler. Ses sorguları da, Yol Bilgisi olarak "Audios" içerirler ve "AudioOID" parametresinde de istenen Ses nesnesinin OID bilgisini verirler. Aşağıda buna bir örnek verilmiştir.

/mmnews.dll/Audios? AudioOID=50000001

Ses açıklamaları ile kastedilen bilgi alanları şunlardır

- Ses nesnesinin OID değeri.
- Ses nesnesi adı.
- Ses nesnesinin tanımı.
- Ses nesnesinin anahtar kelimeleri.

- Ses nesnesinin süresi.
- Ses nesnesine ait karakteristik vektörlerin bir listesi. Bu listedeki her vektör için, vektörün alındığı zaman aralığının başlangıç ve bitiş noktaları ile karakteristik vektörün değeri verilir.
- Ses nesnesi ile ilgili (yani ses nesnesinin ortam bilgisini saklayan) İçerik nesnesini sorgulayacak bir HTML bağlantısı. Eğer böyle bir referans girilmemiş ise, bağlantı yerinde kullanıcıya referans eksikliğini söyleyen bir not gönderilir.

anı - Haberler Veritabanı - Haberler Veritabanı - Haberler Veritabanı - Haberler Veritabanı

Ana Sayfa	Video OID : 50000003
Belge Listesi	İsim : Ken's singing
Sorgulama	Tanım : Ken Müller is singing Jazz
Kelime Arama	Süre : 00:01:05
	Anahtar Kelimeler : Ken Müller jazz

Microsoft Internet Explorer

Audio

Şekil 5-43 : Ses açıklamaları

5.3.6.7 Resim açıklamaları

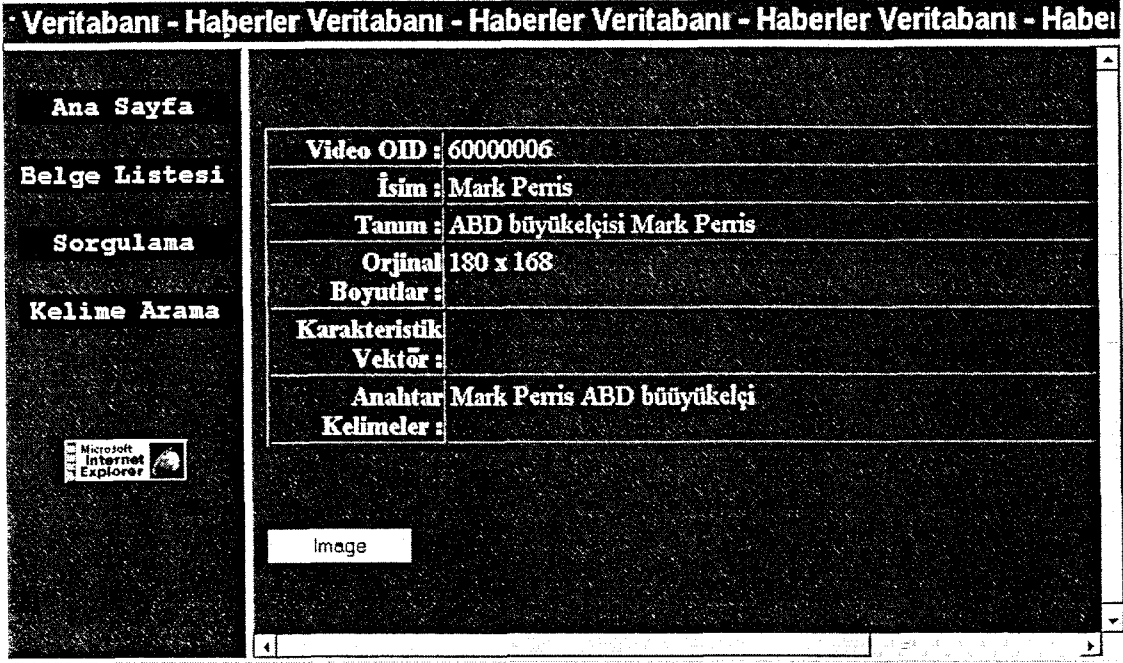
Resim sorguları, Veri Yayıncısı'nın Resim açıklamalarının bulunduğu sayfalar üretmesine neden olurlar. Tipik bir Resim sorgusu şu biçimde yapılabilir,

/mmnews.dll/ Images?ImageOID=60000001

Burada, “Images” Yol Bilgisi ve “ImageOID” istenen Resim nesnesinin OID deęerini ieren parametredir.

Resim aıklama sayfalarında Őu bilgiler bulunur.

- Resim nesnesinin OID deęeri.
- Resim nesnesinin ismi.
- Resim nesnesinin tanımı.
- Resim nesnesinin anahtar kelimeleri.
- Resim nesnesinin karakteristik vektörü. Resim nesnelerinin temsil ettikleri ortamlar, zamanla deęiŐen yapı gstermediklerinden her Resim nesnesi iin bir tane karakteristik vektör dŐünmüŐtür.
- Resim nesnesinin temsil ettięi ortamı ieren bir İerik nesnesine sorgu isteęi ieren HTML baęlantısı. Eęer byle bir referans girilmemiŐ ise, baęlantı yerinde kullanıcıya referans eksiklięini syleyen bir not gnderilir.



Şekil 5-44 : Resim açıklamaları

5.3.6.8 İçerik açıklamaları

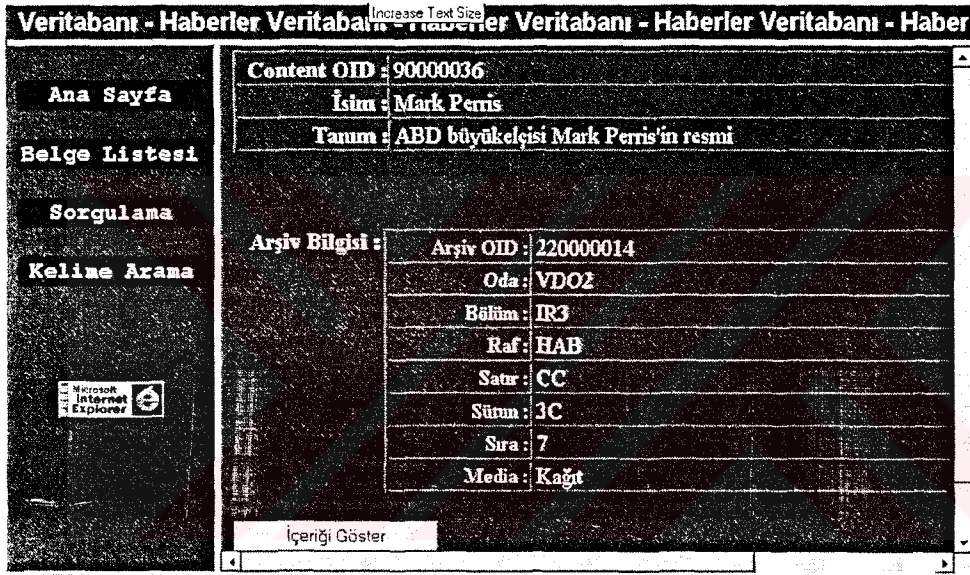
İçerik açıklamalarını içeren sayfalar, Veri Yayıncısı tarafından ve İçerik sorgularına cevaben üretilirler. Sorgu Yöneticisinin bilgiyi doğru bir şekilde alabilmesi için, bir İçerik sorgusunda, nesne tipi (Content) ve nesne belirteci (ContentOID) verilmelidir. Buna göre bir içerik sorgusu aşağıdaki biçimde yapılır,

/mmnews.dll/Content?ContentOID=90000001

İçerik sorgularında üretilen sayfalarda şu bilgiler yer alır.

- İçerik nesnesinin OID değeri.
- İçerik nesnesinin adı.
- İçerik nesnesinin tanımı.

- İçerik nesnesinin sahip olduğu ortamın arşiv saklama bilgisi. Arşiv bilgisi arşiv adresine ait oda, bölüm, raf, satır, sütun ve sıra bilgileri ile orjinal ortam kaynağının saklama ortamı bilgilerini içerir.
- İçerik nesnesinin sakladığı ortam bilgisine erişim için HTML bağlantısı. Eğer ortam veritabanındaki BLOB alanında tutuluyorsa, bağlantı Veri Yayıncısı'na içeriğin çalınması için bir sorgu isteği, eğer ortam işletim sisteminde bir dosya olarak tutuluyorsa bu bağlantı ortam dosyasını gösteren basit bit HTML bağlantısıdır.



Şekil 5-45 : İçerik açıklamaları

5.3.6.9 Ortamın istemciye gönderilmesi

İçerik nesnesindeki BLOB alanında saklanan ortam bilgisi bir ortam sorgusu ile istemciye gönderilir. Bu sorgu aşağıdaki biçimdedir.

/mmnews.dll/Play?ContentOID=90000001

İstek yapıldığında, Veri Yayıncısı ilgili İçerik nesnesini bulur, başlığındaki "Content Type" alanı İçerik nesnesinin "MIME Tipi" alanına eşit olan ve mesaj

gövdesi olarak BLOB alanındaki ikili veriyi içeren bir HTTP cevap mesajı üretir ve cevabı istemciye gönderir.

Eğer İçerik nesnesi ortam bilgisini dosya sisteminde tutuyorsa, sadece o dosyaya bağlantıyı içeren bir HTML sayfa; eğer BLOB alanında hiç veri bulunmuyorsa hata mesajını içeren bir HTML sayfa gönderilir.

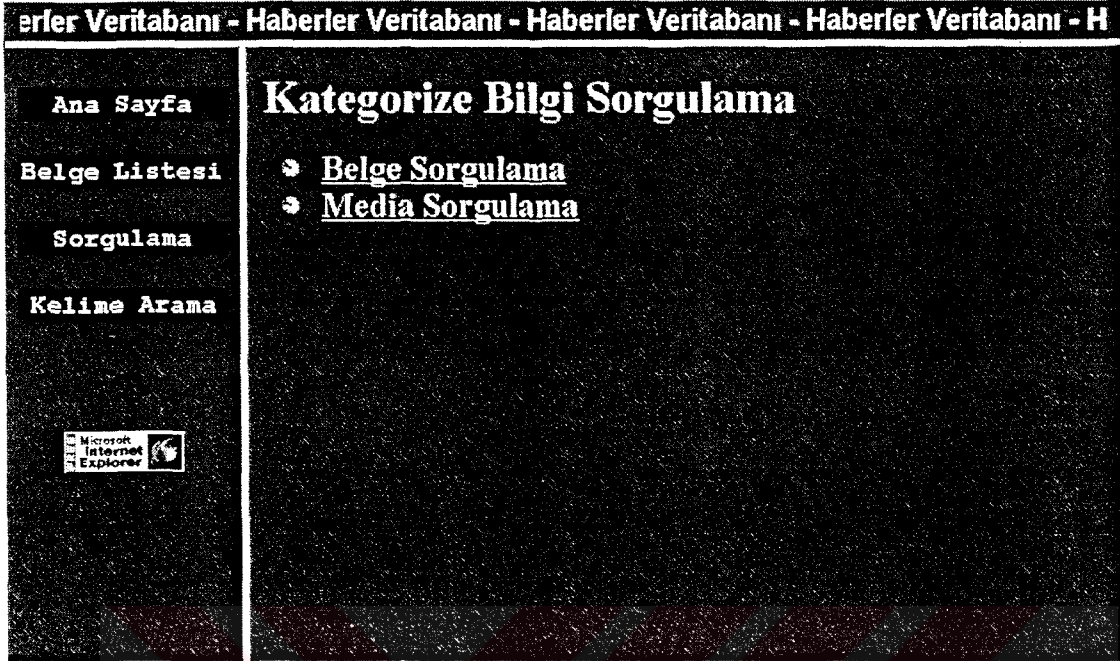


Şekil 5-46 : İstemcide video oynatılması

5.3.6.10 Özelleştirilmiş açıklama sorguları

Sistemde tutulan açıklama bilgilerinin sorgulanması için iki seçenek sunulmuştur,

- Belge açıklamalarının sorgulanması.
- Ortam (Video, Ses, Resim, İçerik) açıklamalarının sorgulanması.



Şekil 5-47 : Özelleştirilmiş açıklama sorgusu

5.3.6.11 Özelleştirilmiş belge sorgusu

Özelleştirilmiş belge sorgusu, kullanıcının Belge açıklamaları üzerinde istediği gibi sorgulama yapmasına ve sonuçları, Klasör sorgusundaki gibi, bir Belge listesi halinde almasına imkan tanır. İlk istek yapıldığında boş açıklamalar formu içeren bir HTML sayfa hazırlanıp istemciye gönderilir. Bundan sonra kullanıcı üzerinde sorgulama yapmak istediği alanları doldurarak formu sunucuya geri gönderir. Gönderme eylemi tipi "POST" ve eylem ise formu isterken kullanılan URL'nin aynısıdır.

Özelleştirilmiş belge sorgusu isteği aşağıdaki URL ile yapılır.

/mmnews.dll/QueryDoc

Veri Yayıncısı içindeki ilgili eylem işleyici gelen HTTP isteğindeki form alanlarına bakarak nasıl davranacağına karar verir. Eğer istek mesajı istenen form alanlarını içeriyorsa, bu alanlar üzerinden sorgulama işlemi yapar. Eğer

istenen form alanları istek mesajında bulunamıyorsa, bu büyük ihtimalle sorgulama formu isteğidir diye düşünülüp boş sorgulama sayfası hazırlanır ve istemciye gönderilir.

Sorgulama başarı ile tamamlandığında, kullanıcının belirlediği koşullara uyan Belge'lerin bir listesi HTML sayfa halinde istemciye gönderilir. Aslında bu listedeki her satır, Belge listesi isteklerinde olduğu gibi, ilgili Belge'yi isteyen belge sorgusuna bir bağlantıdır.

Belge nesnesinin herhangi bir açıklama alanı için koşul girilebilir. Bu alanların listesi "HTML Üretici – Belge açıklamaları" ve "Sorgu Yöneticisi" bölümlerinde verilmiştir.

Eğer bir açıklama alanına sorgu formunda değer girilmiş ise, Sorgu Yöneticisi, bu alanı ve girilen değeri SQL (Structured Query Language) cümlesinin "WHERE" koşuluna ekler. Eğer açıklama alanı veritabanında karakter tabanlı olarak tanımlanmış ise, koşul sorgu formunda girilen karakterlerle başlayan değerleri arayacak biçimde düzenlenir.

SQL cümlesinin oluşturulması şöyle olur,

1. SQL cümlesi, ilk olarak, tablo ismi olarak "DOCUMENTS" ve alan ismi olarak "DOCUMENTS.*" içeren bir "SELECT" cümlesidir.
2. Form alanları, bilgi girilmiş olanları bulmak için teker teker incelenir. Dolu bir form alanı bulunduğu anda o alana ait şart cümlesi, SQL cümlesinin "WHERE" ifadesine eklenir.
3. Eğer form alanı başka bir tablodaki değeri temsil ediyorsa, o tablonun adı da "FROM" ifadesinin sonuna eklenir ve "DOCUMENTS" tablosu ile bu tablo arasındaki birleştirme (join) ifadesi de oluşturularak "WHERE" ifadesinin sonuna eklenir.

Şekil 5-48 : Özelleştirilmiş belge sorgusu

Form alanlarının isimleri için “Sorgu Yöneticisi” bölümüne bakınız..

5.3.6.12 Özelleştirilmiş ortam sorgusu

Sistemde kayıtlı çokluortam nesnelere de açıklamaları bazında sorgulanabilirler. Özelleştirilmiş ortam sorgusu bu işe yarar. İlk istek yapıldığında boş bir form içeren HTML sayfa üretilip istemciye gönderilir. Bundan sonra kullanıcı üzerinde sorgulama yapmak istediği alanları doldurarak formu sunucuya geri gönderir. Gönderme eylemi tipi “POST” ve eylem ise formu isterken kullanılan URL’ nin aynısıdır.

Özelleştirilmiş ortam sorgusu için gerekli URL şudur.

/mmnews.dll/QueryMedia

Veri Yayıncısı içindeki ilgili eylem işleyici gelen HTTP isteğindeki form alanlarına bakarak nasıl davranacağına karar verir. Eğer istek mesajı istenen form alanlarını içeriyorsa, bu alanlar üzerinden sorgulama işlemini yapar. Eğer

istenen form alanları istek mesajında bulunamıyorsa, bu büyük ihtimalle sorgulama formu isteğidir diye düşünülüp boş sorgulama sayfası hazırlanır ve istemciye gönderilir.

Sorgulama başarı ile tamamlandığında, kullanıcının belirlediği koşullara uyan çokluortam nesnelerin bir listesi HTML sayfa halinde istemciye gönderilir. Aslında, bu listedeki her satır ilgili çokluortam nesneyi isteyen nesne sorgusuna bir bağlantıdır. Örneğin istenen koşulları sağlayan bir Resim nesnesi için olan satır, şuna benzer bir bağlantı içerir.

/mmnews/Images/ImageOID=60000001

Çokluortam nesnelerinin sorgulanması için açıklama alanlarına koşullar verilir. Sorgulamada kullanılan alanlar ve açıklamaları aşağıda verilmiştir.

- Nesne tipi hangi tip çokluortam nesnelerinin sorgulanacağına karar verir. Geçerli seçenekler “Hepsi”, “Resim”, “Video”, “Ses” ve “İçerik”tir.
- Nesnenin ismi.
- Nesnenin tanımı.
- Nesnenin boyutları (sadece Video ve Resim nesneleri için geçerlidir).
- Nesnenin süresi (sadece Video ve Ses nesneleri için geçerlidir).
- Karakteristik vektör değeri (sadece Ses, Video ve Resim nesneleri için geçerlidir).
- Anahtar kelimeler (sadece Ses, Video ve Resim nesneleri için geçerlidir).
- İçerik nesnesinin sahip olduğu ortamın arşiv saklama bilgisi. Arşiv bilgisi arşiv adresine ait Oda, Bölüm, Raf, Satır, Sütun ve Sıra bilgileri ile orjinal ortam kaynağının saklama ortamı bilgilerini içerir.

Özelleştirilmiş ortam sorgusu isteğini yerine getirmek için, her çokluortam nesne tipi için bir tane olmak üzere, dört taneye kadar SQL cümlesi oluşturulabilir. Sorgulama da kullanılan her bir SQL cümlesinin oluşturulması, “Özelleştirilmiş belge sorgusu”nda anlatıldığı gibidir. Bir nesne için SQL

cümlesinin oluşturulması sorgulanacak nesne tipini belirten form alanının değerinin o nesnenin adını veya "Hepsi" olması gerekir.

Veritabanı - Haberler Veritabanı - Haberler Veritabanı - Haberler Veritabanı - Haberler

Ana Sayfa

Belge Listesi

Sorgulama

Kelime Arama

Microsoft Internet Explorer

Ortam Sorgulama

Ortam Tipi :

İsim :

Tanım :

Orjinal Boyutlar : x * Video ve Resim için

Süre : * Video ve Audio için

Karakteristik Vektör : * Video, Audio ve Resim için

Anahtar Kelimeler :

Şekil 5-49 : Özelleştirilmiş ortam sorgusu

Bu sorgulama formunda kullanılan form alanlarının isimleri için "Sorgu Yöneticisi" bölümüne bakınız.

5.3.6.13 Kelime arama

Kelime arama sorgusu ile istenilen kelimeler sistemdeki tüm Belge Sayfa'larının biçimsiz Metin'leri içinde aranır ve kelimelerin en az birinin bulunduğu Sayfa'ların ait oldukları Belge'lerin bir listesi "Özelleştirilmiş belge sorgusu" veya "Klasör sorgusu"nda olduğu gibi getirilir. Kelime arama istekleri, aşağıda görüldüğü gibi, "QueryWord" Yol Bilgisi ile yapılırlar.

/mmnews.dll/QueryWord

Eğer HTTP istek mesajı form bilgisini içermiyorsa, kullanıcının aranacak kelimeleri girip sorgulamayı başlatacağı ekran hazırlanıp istemciye gönderilir.



Şekil 5-50 : Kelime arama

5.4 Veritabanının Uygulanması

Sistemin gereklenme surcünde veritabanının uygulanması nemli bir yer tutar. Burada verilecek karar, daha sonra sistemin uygulanma ve bakımındaki kolaylık ile, zellikle asistemin ađır ykl olduđu, bir ok kullanıcıya aynı anda hizmet vermek zorunda kaldıđı durumlarda, performans ve dayanıklılıđını nemli lde etkiler.

Nesne tabanlı bir sistemde veritabanı uygulaması iin temel olarak drt alternatif gze arpmaktadır.

5.4.1 Nesne tabanlı veritabanı zerinde nesne tabanlı tasarımın uygulanması

Bu ilk akla gelen ve en kolay uygulama eklidir. Nesne tabanlı bir veritabanı miras, ok ekillilik (polymorphism) ve domen bilgisinin nesnelere iinde tutulması gibi nesne tabanlı bir tasarımın anahtar zelliklerini ierir. Sonu olarak, hi bir eŐleŐtirme veya evrime gerek kalmadan uygulama tasarlandıđı gibi yapılır. Nesne tabanlı veritabanları, ayrıca, kendiliđinden nesne tabanlı oklu ortam uygulamaları iin de daha uygundur.

te yandan nesne tabanlı veritabanlarının da bazı sorunları vardır. İlk olarak, nesne tabanlı veritabanları, iliŐkisel veritabanları kadar gl bir matematiksel temele dayanmazlar. Bundan baŐka, ticari olarak uygulanan nesne tabanlı veritabanları bulunmasına rađmen, yaygın olarak kullanılmamaktadırlar ve bylece, ok yođun veri iŐlemlerindeki performans ve dayanıklılıkları iliŐkisel veritabanları kadar iyi sınınamamıŐtır. Son olarak, bu veritabanlarının birođu programlama dili olarak sadece C++'ı desteklemekte ve nesne deđiŐim standartları olan Microsoft[®]'un OLEsi (yine Microsoft[®]'un COM -Common Object Model- standardına dayalı) veya IBM[®]'in OpenDoc'u (Object Management Group'un Common Object Request Broker Architecture'ına dayalı) gibi yapıları desteklememektedirler.

5.4.2 Dayanıklı nesne dosya sistemi üzerinde nesne tabanlı tasarımın uygulanması

Dayanıklı nesne dosya sistemi nesnelerin dayanıklı bir depolama ortamında saklanıp erişimlerini sağlar. PSOM (SOM -System Object Model- Persistent framework) (IBM 1993 and Lau 1994), OMG (Object Management Group) Persistence Service (Sessions 1996) ve Taligent Frameworks' Persistence Infrastructure dayanıklı nesne dosya sistemi uygulamalarına örnektirler. Dayanıklı nesne dosya sistemi, sorgulama ve indeksleme imkanı tanımadıklarından, bu yaklaşım, sorgulamaların gerçekleştirilebilmesi için çok fazla programlama yapılmasını gerektirir. Ayrıca, basit bir yapıda olmaları nedeni ile ilişkisel veritabanları kadar dayanıklı ve performanslı olmayabilirler. Bu sebeplerle, dayanıklı nesne dosya sistemi modeli de çokluortam uygulamaları için uygun görülmemektedir..

5.4.3 İVTYS üzerinde nesne tabanlı tasarımın uygulanması — Nesnelerin İVTYS'te depolanması

Nesnelerin İVTYS'de (İlişkisel Veritabanı Yönetim Sistemi) depolanması yaklaşımı (Lehman et al. 1996)'da uygulanmıştır. Bu uygulamada C++ tip sisteminde yaratılan uygulamalar veritabanında depolanır. Sonuç olarak nesnelerin veritabanına depolanması ve geri alınmalarında tip çevrimine gerek kalmaz. İlişkisel tablonun alanları nesnelerin depolandığı kaplar görevini yaparlar. C++ nesnelerinin kaplara depolanmasında iki tür eşleştirme metodu kullanılmıştır. ADT (Abstract Data Type) eşleştirmesi ve BLOB (Binary Large Object) eşleştirmesi. ADT eşleştirmesinde bir sınıf veya sınıf hiyerarşisine ait bir tek nesne depolanırken, BLOB eşleştirmesi değişik sınıf tanımlarına ait birçok nesneyi bir kaptan toplar.

Ancak, bu yaklaşım veritabanını tamamen uygulamaya özel kılmakta ve bilgiye sadece programın izin verdiği ölçüde ulaşımı sağlamaktadır. Ayrıca, Nesnelerin İVTYS'de depolanması yöntemi ticari bir veritabanında da uygulanmamıştır.

5.4.4 İVTYS üzerinde nesne tabanlı tasarımın uygulanması —

Nesne tabanlı sistemin ilişkisel sisteme çevrilmesi

Nesne tabanlı sistemin ilişkisel karşılığına çevrimi yaklaşımı ile, sistemdeki sınıflar domen bilgileri çıkartılarak, tablolara dönüştürülürler.

Bu yöntemin faydası, ilişkisel sistemlerin matematik temeli ve yaygın kullanımlarının getirdiği olanakları son dereceye kadar kullanmalarınıdır. Sonuç olarak geliştirilen sistem, daha dayanıklı , geliştirilmesi daha kolay ve kurulumu ve bakımı daha ucuz bir yapıya kavuşmuş olur.

Bu yaklaşım, Çokluortam Haberler Veritabanının uygulanmasında da yukarıdaki sebeplerle tercih edilmiştir.

5.4.5 İVTYS uygulaması

Kullanılacak İVTYS aracının seçimi sistemin etkin işleyişini büyük ölçüde etkiler.

Geliştirme kolaylığı için seçilen İVTYS aracının, diğer geliştirme araçları ile birlikte aynı bilgisayarda çalışabilecek olması da tercih edilmiştir.

Geliştirme sürecinde, Borland® InterBase® 4.2 İVTYS sunucu, İVTYS geliştirme aracı olarak kullanılmıştır.

İVTYS uygulaması, veritabanı nesnelere yaratılmasını öngörür. Çokluortam Haberler Veritabanı'ndaki nesnelere şu tiplerdedirler.

- Tablolar (Tables)
- İndeksler (Indexes)
- Tetikleyiciler (Triggers)
- Üreticiler (Generators)
- Domenler (Domains)

Bu nesnelerin oluşturulması için gerekli SQL/DDL (Data Definition Language) kodu “Ekler” bölümünde verilmiştir.

5.4.5.1 Tablolar

Tablolar çevrimden sonra elde edilen ilişkisel şemaya göre yaratılmışlardır. Veritabanındaki tablo isimleri ve bunlara karşılık gelen ilişkisel şema varlık isimleri aşağıdaki tabloda verilmiştir.

Tablo 5-3 : İlişkisel şema isimleri ile veritabanı nesne isimlerinin eşleştirilmesi

İlişkisel Şema Varlığı İsmi	Veritabanı Nesnesi İsmi
Folder	KLASOR
Document	DOCUMENTS
Page	DOC_PAGE
Text	TEXT
Text Attributes	TEXT_ATTRIBUTES
Storage	DOC_STORAGE
Image	IMAGE
Video	VIDEO
Video Feature Vector	VIDEO_FEATURE_VECTORS
Audio	AUDIO
Audio Feature Vector	AUDIO_FEATURE_VECTORS
Content	CONTENT
Geographical Location	LOCATION
Role	ROLE

İlişkisel Şema Varlığı İsmi	Veritabanı Nesnesi İsmi
Person	PERSON
Related Persons	RELATED_PERSONS
Related News	RELATED_DOCUMENTS
Style	HTML_ANNOTATIONS
Web Link	WEB_LINKS
News Type	NEWS_TYPE
News Subject	NEWS_SUBJECT
Importance	IMPORTANCE
MIME Type	MIME_TYPE
Media Type	MEDIA_TYPE

5.4.5.2 İndeksler

İndeksler, veritabanındaki veri yoğun işlemler ve içerilen verinin büyüklüğü ile ortaya çıkan, sorgulamalardaki performans problemlerini çözmek için kullanılmışlardır.

Nesne belirteçleri (OID), ilgili tablolarının birincil anahtarları olarak kullanılmışlardır.

Aşağıda, veritabanı sistemindeki index nesnelerinin ve karşılık gelen tabloların bulunduğu bir tablo verilmiştir.

Tablo 5-4 : İndeksler

İndeks Alan(lar)ı	Tablo İsmi	İndeks Türü
OID	KLASOR	Birincil Anahtar
PARENT_OID, ISIM	KLASOR	İkincil Anahtar
OID	DOCUMENTS	Birincil Anahtar
OID	DOC_PAGE	Birincil Anahtar
OWNER, PAGENO	DOC_PAGE	İkincil Anahtar
OID	TEXT	Birincil Anahtar
TEXT, ANNOTATION, OBJECT, START, LEN, INLINEPATH	TEXT_ATTRIBUTES	Birincil Anahtar
OID	DOC_STORAGE	Birincil Anahtar
OID	IMAGE	Birincil Anahtar
OID	VIDEO	Birincil Anahtar
OID, FRAME, START_TIME, END_TIME	VIDEO_FEATURE_VECTORS	İkincil Anahtar
OID	AUDIO	Birincil Anahtar

İndeks Alan(lar)ı	Tablo İsmi	İndeks Türü
OID, FRAME, START_TIME, END_TIME	AUDIO_FEATURE_VECTORS	İkincil Anahtar
OID	CONTENT	Birincil Anahtar
OID	LOCATION	Birincil Anahtar
OID	ROLE	Birincil Anahtar
OID	PERSON	Birincil Anahtar
DOC, PERSON, ROLE	RELATED_PERSONS	Birincil Anahtar
DOC1, DOC2	RELATED_DOCUMENTS	Birincil Anahtar
OID	HTML_ANNOTATIONS	Birincil Anahtar
ISIM	HTML_ANNOTATIONS	İkincil Anahtar
OID	WEB_LINKS	Birincil Anahtar
OID	NEWS_TYPE	Birincil Anahtar
OID	NEWS_SUBJECT	Birincil Anahtar
OID	IMPORTANCE	Birincil Anahtar
OID	MIME_TYPE	Birincil Anahtar
OID	MEDIA_TYPE	Birincil Anahtar

5.4.5.3 Tetikleyiciler

Tetikleyiciler, kayıt ekleme, silme veya günleme gibi bir veritabanı işlemi gerçekleştiğinde “tetiklenen”, yani veritabanı sistemi tarafından otomatik olarak çalıştırılan kod parçacıklarıdır.

Veritabanı tetikleyicileri ISQL (InterBase SQL) dilinde geliştirilmişlerdir.

Tetikleyiciler, birincil nesneye karşılık gelen (yani OID’i birincil anahtar olan) tüm tablolarda, OID değerlerini otomatik olarak üretmek için kullanılmışlardır. Tabloya yeni bir kayıt eklenmeden önce, “Ekleme Öncesi” (Before Insert) tetikleyicisi çalışıp, eklenen yeni ilgili nesne için OID numarasını üretir. OID üreten tetikleyiciler, bir sonraki numarayı bulmak için veritabanının “Üretici” (Generator) nesnelerini kullanırlar.

“Ekleme Öncesi” (Before Insert) tetikleyicisi, ayrıca, Sayfa nesnelerini tutan “DOC_PAGE” tablosunda, eklenen Sayfa için bir sonraki sayfa numarasının üretilmesinde kullanılır.

Bir kayıt silindiğinde tetiklenen “Silme Sonrası” (After Delete) tetikleyicileri, ilgili kayıtların diğer tablolardan da silinmesini veya ilgili referansların kaldırılmasını sağlarlar. Böylece, veritabanının bütünlüğünü bozan ve performansı olumsuz yönde etkileyen öksüz kayıtların oluşması önlenir.

“KLASOR” tablosuna yeni bir kayıt eklenmeden önce çalışan “TR_KLASOR_BI” tetikleyicisi için kod örnek olarak aşağıda verilmiştir.

```
CREATE TRIGGER TR_KLASOR_BI FOR KLASOR BEFORE  
INSERT POSITION 0 AS
```

```
BEGIN
```

```
    NEW.OID = GEN_ID(GN_KLASOR_OID,1);
```

```
END
```

Burada, “NEW” belirteci yeni kaydın değerleri anlamındadır.

Herhangi veritabanı olayı için birden fazla tetikleyici tanımlanabilir. Bu durumda, tetikleyicilerin çalıştırılma sırası “POSITION” ifadesinde verilen sıraya göre küçükten büyügedir.

5.4.5.4 Üreticiler

Üreticiler otomatik sıra numarası üretilmesine yarayan veritabanı nesnelere sahiptirler. Bir üretici nesnesi, bir isim ve başlangıç sayısı ile yaratılır. Bundan sonra “GEN_ID” işlevi, üretici ismi ve artım miktarı ile her çağırılışında, üreticinin değeri atomik bir transaction içinde arttırarak, elde edilen değeri geri döndürür.

“GEN1” adında ve “1” ilk değerine sahip bir üreticinin yaratılması için sentaks şöyledir.

```
CREATE GENERATOR GEN1 ;
SET GENERATOR GEN1 TO 1 ;
```

Yaratılan bu üreticinin değerini atomik transaction içinde 1 arttıracak ifade ise aşağıdaki gibidir.

```
NEW.OID = GEN_ID(GEN1,1);
```

Çokluortam Haberler Veritabanı’ndaki üreticilerin isimleri “GN_” öneki ile başlar ve açıklayıcı bilgi ile devam eder.

5.4.5.5 Domenler

Domenler, tüm veritabanında kullanılan veri tiplerinin standardizasyonunu sağlarlar. Bir alanın domen tanımında belirtilen tüm kısıtlar, o alana girilen veriye uygulanır.

Domenlerin tablo alan tanımlarındaki kullanımları, diğer geleneksel veri tiplerinin kullanımları ile aynıdır. Aşağıdaki örnekte, “DESCRIPTION” adlı tablo alanı “DOM_DESCRIPTION” domeninde tanımlanmıştır.

CREATE TABLE DOCUMENTS

```
(
  ....
  X          INTEGER
  DESCRIPTION DOM_DESCRIPTION
  ....
)
```

Sistemdeki her OID alanı kendi domen tanımına sahiptir. OID domen tanımları OID bilgisinin tasarımdaki tanımına uymasını sağlarlar (bkz. “Nesne Belirteçlerinin Yapısı”). Bundan başka, domenlerin kullanılması ile ilişkiyel uyumluluk (referential integrity), veri tiplerinin uyumluluğu olarak kısmen zorlanmış olur.

Aşağıdaki tabloda Çokluortam Haberler Veritabanında kullanılan domen tanımları ve karşılık gelen kısıtları listelenmiştir.

Tablo 5-5 : Domenler

Domen İsmi	Veri Tipi	Boy	Kısıtlar
DOM_OID	NUMERIC	9	
DOM_KLASOR_OID	NUMERIC	9	IS NULL OR >=10000000 OR <=19999999
DOM_DOCUMENT_OID	NUMERIC	9	IS NULL OR >=20000000 OR <=29999999

Domain İsmi	Veri Tipi	Boy	Kısıtlar
DOM_PAGE_OID	NUMERIC	9	IS NULL OR >=30000000 OR <=39999999
DOM_VIDEO_OID	NUMERIC	9	IS NULL OR >=40000000 OR <=49999999
DOM_AUDIO_OID	NUMERIC	9	IS NULL OR >=50000000 OR <=59999999
DOM_IMAGE_OID	NUMERIC	9	IS NULL OR >=60000000 OR <=69999999
DOM_TEXT_OID	NUMERIC	9	IS NULL OR >=80000000 OR <=89999999
DOM_CONTENT_OID	NUMERIC	9	IS NULL OR >=90000000 OR <=99999999
DOM_LOCATION_OID	NUMERIC	9	IS NULL OR >=100000000 OR <=109999999
DOM_ROLE_OID	NUMERIC	9	IS NULL OR >=110000000 OR <=119999999

Domen İsmi	Veri Tipi	Boy	Kısıtlar
DOM_PERSON_OID	NUMERIC	9	IS NULL OR >=120000000 OR <=129999999
DOM_NEWS_TYPE_OID	NUMERIC	9	IS NULL OR >=150000000 OR <=159999999
DOM_NEWS_SUBJECT_OID	NUMERIC	9	IS NULL OR >=160000000 OR <=169999999
DOM_IMPORTANCE_OID	NUMERIC	9	IS NULL OR >=170000000 OR <=179999999
DOM_HTML_ANNOTATIONS_OID	NUMERIC	9	IS NULL OR >=190000000 OR <=199999999
DOM_STORAGE_MEDIA_TYPE_O ID	NUMERIC	9	IS NULL OR >=200000000 OR <=209999999
DOM_MIME_TYPE_OID	NUMERIC	9	IS NULL OR >=210000000 OR <=219999999
DOM_DOC_STORAGE_OID	NUMERIC	9	IS NULL OR >=220000000 OR <=229999999

Domain İsmi	Veri Tipi	Boy	Kısıtlar
DOM_WEB_LINKS_OID	NUMERIC	9	IS NULL OR >=250000000 OR <=259999999
DOM_USERNAME	VARCHAR	20	
DOM_PATH_NAME	VARCHAR	200	
DOM_SHORT_NAME	CHAR	30	
DOM_LONG_NAME	CHAR	100	
DOM_VIDEO_FEATURE_VECTOR	CHAR	100	
DOM_IMAGE_FEATURE_VECTOR	CHAR	100	
DOM_AUDIO_FEATURE_VECTOR	CHAR	100	
DOM_KEYWORDS	VARCHAR	200	
DOM_DESCRIPTION	VARCHAR	200	
DOM_TIME	CHAR	8	

5.4.5.6 İlişkiler

Çokluortam Haberler Veritabanı'ndaki tablolar arası ilişkiler, aşağıdaki tabloda verildiği üzere tanımlanmışlardır.

Tablo 5-6 : Tablolararası ilişkiler

Birinci Tablo		İkinci Tablo		İlişki Tipi
Tablo İsmi	Alan İsmi	Tablo İsmi	Alan İsmi	
AUDIO	OID	AUDIO_FEATURE_ VECTORS	OID	Bire-Çok
CONTENT	OID	AUDIO	CONTENT	Bire-Çok
CONTENT	OID	VIDEO	CONTENT	Bire-Çok
CONTENT	OID	IMAGE	CONTENT	Bire-Çok
DOC_IMPORTANCE	OID	DOCUMENTS	IMPORTANCE	Bire-Çok
DOC_STORAGE	OID	CONTENT	CSTORAGE	Bire-Çok
DOC_STORAGE	OID	DOCUMENTS	DOC_STORAGE	Bire-Çok
DOCUMENTS	OID	DOC_PAGE	OWNER	Bire-Çok
DOCUMENTS	OID	RELATED_ DOCUMENTS	DOC1	Bire-Çok
DOCUMENTS	OID	RELATED_ DOCUMENTS	DOC2	Bire-Çok
DOCUMENTS	OID	RELATED_PERSONS	DOC	Bire-Çok
HTML_ANNOTATIONS	OID	TEXT_ATTRIBUTES	ANNOTATION	Bire-Çok

Birinci Tablo		İkinci Tablo		İlişki Tipi
Tablo İsmi	Alan İsmi	Tablo İsmi	Alan İsmi	
KLASOR	OID	DOCUMENTS	FOLDER_OID	Bire-Çok
LOCATION	OID	DOCUMENTS	LOCATION	Bire-Çok
MEDIA_TYPES	OID	DOC_STORAGE	MEDIA_TYPE	Bire-Çok
MIME_TYPE	OID	CONTENT	OBJECT_TYPE	Bire-Çok
NEWS_SUBJECT	OID	DOCUMENTS	SUBJECT	Bire-Çok
NEWS_TYPE	OID	DOCUMENTS	TYPE	Bire-Çok
PERSON	OID	DOCUMENTS	COPYRIGHT	Bire-Çok
PERSON	OID	DOCUMENTS	SOURCE	Bire-Çok
PERSON	OID	DOCUMENTS	APPEARED_IN	Bire-Çok
PERSON	OID	RELATED_PERSONS	PERSON	Bire-Çok
ROLE	OID	RELATED_PERSONS	ROLE	Bire-Çok
TEXT	OID	DOC_PAGE	TEXT	Bire-Sıfır- veya-Bir
TEXT	OID	TEXT_ATTRIBUTES	TEXT	Bire-Çok
VIDEO	OID	VIDEO_FEATURE_ VECTORS	OID	Bire-Çok

6. SONUÇ ve ÖNERİLER

Bu tez, sonucunda bir Çokluortam Haberler Sistemi üretmiştir. Sistemin tasarımında büyük ölçüde nesne tabanlı tekniklerin kullanımı ile, tüm projenin böl-ve-yönet geliştirme için küçük parçalara ayrılmasını kolaylaştırılmış ve tasarım daha anlaşılır hale getirilmiştir.

Projenin uygulanmasında WWW kullanılması, sonuçta sistemin doğal olarak herkese açık olmasını sağlamıştır. Uygulamada ayrıca HTTP ve HTML üzerinde veri yönetimi için özel teknikler kullanılmıştır. Bunu sağlamak için de, internet ve ilgili protokoller üzerinede fazlaca çalışılmıştır.

Çokluortam verisi, doğası itibarı ile yoğun olduğundan, tasarımda ve programların geliştirilmesinde, gereksiz verinin veritabanından alınmasını önleyecek teknikler uygulanmıştır.

Veri Girişi modülü, istemci/sunucu mimaride ve LAN (Local Area Network) üzerinde çalışacak şekilde tasarlanmıştır. Böylece güvenlik sorunları sadece küçük ağlara sıkıştırılmış ve çözümleri daha kolay hale gelmiştir. Ayrıca, istemci/sunucu mimaride, özellikle veritabanına bilgi girişi yapılırken, performans etkin görünmektedir.

Çokluortam Haberler Sistemi için, hala geliştirilmesi gereken birçok özellikten söz edilebilir. Bundan sonraki bölümlerde sistemin eksik görülen yanları ve yapılabilecek ekler için öneriler yer almaktadır.

6.1 *Internet Arama*

Internet arama araçları, Internet üzerinde herhangi bir bilginin aranmasında yaygın olarak kullanılırlar. Gerçekte, Internet o kadar hızlı büyümektedir ki, bu araçlar Internet'te aradığını bulmanın yegane yolu haline gelmişlerdir. Bu araçlar Internet robotları vasıtası ile aranılan bilgiyi tüm Internet'ten toplarlar. Sonuç olarak, eğer Çokluortam Habeler Sistemi tüm Internet'te yayın yapacaksa, bu arama araçları yolu ile erişilebilir olması kaçınılmazdır.

Veri Yayıncısı, İnternet robotlarının çalışmasını engelleyecek şekilde sayfalar ürettiğinden, arama araçları Çokluortam Haberler sitesinde arama yapamazlar.

Arama araçlarının, Çokluortam Haberler sitesini kullanabilmeleri için aşağıdaki yöntemler önerilmiştir.

- “NO-ROBOTS” açıklaması olmayan bir HTML sayfa üretilebilir. Bu sayfada, sistemdeki tüm Belge ve çokluortam nesnelerinin bazı alanları (örneğin anahtar kelimeler veya açıklama alanları) bulunur. Daha sonra siteye uğrayan robotlar bu sayfaya erişip içindeki kelimeleri aramalarda kullanabilirler.
- Veri Yayıncısı, siteyi ziyaret eden robotu algılayacak şekilde değiştirilebilir. Bu durumda siteyi ziyaret eden robot için yukarıdakine benzer, özel bir HTML sayfa üretilebilir. Hatta, bu sayfa eğer uygulanmış ise, FTS (Full Text Search) kelime indekslerini de içerebilir.

6.2 Metin Arama

Sistem Belge Sayfa'larında kelime aramaya izin vermesine rağmen, bu işlem için özel bir teknik kullanılmamıştır. Bu yüzden ki, şu anda uygulanan kelime arama özelliği, sistem yükü ve veri yoğunluğu arttıkça daha düşük performans gösterir.

Metin aramada kullanılacak bir alternatif FTS (Full Text Search)'tir. FTS uygulamasında, kelimeler kök ve eklerine ayrılarak indekslenirler. FTS tüm kelimeleri indekslere katmaz. Dilin özelliğine göre çok kullanılan kelimeler, bağlaçlar, yardımcı kelimeler indekslenmezler. Bu özellikleri ile FTS, çok yer ve hız açısından etkin arama olanakları sağlar.

Diğer bir alternatif ise, WWW sitelerindeki kelimelerin bu tip indekslenmesi için, Digital®'in AltaVista Seek®'i veya Microsoft®'un Index Server®'ı gibi ticari programların kullanılmasıdır.

6.3 İçerik Arama (Content Based Retrieval – CBR)

Çokluortam Haberler Veritabanı uygulamasında, karakteristik vektörler veritabanında depolanabilmektedirler. Ancak, bu bilgi, karakteristik vektör alanları üzerinde yapılabilen basit metin arama dışında, sistem tarafından değerlendirilmemektedir.

Bir çokluortam sistemin istenen özelliklerinden biri de, depolanan ortam bilgisinin ortamın içeriğine göre sorgulanabilmesidir. Bu özelliği kazandırmak için, Veri Yayıncısı ve belki de Veri Girişi modüllerine ekler yapmak gerekmektedir.

Bu ekler ilgili karakteristik vektörlerin sisteme girişi, içerik sorgulanmasının kullanıcı tarafından hazırlanabilmesi ve verilen sorgu şartları ile ilgili karakteristik vektörlerin eşleşmesini sağlamalıdır.

6.4 WWW bilgi girişi

Çokluortam Haberler Sistemi'nin Veri Giriş modülü, istemci/sunucu mimaride ve LAN üzerinde çalışmaktadır. İşletim sistemi olarak Windows 95, Windows NT Workstation 3.51 veya üst sürümleri kullanılabilir. Veritabanı sunucusu da Borland® InterBase® İVTYS aracını kullanır ve istemci ile aynı işletim sistemlerinde çalışabilir.

Eğer uygulama bilgi girişinin de internet üzerinden yapılmasını gerektirirse, yeni bir Veri Giriş modülü tasarlanmalıdır. Bu yeni uygulama, Veri Yayıncısı'na benzer bir mantıkla WWW sunucu üzerinde çalışacaktır. Yani, HTTP protokolünde gelen isteklerle veritabanını düzenleyecektir. Bu yeni tasarımda ayrıca, sayfa metinleri ve ortam verisi sunucuya gönderilmelidir. Bunun için HTTP 1.1 protokolünde gelen "PUT" eylem tipi kullanılmalıdır. Yani, yeni geliştirilecek program HTTP 1.1 uyumluluğu gerektirecektir.

Veri Girişi uygulaması Internet üzerine aktarıldığında, güvenlik ile ilgili de yeni düzenlemeler yapılması gereklidir.

6.5 Servis Kalitesi

Pratikte servis kalitesi yönetimi (QOS management) çokluortam sunum yapan bir sistemin en önemli fonksiyonlarından biridir. Servis kalitesi yönetimi gecikme, eşzamansızlık ve kayma gibi bir dizi parametrenin yönetimini gerektir ve en çok ortam bilgisinin istemciye aktarımında önemlidir.

Çokluortam Haberler Sistemi, şu anda servis kalitesi yönetimi için bir imkan sağlamamaktadır. Bunun bir sebebi de hat kapasitelerinin çok değişken olduğu Internette, servis kalitesi parametrelerinin, belki de birçok istemci bilgisayar için sağlanamayacak olmasıdır. Ayrıca, Çokluortam Haberler Sistemi'nde bu hizmetin verilebilmesi için, daha alt seviyedeki iletişim katmanlarının da buna destek vermesi gerekmektedir. Şu anda yaygın olarak kullanılan HTTP ve TCP/IP (Transport Control Protocol/Internet Protocol) versiyonlarında bu destek yoktur ancak çalışmalar göstermektedir ki çokluortam desteği Internet için uzak bir gelecekte değildir. Halen, düşük seviyeli ağ katmanlarında ATM (Asynchronous Transfer Mode), FR (Frame Relay) ve FDDI-II (Fiber Distributed Data Interface – II); orta seviyeli ağ katmanlarında IPng (IP next generation veya IPv6); yüksek seviyeli ağ katmanlarında ise HTTP 2.0, HTML 4.0 gibi çokluortam veriyi de destekleyen iletişim protokolleri ve belge modelleri üzerinde, standartların oluşturulması ve uygulamada oturtulması sürecini yaşıyoruz.

6.6 Bilgi Girişi

Çokluortam sistemlerde veri çok yoğun olduğundan, verinin toplanması ve düzenlenmesi için de özel teknikler gerekmektedir.

Çokluortam Haberler Sistemi, sayfa ve çokluortam veri düzenlemesi için imkanlar sağlamasına rağmen, verinin dış dünyadan toplanmasını kolaylaştıracak hiçbir özelliğe sahip değildir.

Böylece, sistemin daha da geliştirilmesi için yapılacak önerilerden biri olarak resim, ses ve video verisinin pratik olarak toplanmasını sağlayacak ekleri

sayabiliriz. Hatta, bu ekler daha geliştirilerek programın gerçek zamanlı olarak veriyi toplayıp konuşma, karakter, resim ve video tanıma araçlarını kullanarak çokluortam doküman kompozisyonunu da, kullanıcı aracılığı olmadan yapabilir.



7. KAYNAKLAR

(Agarwal et al., 1994), A.M. Keller Stanford University, R. Jansen Persistence Software, S. Agarwal Persistence Software, "Bridging Object-Oriented Programming and Relational Databases", 1994

(Agarwal et al., 1995), A.M. Keller Stanford University, C. Keene Persistence Software, S. Agarwal Persistence Software, "Architecting Object Applications for High Performance with Relational Databases", 1995

(Baker and Khoshafian, 1996), S. Khoshafian, A.B. Baker, "Multimedia and Imaging Databases", Morgan Kaufmann Publishers 1996.

(Berra et al., 1996), K.C. Nwosu, B. Thuraisingham, P.B. Berra, Multimedia Database Systems - Design and Implementation Strategies, Kluwer Academic Publishers 1996.

(Boll et al., 1996), S. Boll, W. Klas, M. Löhr, Integrated Database Services for Multimedia Presentations, Multimedia Information Storage, Kluwer Academic Publishers 1996.

(Chakravarthy et al., 1993), S. Chakravarthy, V.Krishnaprasad, Z. Tamizuddin, F. Lambay, "A Federated Multimedia DBMS for Medical Research: Architecture and Functionality", Dept. of Comp. Science, University of Florida, 1993.

(DeVries, 1995), A.P. DeVries, "Multimedia Information Access", Database Systems Group, Faculty of Computer Science, University of Twente, 1995

(Docherty, 1993), M.H. O'Docherty, "A Multimedia Information System with Automatic Content Retrieval", TR-UMCS-93-2-2, Dept. of Comp. Science, University of Manchester, 1993

(ElMadani et al., 1994), C. Vittal, T. Özsu, D. Szarfon, G. ElMadani, "The Logical Design of a Multimedia Database for a News-On-Demand Application", TR 94-16 Dept. of Computer Science, University of Alberta, 1994.

(ElMadani, 1996), S. ElMadani, "Support for Document Entry in the Multimedia Database", Dept. of Computer Science, University of Alberta, 1996.

(Galchuk et al., 1995), N. Hirzalla, B. Galchuk, and A. Karmouch. "A Temporal Model For Interactive Temporal Scenarios". IEEE Multimedia, pages 24-31, Fall 1995.

(Gemmell et al., 1994), D.J. Gemmell, H.M. Vin, D. Kandlur, V. Rangan, Multimedia "Storage Servers: A Tutorial and Survey", IEEE Computer, 1995

(Gettys et al., 1997), H.F. Nielsen, J. Gettys, A. Smith, E. Prud'hommeaux, H.W. Lie, C. Lilley, "Network Performance Effects of HTTP/1.1, CSS1 and PNG", WWW Consortium Technical Report, February 1997.

(Gollapudi and Zhang a), A. Zhang, S. Gollapudi, "Multimedia Transaction Management in Database Systems", Dept. of Comp. Science, State University of New York.

(Gollapudi and Zhang b), A. Zhang, S. Gollapudi, NetMedia : "A Client-Server Distributed Multimedia Database Environment", Dept. of Comp. Science, State University of New York.

(Gottemukkala et al., 1996), B.Reinwald, T.J. Lehman, H. Pirahesh, V. Gottemukkala, "Storing and Using Objects in a Relational Database", IBM Systems Journal Vol 35 No 2, 1996

(Goyal et al., 1994), P.J. Shenoy, P. Goyal, H.M. Vin, "Issues in Multimedia Server Design", Distributed Multimedia Computing Lab., Dept. of Comp. Science, University of Texas, 1994.

(Hodges and Ramanathan, 1996), S. Ramanathan, J. Hodges, "Reverse Engineering Relational Schemas to Object-Oriented Schemas", TR-MSU-960701, Dept. of Comp. Science, Mississippi State University, July 1996

(IBM, 1993), "SOM Objects", User's Guide, IBM Corporation, June 1993

(Jagadish and Story, 1993), G. Story, H.V. Jagadish, "A Data Model for Multimedia Documents", AT&T Bell Laboratories, 1993.

(Jajodia and Subrahmanian, 1996), V.S. Subrahmanian, S. Jajodia, "Multimedia Database Systems - Issues and Research Directions, Springer" 1996.

(Kim 1994), YuMi Kim, Multimedia Database Systems Chapter 42, 1994.

(Lau 1994), C. Lau, "Object Oriented Programming Using SOM and DSOM", 1994

(Löhr et al., 1995), T. Rakow, E.J. Neuhold, and M. Löhr. Multimedia Database Systems - The Notions And The Issues. In G. Lausen, editor, Tagungsband GI-Fachtagung Datenbanksysteme in Büro, Technik und Wissenschaft (BTW), Dresden März 1995.

(Meyer and Wegener, 1991), K. Meyer-Wegener. Multimedia Databases. 1991.

(Mogul et al., 1997), Venkata N. Padmanabhan (University of California -- Berkeley), Jeffrey C. Mogul (Digital Equipment Corporation Western Research Laboratory), "Improving HTTP Latency", 1997.

(Nahrstedt and Steinmetz, 1995), R. Steinmetz, K. Nahrstedt. Multimedia : Computing, Communications and Applications. Prantice Hall 1995.

(Rakow and Böhm, 1994), K. Böhm, T.C. Rakow, "Metadata for Multimedia Documents", SIGMOD-Special Issue on Metadata for Digital Media, Vol 23 No 4, Dec. 1994.

(Ramanathan and Koduri 1996), C. Ramanathan, S. Koduri, "An Object-Oriented Prototype for a Geophysical Database", Dept. of Comp. Science, Mississippi State University, 1996

(Ramanathan, 1996), C. Ramanathan, "Providing Object-Oriented Access to a Relational Database", Dept. of Comp. Science, Mississippi State University, 1996

(Rao et al., 1995), S. Rao, H.M. Vin, A. Tarafdar, "Comperative Evaluation of Server-Push and Client-Pull Architectures for Multimedia Servers", Distributed Multimedia Computing Lab., Dept. of Comp. Science, University of Texas, 1995.

(Recker et al., 1995), M. Recker, A. Ram, T. Shikano, G. Li, J. Stasko, "Cognitive Media Types for Multimedia Information Access", Collage Of Computing, Georgia Inst. of Tech. Atalanta, 1995.

(RFC1866), Request For Comments 1866, "Hypertext MarkUp Language 2.0 Specification", November 1995.

(RFC1867), Request For Comments 1867, "Form-Based Upload in HTML", November 1995.

(RFC1945), Request For Comments 1945, "Hypertext Transfer Protocol 1.0 Specification", May 1996.

(RFC2045), Request For Comments 2045, "Multipurpose Internet Mail Extensions (MIME), Part One : Format of Internet Message Bodies", November 1996.

(RFC2046), Request For Comments 2046, "Multipurpose Internet Mail Extensions (MIME), Part Two : Media Types", November 1996.

(RFC2047), Request For Comments 2047, "Multipurpose Internet Mail Extensions (MIME), Part Three : Message Header Extensions for Non-ASCII Text", November 1996.

(RFC2048), Request For Comments 2048, "Multipurpose Internet Mail Extensions (MIME), Part Four : Registration Procedures", November 1996.

(RFC2049), Request For Comments 2049, "Multipurpose Internet Mail Extensions (MIME), Part Five : Conformance Criteria and Examples", November 1996.

(RFC2068), Request For Comments 2068, "Hypertext Transfer Protocol 1.1 Specification", January 1997.

(RFC822), Request For Comments 822, "Standard for The Format of ARPA Internet Text Messages", 1982.

(Rothermel and Wahl, 1994), T. Wahl and K. Rothermel. "Representing Time in Multimedia Systems". Proceedings of the International Conference on Multimedia Computing and Systems, pages 538-543, May 1994.

(Schöne, 1996), M. Schöne, "A Generic Type System for an Object-Oriented Multimedia Database System", Dept. of Computer Science, University of Alberta, 1996.

(Sessions 1996), R. Sessions, "Object Persistence - Beyond Object Oriented Databases", 1996

(Wei, 1994), Sha Xin Wei, "Situating Multimedia Databases With Respect To WWW, Portflio, PREMO, Hyper-G, ScriptX", Standford University, 1994.

(Zhang), A. Zhang, "On Synchronised Presentation Management in Multimedia" Database Systems, Dept. of Comp. Science, State University of New York.

8. EKLER

8.1 A : Tanım Listeleri

- Haber Tipleri

1. TV/Radyo Haberleri

- ◆ Haberler
- ◆ Eğlence
- ◆ Belgesel
- ◆ Sinema/Tiyatro
- ◆ Reklamlar
- ◆ Diğer

2. Gazete Haberleri

- ◆ Haberler
- ◆ Makale
- ◆ Yazı Dizisi
- ◆ Ropörtaj
- ◆ İlanlar
- ◆ Eğlence
- ◆ Magazin
- ◆ TV/Radyo programları
- ◆ Diğer

- Haber Konuları

- ◆ Politika

Hükümet

Muhalefet

- ◆ Ekonomi

Enflasyon

Ekonomik Kriz

- ◆ Suç

Cinayet

Soygun

Tecavüz

Terörizm

Kan Davası

- ◆ Sosyal

Miting

- Önem Seviyeleri

1. Değerlendirme Dışı
2. Normal
3. Önemli
4. Hayati
5. Flaş

- Saklama Ortamı Tipleri

- ◆ Video Bant
- ◆ Ses Bandı
- ◆ Kağıt
- ◆ Dijital

- Roller

- ◆ Yazan/Hazırlayan
- ◆ Maktul
- ◆ Zanlı
- ◆ Suç Ortağı



8.2 B : Veritabanı Şeması SQL Cümleleri

8.2.1 Domen tanım cümleleri

```
CONNECT "D:\ONDER\MMNEWS\DATA\NEWSDB.GDB"
USER "SYSDBA" PASSWORD "masterkey";
```

```
CREATE DOMAIN DOM_OID
  AS NUMERIC (9,0);
```

```
CREATE DOMAIN DOM_KLASOR_OID
  AS NUMERIC (9,0),
  CHECK ((VALUE IS NULL ) OR
        (VALUE >= 10000000) OR
        (VALUE <= 19999999));
```

```
CREATE DOMAIN DOM_DOCUMENT_OID
  AS NUMERIC (9,0),
  CHECK ((VALUE IS NULL ) OR
        (VALUE >= 20000000), OR
        (VALUE <= 29999999));
```

```
CREATE DOMAIN DOM_PAGE_OID
  AS NUMERIC (9,0),
  CHECK ((VALUE IS NULL ) OR
        (VALUE >= 30000000) OR
        (VALUE <= 39999999));
```

```
CREATE DOMAIN DOM_VIDEO_OID
  AS NUMERIC (9,0),
  CHECK ((VALUE IS NULL ) OR
        (VALUE >= 40000000) OR
        (VALUE <= 49999999));
```

```
CREATE DOMAIN DOM_AUDIO_OID
  AS NUMERIC (9,0),
  CHECK ((VALUE IS NULL ) OR
        (VALUE >= 50000000) OR
        (VALUE <= 59999999));
```

```
CREATE DOMAIN DOM_IMAGE_OID
  AS NUMERIC (9,0),
  CHECK ((VALUE IS NULL ) OR
        (VALUE >= 60000000) OR
        (VALUE <= 69999999));
```

```
CREATE DOMAIN DOM_TEXT_OID
  AS NUMERIC (9,0),
  CHECK ((VALUE IS NULL ) OR
        (VALUE >= 80000000) OR
        (VALUE <= 89999999));
```

```
CREATE DOMAIN DOM_CONTENT_OID
  AS NUMERIC (9,0),
  CHECK ((VALUE IS NULL ) OR
        (VALUE >= 90000000) OR
        (VALUE <= 99999999));
```

```
CREATE DOMAIN DOM_LOCATION_OID
AS NUMERIC (9,0),
CHECK ((VALUE IS NULL ) OR
(VALUE >= 100000000) OR
(VALUE <= 109999999));

CREATE DOMAIN DOM_ROLE_OID
AS NUMERIC (9,0),
CHECK ((VALUE IS NULL ) OR
(VALUE >= 110000000) OR
(VALUE <= 119999999));

CREATE DOMAIN DOM_PERSON_OID
AS NUMERIC (9,0),
CHECK ((VALUE IS NULL ) OR
(VALUE >= 120000000) OR
(VALUE <= 129999999));

CREATE DOMAIN DOM_STORAGE_INFO_OID
AS NUMERIC (9,0),
CHECK ((VALUE IS NULL ) OR
(VALUE >= 140000000) OR
(VALUE <= 149999999));

CREATE DOMAIN DOM_NEWS_TYPE_OID
AS NUMERIC (9,0),
CHECK ((VALUE IS NULL ) OR
(VALUE >= 150000000) OR
(VALUE <= 159999999));

CREATE DOMAIN DOM_NEWS_SUBJECT_OID
AS NUMERIC (9,0),
CHECK ((VALUE IS NULL ) OR
(VALUE >= 160000000) OR
(VALUE <= 169999999));

CREATE DOMAIN DOM_IMPORTANCE_OID
AS NUMERIC (9,0),
CHECK ((VALUE IS NULL ) OR
(VALUE >= 170000000) OR
(VALUE <= 179999999));

CREATE DOMAIN DOM_HTML_ANNOTATIONS_OID
AS NUMERIC (9,0),
CHECK ((VALUE IS NULL ) OR
(VALUE >= 190000000) OR
(VALUE <= 199999999));

CREATE DOMAIN DOM_STORAGE_MEDIA_TYPE_OID
AS NUMERIC (9,0),
CHECK ((VALUE IS NULL ) OR
(VALUE >= 200000000) OR
(VALUE <= 209999999));

CREATE DOMAIN DOM_MIME_TYPE_OID
AS NUMERIC (9,0),
CHECK ((VALUE IS NULL ) OR
(VALUE >= 210000000) OR
(VALUE <= 219999999));
```

```

CREATE DOMAIN DOM_DOC_STORAGE_OID
  AS NUMERIC (9,0),
  CHECK ((VALUE IS NULL ) OR
        (VALUE >= 220000000) OR
        (VALUE <= 229999999));
COMMIT;

CREATE DOMAIN DOM_WEB_LINKS_OID
  AS NUMERIC (9,0),
  CHECK ((VALUE IS NULL ) OR
        (VALUE >= 250000000) OR
        (VALUE <= 259999999));
COMMIT;

CREATE DOMAIN DOM_USERNAME AS VARCHAR(20) DEFAULT USER;
CREATE DOMAIN DOM_PATH_NAME AS VARCHAR(200) DEFAULT NULL;
CREATE DOMAIN DOM_SHORT_NAME AS CHAR(30);
CREATE DOMAIN DOM_LONG_NAME AS CHAR(100);
CREATE DOMAIN DOM_VIDEO_FEATURE_VECTOR CHAR(100);
CREATE DOMAIN DOM_IMAGE_FEATURE_VECTOR CHAR(100);
CREATE DOMAIN DOM_AUDIO_FEATURE_VECTOR CHAR(100);
CREATE DOMAIN DOM_KEYWORDS VARCHAR(200);
CREATE DOMAIN DOM_DESCRIPTION AS VARCHAR(200);
CREATE DOMAIN DOM_TIME AS CHAR(8);
COMMIT;

DISCONNECT;

```

8.2.2 Üretici tanım cümleleri

```

CONNECT "D:\ONDER\MMNEWS\DATA\NEWSDB.GDB"
USER "SYSDBA" PASSWORD "masterkey";

CREATE GENERATOR GN_KLASOR_OID ;
SET GENERATOR GN_KLASOR_OID TO 10000000 ;

CREATE GENERATOR GN_DOCUMENT_OID ;
SET GENERATOR GN_DOCUMENT_OID TO 20000000 ;

CREATE GENERATOR GN_PAGE_OID ;
SET GENERATOR GN_PAGE_OID TO 30000000 ;

CREATE GENERATOR GN_VIDEO_OID ;
SET GENERATOR GN_VIDEO_OID TO 40000000 ;

CREATE GENERATOR GN_AUDIO_OID ;
SET GENERATOR GN_AUDIO_OID TO 50000000 ;

CREATE GENERATOR GN_IMAGE_OID ;
SET GENERATOR GN_IMAGE_OID TO 60000000 ;

CREATE GENERATOR GN_TEXT_OID ;
SET GENERATOR GN_TEXT_OID TO 80000000 ;

CREATE GENERATOR GN_CONTENT_OID ;
SET GENERATOR GN_CONTENT_OID TO 90000000 ;

```

```

CREATE GENERATOR GN_LOCATION_OID ;
SET GENERATOR GN_LOCATION_OID TO 100000000 ;

CREATE GENERATOR GN_ROLE_OID ;
SET GENERATOR GN_ROLE_OID TO 110000000 ;

CREATE GENERATOR GN_PERSON_OID ;
SET GENERATOR GN_PERSON_OID TO 120000000 ;

CREATE GENERATOR GN_NEWS_TYPE_OID ;
SET GENERATOR GN_NEWS_TYPE_OID TO 150000000 ;

CREATE GENERATOR GN_NEWS_SUBJECT_OID ;
SET GENERATOR GN_NEWS_SUBJECT_OID TO 160000000 ;

CREATE GENERATOR GN_IMPORTANCE_OID ;
SET GENERATOR GN_IMPORTANCE_OID TO 170000000 ;

CREATE GENERATOR GN_HTML_ANNOTATIONS_OID ;
SET GENERATOR GN_HTML_ANNOTATIONS_OID TO 190000000 ;

CREATE GENERATOR GN_STORAGE_MEDIA_TYPE_OID ;
SET GENERATOR GN_STORAGE_MEDIA_TYPE_OID TO 200000000 ;

CREATE GENERATOR GN_MIME_TYPE_OID ;
SET GENERATOR GN_MIME_TYPE_OID TO 210000000 ;

CREATE GENERATOR GN_DOC_STORAGE_OID ;
SET GENERATOR GN_DOC_STORAGE_OID TO 220000000 ;

CREATE GENERATOR GN_VIDEO_FEATURE_VECTORS_OID ;
SET GENERATOR GN_VIDEO_FEATURE_VECTORS_OID TO 230000000 ;

CREATE GENERATOR GN_AUDIO_FEATURE_VECTORS_OID ;
SET GENERATOR GN_AUDIO_FEATURE_VECTORS_OID TO 240000000 ;

CREATE GENERATOR GN_WEB_LINKS_OID ;
SET GENERATOR GN_WEB_LINKS_OID TO 250000000 ;

COMMIT;

```

8.2.3 Tablo ve indeks tanım cümleleri

```

CONNECT "D:\ONDER\MMNEWS\DATA\NEWSDB.GDB"
USER "SYSDBA" PASSWORD "masterkey";

CREATE TABLE KLASOR
(
    OID          DOM_KLASOR_OID DEFAULT -1 NOT NULL PRIMARY KEY,
    SEVIYE       INTEGER,
    PARENT_OID   DOM_KLASOR_OID,
    ISIM         CHARACTER(100), DEFAULT 'Yeni Klasör'
);

COMMIT;

CREATE UNIQUE INDEX KLASOR_IDX_PK ON KLASOR (OID);
CREATE UNIQUE INDEX KLASOR_IDX_PARENTOID_ISIM ON KLASOR
(PARENT_OID, ISIM);
COMMIT;

```

```

CREATE TABLE HTML_ANNOTATIONS
(
  OID          DOM_HTML_ANNOTATIONS_OID DEFAULT -1
              NOT NULL PRIMARY KEY,
  ISIM        CHAR(30) DEFAULT 'Yeni Stil' NOT NULL,
  BAS_ETIKETI CHAR(20),
  SON_ETIKETI CHAR(20),
  FONT_ISIM   CHAR(20) DEFAULT 'Arial',
  FONT_BOY    INTEGER DEFAULT 10,
  FONT_RENK   NUMERIC(9,0) DEFAULT 0,
  FONT_BOLD   INTEGER DEFAULT 0,
  FONT_ITALIC INTEGER DEFAULT 0,
  FONT_STRIKEOUT INTEGER DEFAULT 0,
  FONT_UNDERLINE INTEGER DEFAULT 0
);

COMMIT;

CREATE UNIQUE INDEX HTML_ANNOTATIONS_IDX_PK ON
HTML_ANNOTATIONS (OID);
CREATE UNIQUE INDEX HTML_ANNOTATIONS_IDX_ISIM ON
HTML_ANNOTATIONS (ISIM);
COMMIT;

CREATE TABLE LOCATION
(
  OID DOM_LOCATION_OID DEFAULT -1 NOT NULL PRIMARY KEY,
  NAME DOM_LONG_NAME,
  DOMESTIC_CHAR /* 'E' OR 'H' *//,
  COUNTRY DOM_SHORT_NAME,
  STATE DOM_SHORT_NAME,
  DISTRICT DOM_LONG_NAME,
  REGION DOM_LONG_NAME
);

COMMIT;

CREATE TABLE DOC_IMPORTANCE
(
  OID DOM_IMPORTANCE_OID DEFAULT -1 NOT NULL PRIMARY KEY,
  NAME DOM_SHORT_NAME
);

CREATE TABLE ROLE
(
  OID DOM_ROLE_OID DEFAULT -1 NOT NULL PRIMARY KEY,
  NAME DOM_SHORT_NAME
);

COMMIT;

CREATE TABLE PERSON
(
  OID DOM_PERSON_OID DEFAULT -1 NOT NULL PRIMARY KEY,
  NAME DOM_LONG_NAME
);

```


COMMIT;

```
CREATE TABLE RELATED_DOCUMENTS
(
  DOC1 DOM_DOCUMENT_OID NOT NULL,
  DOC2 DOM_DOCUMENT_OID NOT NULL,
  PRIMARY KEY (DOC1, DOC2)
);
```

COMMIT;

```
CREATE TABLE RELATED_PERSONS
(
  DOC DOM_DOCUMENT_OID NOT NULL,
  PERSON DOM_PERSON_OID NOT NULL,
  ROLE DOM_ROLE_OID NOT NULL,
  PRIMARY KEY (DOC, PERSON, ROLE)
);
```

COMMIT;

```
CREATE TABLE NEWS_SUBJECT
(
  OID DOM_NEWS_SUBJECT_OID DEFAULT -1 NOT NULL
  PRIMARY KEY,
  NAME DOM_SHORT_NAME
);
```

```
CREATE TABLE NEWS_TYPE
(
  OID DOM_NEWS_TYPE_OID DEFAULT -1 NOT NULL PRIMARY KEY,
  NAME DOM_SHORT_NAME
);
```

COMMIT;

```
CREATE TABLE DOCUMENTS
(
  OID DOM_DOCUMENT_OID DEFAULT -1 NOT NULL
  PRIMARY KEY,
  NAME DOM_LONG_NAME,
  FOLDER_OID DOM_KLASOR_OID,
  DOC_DATE DATE,
  CREATOR DOM_USERNAME,
  DESCRIPTION DOM_DESCRIPTION,
  HEADER DOM_LONG_NAME,
  ABSTRACT VARCHAR(2000),
  KEYWORDS DOM_KEYWORDS,
  DOC_STORAGE DOM_OID,
  COPYRIGHT DOM_OID,
  SOURCE DOM_OID,
  APPEARED_IN DOM_PERSON_OID,
  APPEARED_AT DATE,
  STARTED CHAR(20),
  ENDED CHAR(20),
  TYPE DOM_NEWS_TYPE_OID,
  SUBJECT DOM_NEWS_SUBJECT_OID,
  LOCATION DOM_LOCATION_OID,
```

```
IMPORTANCE  DOM_IMPORTANCE_OID
);
```

```
COMMIT;
```

```
CREATE TABLE DOC_PAGE
(
  OID      DOM_PAGE_OID  DEFAULT -1 NOT NULL PRIMARY KEY,
  NAME     DOM_LONG_NAME DEFAULT 'Yeni Sayfa' NOT NULL,
  OWNER    DOM_DOCUMENT_OID
          CONSTRAINT FK_PAGE_DOCUMENT_OID
          REFERENCES DOCUMENTS(OID),
  TEXT     DOM_TEXT_OID,
  PAGENO   INTEGER
);
```

```
CREATE UNIQUE INDEX DOC_PAGE_IDX_OWNER_PAGENO
ON DOC_PAGE (OWNER, PAGENO);
```

```
COMMIT;
```

```
CREATE TABLE MIME_TYPE
(
  OID      DOM_MIME_TYPE_OID DEFAULT -1 NOT NULL PRIMARY KEY,
  NAME     DOM_SHORT_NAME,
  TYPE     DOM_SHORT_NAME
);
```

```
CREATE TABLE TEXT
(
  OID      DOM_TEXT_OID  DEFAULT -1 NOT NULL PRIMARY KEY,
  NAME     DOM_LONG_NAME,
  TEXT_DATA BLOB SUB_TYPE TEXT SEGMENT SIZE 400
);
```

```
COMMIT;
```

```
CREATE TABLE TEXT_ATTRIBUTES
(
  TEXT      DOM_TEXT_OID
          CONSTRAINT FK_TEXT_ATTRIBUTES_TEXT_OID
          REFERENCES TEXT(OID),
  ANNOTATION DOM_HTML_ANNOTATIONS_OID,
  OBJECT     DOM_OID, /*FOR APPLETS AND IMAGES*/
  START     INTEGER,
  LEN       INTEGER DEFAULT 0,
  INLINEPATH DOM_PATH_NAME
);
```

```
COMMIT;
```

```
CREATE UNIQUE INDEX TEXT_ATTRIBUTES_IDX_PK ON TEXT_ATTRIBUTES
(TEXT, ANNOTATION, OBJECT, START, LEN, INLINEPATH);
```

```
CREATE TABLE DOC_STORAGE
(
  OID      DOM_DOC_STORAGE_OID DEFAULT -1 NOT NULL PRIMARY KEY,
```

```

MEDIA_TYPE DOM_STORAGE_MEDIA_TYPE_OID,
ROOM      CHAR(10),
SECTION   CHAR(10),
SHELF     CHAR(10),
ROW       CHAR(10),
COL       CHAR(10),
ORDERNUM  CHAR(10)
);

COMMIT;

CREATE TABLE WEB_LINKS
(
  OID DOM_WEB_LINKS_OID
      DEFAULT -1 NOT NULL PRIMARY KEY,
  NAME DOM_SHORT_NAME,
  WEBADDRESS DOM_LONG_NAME
);

CREATE TABLE MEDIA_TYPES
(
  OID DOM_STORAGE_MEDIA_TYPE_OID
      DEFAULT -1 NOT NULL PRIMARY KEY,
  NAME DOM_SHORT_NAME NOT NULL
);

COMMIT;

CREATE TABLE VIDEO
(
  OID DOM_VIDEO_OID DEFAULT -1 NOT NULL
      PRIMARY KEY,
  NAME DOM_SHORT_NAME NOT NULL,
  DESCRIPTION DOM_DESCRIPTION,
  CONTENT DOM_CONTENT_OID,
  KEYWORDS DOM_KEYWORDS,
  WIDTH INTEGER,
  HEIGHT INTEGER,
  LEN DOM_TIME
);

COMMIT;

CREATE TABLE VIDEO_FEATURE_VECTORS
(
  OID DOM_VIDEO_OID DEFAULT -1 NOT NULL,
  FRAME INTEGER,
  START_TIME DOM_TIME,
  END_TIME DOM_TIME,
  FEATURE_VECTOR DOM_VIDEO_FEATURE_VECTOR
);

COMMIT;

CREATE UNIQUE INDEX VIDEO_FEATURE_VECTORS_IDX_PK ON
VIDEO_FEATURE_VECTORS (OID, FRAME, START_TIME, END_TIME);

```

```

CREATE TABLE AUDIO
(
  OID DOM_AUDIO_OID DEFAULT -1 NOT NULL PRIMARY KEY,
  NAME DOM_SHORT_NAME NOT NULL,
  DESCRIPTION DOM_DESCRIPTION,
  CONTENT DOM_CONTENT_OID,
  KEYWORDS DOM_KEYWORDS,
  LEN DOM_TIME
);

COMMIT;

CREATE TABLE AUDIO_FEATURE_VECTORS
(
  OID DOM_AUDIO_OID NOT NULL,
  FRAME INTEGER,
  START_TIME DOM_TIME,
  END_TIME DOM_TIME,
  FEATURE_VECTOR DOM_AUDIO_FEATURE_VECTOR
);

COMMIT;

CREATE UNIQUE INDEX AUDIO_FEATURE_VECTORS_IDX_PK
ON AUDIO_FEATURE_VECTORS
(OID, FRAME, START_TIME, END_TIME);

CREATE TABLE IMAGE
(
  OID DOM_IMAGE_OID DEFAULT -1 NOT NULL PRIMARY KEY,
  NAME DOM_SHORT_NAME,
  DESCRIPTION DOM_DESCRIPTION,
  CONTENT DOM_CONTENT_OID,
  KEYWORDS DOM_KEYWORDS,
  WIDTH INTEGER,
  HEIGHT INTEGER,
  FEATURE_VECTOR DOM_IMAGE_FEATURE_VECTOR
);

COMMIT;

CREATE TABLE CONTENT
(
  OID DOM_CONTENT_OID DEFAULT -1 NOT NULL PRIMARY KEY,
  NAME DOM_SHORT_NAME,
  DESCRIPTION DOM_LONG_NAME,
  C_STORAGE DOM_DOC_STORAGE_OID,
  CPATH DOM_PATH_NAME,
  RAWDATA BLOB SEGMENT SIZE 100,
  OBJECT_TYPE DOM_MIME_TYPE_OID
);

COMMIT;

```

```
SET TERM ##;

CREATE TRIGGER TR_KLASOR_BI FOR KLASOR BEFORE INSERT POSITION 0
AS
BEGIN
    NEW.OID = GEN_ID(GN_KLASOR_OID,1);
END ##

CREATE TRIGGER TR_HTML_ANNOTATIONS_BI FOR HTML_ANNOTATIONS
BEFORE INSERT POSITION 0 AS
BEGIN
    NEW.OID = GEN_ID(GN_HTML_ANNOTATIONS_OID,1);
END ##

CREATE TRIGGER TR_LOCATION_BI FOR LOCATION BEFORE INSERT
POSITION 0 AS
BEGIN
    NEW.OID = GEN_ID(GN_LOCATION_OID,1);
END ##

CREATE TRIGGER TR_DOC_IMPORTANCE_BI FOR DOC_IMPORTANCE BEFORE
INSERT POSITION 0 AS
BEGIN
    NEW.OID = GEN_ID(GN_IMPORTANCE_OID,1);
END ##

CREATE TRIGGER TR_ROLE_BI FOR ROLE BEFORE INSERT POSITION 0 AS
BEGIN
    NEW.OID = GEN_ID(GN_ROLE_OID,1);
END ##

CREATE TRIGGER TR_PERSON_BI FOR PERSON BEFORE INSERT POSITION 0
AS
BEGIN
    NEW.OID = GEN_ID(GN_PERSON_OID,1);
END ##

CREATE TRIGGER TR_DOCUMENTS_BI FOR DOCUMENTS BEFORE INSERT
POSITION 0 AS
BEGIN
    NEW.OID = GEN_ID(GN_DOCUMENT_OID,1);
END ##

CREATE TRIGGER TR_DOC_PAGE_BI FOR DOC_PAGE BEFORE INSERT
POSITION 0 AS
DECLARE VARIABLE LASTPAGE INTEGER;
BEGIN
    NEW.OID = GEN_ID(GN_PAGE_OID,1);
    SELECT MAX(PAGENO), FROM DOC_PAGE
        WHERE OWNER = NEW.OWNER
        INTO :LASTPAGE;
    IF (LASTPAGE IS NULL), THEN
        NEW.PAGENO = 1;
    ELSE
        NEW.PAGENO = LASTPAGE + 1;
END ##

CREATE TRIGGER TR_MIME_TYPE_BI FOR MIME_TYPE BEFORE INSERT
POSITION 0 AS
BEGIN
    NEW.OID = GEN_ID(GN_MIME_TYPE_OID,1);
```

```
END ##

CREATE TRIGGER TR_PAGE_ANNOTATION_BI FOR PAGE_ANNOTATION BEFORE
INSERT POSITION 0 AS
BEGIN
    NEW.OID = GEN_ID(GN_PAGE_ANNOTATION_OID,1);
END ##

CREATE TRIGGER TR_TEXT_BI FOR TEXT BEFORE INSERT POSITION 0 AS
BEGIN
    NEW.OID = GEN_ID(GN_TEXT_OID,1);
END ##

CREATE TRIGGER TR_DOC_STORAGE_BI FOR DOC_STORAGE BEFORE INSERT
POSITION 0 AS
BEGIN
    NEW.OID = GEN_ID(GN_DOC_STORAGE_OID,1);
END ##

CREATE TRIGGER TR_MEDIA_TYPES_BI FOR MEDIA_TYPES BEFORE INSERT
POSITION 0 AS
BEGIN
    NEW.OID = GEN_ID(GN_STORAGE_MEDIA_TYPE_OID,1);
END ##

CREATE TRIGGER TR_VIDEO_BI FOR VIDEO BEFORE INSERT POSITION 0
AS
BEGIN
    NEW.OID = GEN_ID(GN_VIDEO_OID,1);
END ##

CREATE TRIGGER TR_AUDIO_BI FOR AUDIO BEFORE INSERT POSITION 0
AS
BEGIN
    NEW.OID = GEN_ID(GN_AUDIO_OID,1);
END ##

CREATE TRIGGER TR_IMAGE_BI FOR IMAGE BEFORE INSERT POSITION 0
AS
BEGIN
    NEW.OID = GEN_ID(GN_IMAGE_OID,1);
END ##

CREATE TRIGGER TR_APPLET_BI FOR APPLET BEFORE INSERT POSITION 0
AS
BEGIN
    NEW.OID = GEN_ID(GN_APPLET_OID,1);
END ##

CREATE TRIGGER TR_CONTENT_BI FOR CONTENT BEFORE INSERT POSITION
0 AS
BEGIN
    NEW.OID = GEN_ID(GN_CONTENT_OID,1);
END ##

CREATE TRIGGER TR_NEWS_TYPE_BI FOR NEWS_TYPE BEFORE INSERT
POSITION 0 AS
BEGIN
    NEW.OID = GEN_ID(GN_NEWS_TYPE_OID,1);
END ##
```

```
CREATE TRIGGER TR_NEWS_SUBJECT_BI FOR NEWS_SUBJECT BEFORE  
INSERT POSITION 0 AS  
BEGIN  
    NEW.OID = GEN_ID(GN_NEWS_SUBJECT_OID,1);  
END ##
```

```
CREATE TRIGGER TR_WEB_LINKS_BI FOR WEB_LINKS BEFORE INSERT  
POSITION 0 AS  
BEGIN  
    NEW.OID = GEN_ID(GN_WEB_LINKS_OID,1);  
END ##
```

```
SET TERM ; ##
```



9. ÖZGEÇMİŞ

Doğum Tarihi	10.04.1973	
Doğum Yeri	İstanbul	
Lise	1987-1990	Pertevniyal Lisesi, İstanbul
Lisans	1990-1995	Yıldız Teknik Üniversitesi Mühendislik Fakültesi Bilgisayar Bilimleri Mühendisliği Bölümü
Yüksek Lisans	1995-1998	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Bilimleri Mühendisliği Anabilim Dalı, Bilgisayar Bilimleri Mühendisliği Programı
Çalıştığı Kurum(lar)		
	1995-1997	Merlin Bilgisayar A.Ş.
	1997-1998	Çelebi Hava Servisi A.Ş.