

KIRIKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
YÜKSEK LİSANS TEZİ

WEB TABANLI EĞİTİM SİSTEMLERİNDE DERS İÇERİKLERİNİN SCORM
UYUMLU OLUŞTURULMASI AMACIYLA YAZILIM GELİŞTİRİLMESİ

Kadir HALTAŞ

Ocak 2016

Bilgisayar Mühendisliği Anabilim Dalında Kadir HALTAŞ tarafından hazırlanan WEB TABANLI EĞİTİM SİSTEMLERİNDE DERS İÇERİKLERİNİN SCORM UYUMLU OLUŞTURULMASI AMACIYLA YAZILIM GELİŞTİRİLMESİ adlı Yüksek Lisans Tezinin Anabilim Dalı standartlarına uygun olduğunu onaylarım.

Prof. Dr. Erdem Kamil YILDIRIM
Anabilim Dalı Başkanı

Bu tezi okuduğumu ve tezin **Yüksek Lisans Tezi** olarak bütün gereklilikleri yerine getirdiğini onaylarım.

Yrd. Doç. Dr. Murat LÜY
Danışman

Jüri Üyeleri

Başkan	: Doç. Dr. Necaattin BARIŞÇI	_____
Üye (Danışman)	: Yrd. Doç. Dr. Murat LÜY	_____
Üye	: Yrd. Doç. Dr. Halil Murat ÜNVER	_____

08/01/2016

Bu tez ile Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onaylamıştır.

Prof. Dr. Mustafa YİĞİTOĞLU
Fen Bilimleri Enstitüsü Müdürü

ÖZET

WEB TABANLI EĞİTİM SİSTEMLERİNDE DERS İÇERİKLERİNİN SCORM UYUMLU OLUŞTURULMASI AMACIYLA YAZILIM GELİŞTİRİLMESİ

HALTAŞ, Kadir

Kırıkkale Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi

Danışman: Yrd. Doç. Dr. Murat LÜY

Ocak 2016, 89 sayfa

Özellikle son 20 yılda kullanımı gittikçe artan uzaktan öğretim faaliyetleri pek çok birey için önemli fırsatlar sunmaktadır. Örgün olarak eğitim almak isteyen yetişkin bir bireyi sosyal ve ekonomik olarak önemli derecede etkilemektedir. Bir eğitim programına dâhil olmak bireyin bu program süresi zarfında farklı bir uğraşı içerisinde bulunmasını da engellemektedir. Aynı zamanda hangi eğitim programı olursa olsun bireyin maddi olarak da birçok harcama yapmasını gerektirmektedir. Özellikle çalışan bireylerin örgün bir eğitim programında bulunması oldukça zordur. Bu gibi kısıtlılıkları aşmak için uzaktan öğretim uygun bir eğitim faaliyeti olarak karşımıza çıkmaktadır.

Uzaktan öğretim faaliyetlerinde kullanılan yöntemler değişmektedir. Günümüzde uzaktan öğretim faaliyetleri çoğunlukla web teknolojileri üzerinden yürütülmektedir. Bu tip uzaktan öğretim faaliyetlerinde bilgisayarlara kurulan özel programlar sayesinde öğretim faaliyetine katılan bireylerin aynı

ortamda bulunma zorunluluđu ortadan kalkmaktadır. Öyle ki uzaktan öğretim faaliyetlerinin bazılarında bireyin öğretim faaliyetine eş zamanlı olarak katılmasına gerek kalmamaktadır.

Uzaktan öğretim faaliyetlerinin etkin olarak uygulanması, kullanılan alt yapının gelişmiş olmasıyla birlikte faaliyet içerisinde kullanılan öğretim materyallerinin bu ortam için uygun olarak hazırlanmasına da bağlıdır. Öğretim materyallerinin dijital ortama uygun olarak aktarılması için farklı yazılımlar kullanılmaktadır. Bu yazılımların kullanılması emek ve çoğunlukla konusunda uzman personel gerektirmektedir.

Pek çok eğitici personel, örgün öğretimde ders materyali veya yardımcı materyal olarak MS Office PowerPoint programıyla hazırlanan sunum dosyalarını kullanmaktadır. Bu çalışmada daha önceden hazırlanmış olan MS Office PowerPoint sunum dosyalarını web ortamına aktarmak için HTML dosyalarına dönüştürmek ve istenilirse dönüştürme işlemi sonucu olarak üretilen web ortamına uygun öğretim dokümanına SCORM uyumluluđu eklemek için yazılım geliştirilmiştir.

Anahtar kelimeler: Uzaktan Öğretim, PowerPoint, HTML5, CSS3, SCORM, Yazılım

ABSTRACT

DEVELOPING SOFTWARE WITH THE PURPOSE OF FORMING COURSE CONTENTS COMPATIBLE WITH SCORM IN WEB-BASED EDUCATION SYSTEMS

HALTAŞ, Kadir

Kırıkkale University

Graduate School of Natural and Applied Sciences

Department of Computer Engineering, Master Thesis

Supervisor: Assist. Prof. Dr. Murat LÜY

January 2016, 89 pages

Increased use of distance education activities especially in the last 20 years, offers significant opportunities for many individuals. Individuals receive formal training as a significant social and economic impact. Including a training program to prevent the presence in the individual a different pursuit over period of this program. It also requires multiple expenditures as property of the individual, no matter what the training program. Reside in a formal training program in particular is quite difficult working for individuals. To overcome these limitations such as distance education emerges as an appropriate educational activities.

Different methods can be used in distance education activities. Nowadays web technologies are mostly used in distance education activities. This type of distance education activities individuals involved in the training activities through specific programs that installed on computers, eliminate the requirement in the same environment. Such that the there is no need to

connect simultaneously to the individual training activities in some of the distance learning activities. The effectiveness of applying distance learning is depends on the infrastructure that is used and using appropriate training materials for this environment. Different software is used for the transmission of educational materials according to the digital environment. This increases the use of software and expert staff labor needs.

Many educational staff use presentation files that is prepared by using MS Office PowerPoint presentation software for course material or utility material. In this thesis a software has been developed which is converts MS Office PowerPoint presentation files that had been prepared previously, to HTML files for usage in web environment. Also if desired by using this developed software SCORM compliance can be added to the education document that is suitable for web environment, as a result of conversion process.

Key Words: Distance Learning, PowerPoint, HTML5, CSS3, SCORM, Software

TEŞEKKÜR

Tez çalışmam boyunca beni motive eden, yönlendiren, yardımlarını esirgemeyen danışman hocam Sayın Yrd. Doç. Dr. Murat LÜY'e teşekkür ederim. Akademik bilgi birimiyle çalışmalarımda bana ışık tutan Sayın hocam Yrd. Doç. Dr. Atilla ERGÜZEN'e yoğun iş temposunda bana vakit ayırdığı için teşekkür ederim. Akademik çalışmalarımı destekleyen, deneyimlerini paylaşan Doç. Dr. İsmail ÇAKIR'a ve tez araştırmalarımda bana yardımcı olan arkadaşlarım Öğr. Gör. Kutay CÜNERAY, M. Suat BİRECİKLİGİL ve Yrd. Doç. Dr. Erkan HİRİK'e teşekkür ederim.

Yüksek lisans eğitimim süresince zamanlarını çaldığım çok sevgili eşim Kübra HALTAŞ'a, çok değerli çocuklarım Betül Berra HALTAŞ'a ve Kayra HALTAŞ'a tüm içtenliğimle teşekkür ederim.

İÇİNDEKİLER DİZİNİ

Sayfa

ÖZET	i
ABSTRACT	iii
TEŞEKKÜR	v
İÇİNDEKİLER DİZİNİ.....	vi
ŞEKİLLER DİZİNİ.....	viii
ÇİZELGELER DİZİNİ.....	ix
KISALTMALAR DİZİNİ.....	x
1. GİRİŞ.....	1
1.1. Uzaktan Öğretim	1
1.2. Çalışmanın Amacı.....	4
1.3. Literatür Araştırması.....	5
2. MATERYAL, METOT VE YÖNTEM	9
2.1. Uzaktan Öğretim Sistemlerinin Çalışması.....	9
2.1.1. Kullanılan İletişim Teknolojisi.....	10
2.1.1.1. Mektupla Öğretim	10
2.1.1.2. Telekonferansla Uzaktan Öğretim	11
2.1.1.3. Bilgisayar ve WWW Destekli Uzaktan Öğretim	12
2.1.2. Zamanlama Çeşidi	14
2.1.2.1. Eş Zamanlı Öğretim.....	15
2.1.2.2. Eş Zamansız Öğretim	16
2.2. Öğretim Alt Yapısı	17
2.2.1. İçerik Referans Modelleri.....	18
2.2.2. Sanal Sınıf ve Diğer Sosyal Ortamlar	19
2.2.3. Kullanılan Altyapı Ve Teknolojiler	22
2.2.3.1. Öğretim Yönetim Sistemleri (ÖYS)	22
2.2.3.2. Flash.....	27
2.2.3.3. HTML.....	28
2.2.3.4. .NET Framework	33
2.2.3.5. JavaScript.....	34

2.3.	SCORM (Paylaşılabilir İçerik Nesne Referans Modeli)	35
2.3.1.	İçerik Birleşim Modeli	38
3.	GELİŞTİRİLEN YAZILIM.....	42
3.1.	Geliştirilen Yazılımın İşleyişi.....	42
3.1.1.	HTML Aktarım Modülü.....	43
3.1.2.	Ortam Aktarım Modülü	45
3.1.3.	Animasyon Aktarım Modülü.....	48
3.1.4.	SCORM Uyumluluk Modülü.....	50
3.2.	Geliştirilen Yazılımın Kullanımı	52
4.	SONUÇ.....	66
	KAYNAKLAR	68
	EKLER.....	71

ŞEKİLLER DİZİNİ

Şekil	Sayfa
2.1.	Örnek SCORM paketi manifesto dosyası [16]. 19
2.2.	Örnek sanal sınıf oturumu 21
2.3.	Moodle ÖYS web ara yüzünden örnek bir kesit. 25
2.4.	Moodle ÖYS'de yarım kalan dersin bırakılan yerden devam edilmesine ait ekran görüntüsü 26
2.5.	Örnek ActionScript kodları..... 28
2.6	Bir HTML sayfasına resim eklemek için kullanılan "" etiketine ve sayfa çıktısına bir örnek..... 29
2.7.	HTML 4.01 standardı kullanılarak hazırlanmış HTML sayfa kodlaması. 30
2.8.	Örnek varlık çeşitleri. 39
2.9.	Bir SCO nesnesi örnek gösterimi. 39
2.10.	Bir SCO nesnesinin hiyerarşik yapısı [22]. 40
2.11.	İçerik birleşiminin şematik gösterimi [22]. 41
3.1.	HTML, CSS ve JavaScript <i>aktarım algoritması akış şeması</i> 44
3.2.	Örnek HTML aktarım sonucu oluşan HTML kodlama..... 44
3.3.	Slayt dönüşüm işlemi akış şeması. 45
3.4.	Shape nesnesi dönüşüm algoritması akış şeması 46
3.5.	Nesnenin WMF formatlı dışa aktarım için kullanılan kod bloğu 47
3.6.	Nesnenin PNG formatlı dışa aktarımı için kullanılan kod bloğu 47
3.7.	Metin kutusunu HTML olarak dışa aktarmak için kullanılan kod bloğu. 48
3.8.	Animasyonların oluşturulmasına ait algoritma akış şeması 49
3.9.	Efekt dönüşüm akış şeması 50
3.10.	Hiyerarşi düzenleme algoritması akış şeması. 51
3.11.	SCORM aktarım işleminin akış şeması. 52
3.12.	Yazılımın menü görüntüsü 53
3.13.	Yazılım kontrol grupları 53
3.14.	HTML5 aktarım ayarları diyalog formu 54
3.15.	Örnek SVG formatı kodları 55
3.16.	Yazılım kullanılarak oluşturulmuş HTML sayfasından örnek 56
3.17.	Animasyonu aktarmak için kullanılan stil kodlarına örnek 57
3.18.	Aktarımı gerçekleştirilen slayta ait HTML sayfa görüntüsü 58
3.19.	"@keyframes" kullanımına bir örnek 59
3.20.	"@keyframes" anahtar kelimesinin yüzdelik dilimle kullanımı. 60
3.21.	Web tarayıcı üreticilerine göre stil etiketlerinin ön ek ile kullanımına örnek 61
3.22.	JavaScript ile geliştirilen "EffectObject" yapısı..... 62
3.23.	JavaScript kullanılarak geliştirilen "TimingObject" yapısı 62
3.24.	JavaScript kullanılarak geliştirilen "TimeLineObject" yapısı. 63
3.25.	Hiyerarşi düzenleme formu 64

ÇİZELGELER DİZİNİ

Çizelge

Sayfa

2.1.	HTML5 ile gelen yeni etiketler	31
3.1.	Popüler web tarayıcıların kullandıkları HTML yorum motorları ve CSS3 ön ekleri.	60



KISALTMALAR DİZİNİ

ÖYS	Öğretim Yönetim Sistemi
LMS	Learning Management System
SCORM	Sharable Content Object Reference Model
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
W3C	World Wide Web Consortium
IETF	Internet Engineering Task Force
AICC	Aviation Industry CBT (Computer-Based Training) Committee
XHTML	Extensible HyperText Markup Language
ADL	Advanced Distributed Learning
XML	Extensible Markup Language
SCO	Sharable Content Object
RTE	Runtime Environment
CLR	Common Language Runtime
MSIL	Microsoft Intermediate Language
JIT	Just In Time
JSON	JavaScript Object Notation
SVG	Scalable Vector Graphic
SMIL	Synchronized Multimedia Integration Language
CAT	Common Admission Test
CD	Compact Disk
IMS	Information Management System
SDK	Software Development Kit

1. GİRİŞ

Eğitim yüzyıllar boyunca önemini yitirmemiş bir alandır. Bilimsel gelişmelerin sonucu yeni bilgi birikimleri oluşmuş ve bunlar da yeni bilimsel gelişmelere ışık tutmuştur. Süreç bu şekilde devam ettikçe insanoğlunun bilgi birikimi doğal olarak önemli derecede artmıştır. Tüm bunlara bağlı olarak sanayi ve teknolojik alanlarda önemli gelişmeler gerçekleşmiştir. Sanayi devrimi, elektronik eşyaların kullanımına başlanması, internet teknolojilerinin kullanımı bu gelişmelere örnek olarak gösterilebilir. Gerçekleşen bu gelişmeler neticesinde ortaya çıkan bilgi birikiminin etkin kullanımı için daha nitelikli iş gücü ihtiyacı ortaya çıkmıştır. Nitelikli bireylerin yetişmesi eğitim kalitesiyle doğru orantılıdır. Dolayısıyla eğitimin önemi her geçen gün artar hâle gelmiştir. Bu bağlamda etkin ve kaliteli eğitim ortamları sağlanması için gerek kişi, kurum ve kuruluşlar gerekse yönetimler eğitim imkânlarını arttırarak, farklı ortamlar sunarak daha çok bireye daha kaliteli eğitim verilmesini hedeflemektedirler. Bu noktada kimi eğitim faaliyetleri için uygun bir alternatif olarak uzaktan öğretim yöntem(ler)i karşımıza çıkmaktadır.

1.1. Uzaktan Öğretim

Dünya genelinde uzaktan öğretim sıklıkla kullanılsa da net bir tanım üzerinden fikir birliğine ulaşılammıştır. Kısaca öğretici ve öğrencinin farklı ortam ve/veya zamanda bulunarak öğretim-öğrenme eylemlerinin

gerçekleşmesi olarak söylenebilir. Farklı eğitimciler ve farklı eğitim kurumlarınca değişik tanımlamalar getirilmiştir. Sıklıkla kullanılan tanımlar arasında şunları sayabiliriz:

California Uzaktan Öğretim Projesi kurumuna göre uzaktan öğretim: “Uzaktan öğretim, eğitsel kaynaklar ile öğrencileri buluşturan öğretim dağıtım sistemidir.” [1].

Michael MOORE ve Greg KEARSLY uzaktan öğretimi: “Uzaktan öğretim, öğretme ve öğrenme işinin farklı ortamlarda gerçekleştiği, iletişim için özel kurumsal organizasyonlar gibi teknolojilerin kullanıldığı; öğretme ve planlı öğrenmedir.” [2].

Bu tanımlardan da görüldüğü gibi uzaktan öğretim sistemlerinin temelinde farklı ortam ve zamanlarda bulunan öğretici ve öğrencilerin ortak bir teknoloji alt yapısı ve öğretim materyali kullanarak gerçekleşen öğretme işlemi bulunmaktadır. Her geçen gün artan dünya nüfusu, artan bilgi yoğunluğu ve bunlara bağlı olarak artan rekabetçi toplum ortamları geleneksel öğretim yöntemleri için alternatif arayışlarını artırmıştır. Alternatif yöntemlerden belki de en fazla tercih edilenler arasında sayılabilecek uzaktan öğretim sistemleri yüz yüze öğretim ortamların sağlayamadığı pek çok avantajla birlikte gelmektedir.

Yüz yüze öğretim ortamlarının çoğunda, birçok ders için sağlanamayan farklı ortam uyarıcıları uzaktan öğretim sistemlerinde rahatlıkla sağlanabilmektedir. Öğrenci uzaktan öğretim sistemlerinde kullanılan bilgisayar teknolojileri sayesinde bir ders materyali içerisinde içeriğe ait görsel, işitsel ve metin muhtevalı her türlü ortamı kullanılabilmektedir. Yine yüz yüze öğretim ortamlarında zaman, mekân öğretici ve öğrenci unsurlarının tümünün

varlıklarının kesişmesi gerekmektedir. Ancak uzaktan öğretim sistemlerinde gerçekleştirilen herhangi bir eğitsel faaliyet kayıt altına alınarak öğrenci, ortam ve zaman eşlemesi zorunluluğunu ortadan kaldırmaktadır. Bu durum özellikle günümüzde hem çalışan hem de öğrenci rolü bulunan bireylerin arttığı toplumlarda öğrenciler için büyük avantaj sağlamaktadır. Öğrenci katılmadığı ve/veya katılamadığı her hangi bir eğitsel faaliyeti sistemde tutulan kayıt sayesinde orijinal hâliyle, hiçbir eksiği olmadan gerçekleştirebilmektedir. Tüm bunların yanında uzaktan öğretimin önemli artılarından birisi de bireysel farklılıkların göz önünde bulundurulabilmesi ve buna bağlı olarak öğrencilere, öğretimde çok önemli bir fırsat olarak görülen kendi hızında öğrenme olanağı sağlamasıdır.

Uzaktan öğretim faaliyetlerini pek çok teknoloji kullanılarak sürdürmek mümkündür. Uzaktan öğretimin geçmişini incelediğimizde görülecektir ki tarihte mektup ve hatta gazete köşe yazıları dahi uzaktan öğretim faaliyetlerinin yürütülmesi için kullanılmıştır. Son on yıla bakıldığında uzaktan öğretim faaliyetlerinin ağırlıklı olarak bilgisayar teknolojilerini temel aldığını ve faaliyet ortamı olarak sanal dünyanın kullanıldığını görebiliriz.

Uzaktan öğretim popülarlığını arttırsa da hali hazırda büyük bir insan topluğu tarafından farklı sebeplerden dolayı çekinilen bir öğretim yöntemi olarak görülmektedir. Şüphesiz ki bu çekincelerin giderilmesi ancak öğretim kurumlarının uzaktan öğretime faaliyetlerini yaygınlaştırması ve kamu kurumlarının da uzaktan öğretim faaliyetlerinin sürdürülebilir olması için bir politika benimsemesi gerekir. Uzaktan öğretim faaliyetlerinin yaygınlaştırılması konusundaki en büyük etkenlerin başında kuşkusuz ki bu alanda nitelikli eleman sayısının ciddi anlamda az olması ve yürütülecek uzaktan öğretim

faaliyetinde görev almak isteyen öğretici personel sayısının az oluşudur. Bu noktada birey temelli olan öğretici personel eksiğinin giderilmesi çözümü öncelik taşıyan bir sorun olarak göze çarpmaktadır. Uzaktan öğretim faaliyetleri için gönüllü öğretici bulunamamasının sebepleri arasında, öğretim faaliyeti için gerekli olan materyal hazırlama, öğretim ortamında sosyal olarak da bulunma gibi hazırlık faaliyetlerinin öğreticiye büyük yük getirmesidir.

1.2. Çalışmanın Amacı

Sınıf içi öğretim faaliyetlerinde teknolojiden yararlanılmak istenildiğinde bunun birçok yöntemi mevcuttur. Bunlar arasında tepegöz kullanımı, ses kaydı dinletilmesi, video izletilmesi, bilgisayar destekli sunumlar yapılması vb. yöntemler sayılabilir. Şüphesiz ki hem görsel hem de işitsel olarak uyarıcılar barındıran bilgisayar destekli sunumlar eğitimciler açısından sıklıkla tercih edilmektedir. Burada bilgisayar destekli sunumlardan kastedilen MS PowerPoint, LibreOffice Impress gibi ofis yazılımlarınca hazırlanmış sunum dosyaları kullanılarak yapılan sunumlardır.

Eğiticiler bu gibi öğretim amaçlı sunumları, materyalleri hazırlamak için genel olarak düşünüldüğünde büyük zaman ve emek harcamaktadır. Ancak aynı eğitimci personel ilgili materyali herhangi SCORM (Sharable Content Object Reference Model – Paylaşılabilir İçerik Nesne Referans Modeli) uyumlu bir ÖYS (Öğretim Yönetim Sistemi) üzerinde kullanmak istediğinde bu sunum dosyası derse yardımcı malzeme olmaktan ileri gidememektedir. Bunun önüne

geçmek için içerik SCORM uyumlu ve ÖYS tarafından öğrencilere düzgün bir formatta sunulabilecek duruma getirilmelidir. Bu işlem ise eğiticinin hem belirli yazılım becerisinin olmasını hem de sunumu hazırlamak için harcadığı zaman ve emek kadar, belki de daha fazla emeği gerektirmektedir. Hatta belirli durumlar için profesyonel yardım almak durumunda kalılabilmekte ve bu nedenle de eğitime ve/veya kurum-kuruluşa mali yük getirmektedir.

Bu çalışmada MS Office PowerPoint isimli ofis paket programının 2007 ve daha ileri sürümleri kullanılarak hazırlanmış sunum dosyalarının basit bir şekilde, ek olarak herhangi bir yazılım bilgisine sahip olma ihtiyacı olmaksızın SCORM uyumlu HTML5 (Hyper Text Markup Language – Yüksek Seviyeli Metin İşaretleme Dili) kodlama kullanan öğretim paketlerine dönüştürülmesinde kullanılabilecek, .NET Framework tabanlı bir yazılım geliştirilmesi hedeflenmiştir. Geliştirilen yazılımda ayrıca JavaScript ve CSS3 (Cascading Style Sheets - Basamaklı Stil Sayfaları) teknolojilerinden de yararlanılacaktır.

1.3. Literatür Araştırması

Karl ANDERSSON ve Dan JOHANSSON HTML5 ve CSS3 teknolojilerini de kullanarak kamu hizmetleri için katmanlı bir Mobil e-Servis uygulaması önerisinde bulunmuşlardır [3]. Masahiro ARAKI ve Daisuke TAKEGOSHI, “A Rapid Development Framework for Multilingual Spoken

Dialogue Systems” isimli makalelerinde sistemi geliřtirmek iin HTML5 teknolojisinden yararlanmıřlardır [4].

Michael NEBELING ve arkadařları “Adaptive Layout Template for Effective Web Content Presentation in Large-Screen Contexts” isimli alıřmalarında geniř ekranlarda web ieriklerinin yayınlaması iin HTML5 ve CSS3 teknolojilerini ve gnmz yaygın tekniklerini de kullanarak bir řablon geliřtirmiřtir [5]. Michael NEBELING ve arkadařları geliřtirdikleri řablonda HTML5 ve CSS3 teknolojilerin getirdiėi yeniliklerle esnek bir yapı ortaya koymuřlardır [5].

Nelson BALOIAN ve arkadařları gerekleřtirdikleri “An Architecture for Developing Distributed Collaborative Applications Using HTML5” isimli alıřmada HTML5 teknolojisinin geliřmiř zellikleri olan WebDatabase, WebSockets, Canvas, LocalStorage ve LocalFileSystem gibi unsurları kullanarak daėıtık iřbirliėi uygulamaları geliřtirmek iin mimari sunmuřlardır [6]. Nelson BALOIAN ve arkadařları geliřtirdikleri mimaride ayrıca uygulamanın geliřtirme ařamasında JavaScript frameworkleri de kullanmıřlardır. Bu sayede veri deėiřimi iin JSON (JavaScript Object Notation – JavaScript Nesne Notasyonu) nesneleri gibi veri paketleri kullanabilmiřlerdir.

Hiroki MORI ve arkadařları 2013 yilındı yaptıkları alıřmada ėretim alanında kullanılmak zere, HTML5 teknolojisi kullanarak video daėıtım sistemi geliřtirmiřlerdir [7]. Bu sayede geliřtirilmiř olan sistem platform baėımsız hale gelmiřtir. zellikle HTML5 video elemanının kullanılmasıyla videoların mobil cihazlarda oynatılması, eklenti ihtiyacının ortadan kalkması nedeniyle daha kolay hale gelmiřtir [7].

Guolei ZHU ve arkadaşları 2012 yılında yaptıkları çalışmada HTML5 işaretleme dilinin yeni etiketlerinden biri olan “canvas” elemanını, yine HTML5 teknolojisiyle birlikte gelen yeni özelliklerden biri olan WebSocket ve WebWorker özelliklerini birlikte kullanarak gerçek zamanlı medya oynatıcı tasarlamışlardır [8]. Aynı çalışmada H.264 formatlı video bilgisini JavaScript kod çözücü kullanarak işlemişlerdir [8].

Fabien Cazenave ve arkadaşları 2011 yılında JavaScript kullanarak HTML5 ve CSS3 ile oluşturulmuş dokümanların, içeriğinden bağımsız olarak SMIL (Synchronized Multimedia Integration Language - Senkronizeleştirilmiş Multimedya Birleştirme Dili) ile zaman çizelgesine bağlı olarak yürütülmesi üzerine bir çalışma gerçekleştirmiştir [9]. Gerçekleştirilen çalışmada XML (Extensible Markup Language – Genişletilebilen İşaretleme Dili) tabanlı olan SMIL dokümanların zaman çizelgesine yerleştirilmesini sağlamış. JavaScript ise HTML5 ve CSS3 teknolojileriyle oluşturulmuş içerik ile SMIL arasında iletişim için kullanılmıştır [9].

Martin STRÍBNÝ ve Pavel SMUTNÝ 2013 yılında yayınladıkları çalışmada endüstri, okul ve laboratuvarlarda hızlı uygulama geliştirme yazılımı olan ControlWeb yazılımını görselleştirmek amacıyla HTML5 ve JavaScript kütüphaneleri kullanarak web ara yüzü geliştirmişlerdir [10].

Hwa-Young JEONG ve Bong-Hwa HONG gerçekleştirdikleri çalışmada web tabanlı öğrenme ortamında CAT (Common Admission Test – Ortak Kabul Testi) sistemi ve SCORM alt yapısını beraber kullanarak öğrenci odaklı bir sistem ortaya koymuşlardır [11].

A. GARCIA ve A. ESTEBAN 2011 yılında yaptıkları çalışmada mobil cihazlarda SCORM uyumlu uzaktan öğretim materyallerinin ÖYS sistemlerinde olduğu gibi gösterimi için bir uygulama geliştirmişlerdir [12].

Dragoş CAZACU ve arkadaşları 2012 yılında yaptıkları çalışmada bir ÖYS sisteminin mobil cihazlarla uyumlu olabilmesi için gerekli olan temel noktaları ve SCORM gibi standartları destekleyebilmesi için uyulması gereken kuralları analiz etmişlerdir. Tasarlanan bu mÖYS sisteminde uzaktan öğretim materyaline mobil cihazlardaki etkinliği arttırmak için farklı yönlerde eklemeler yapmışlardır [13].

Bartosz SAWICKI ve Bartosz CHABER 2013 yılında yayınladıkları çalışmada 3D modellerin web tarayıcılar üzerinde verimli ve uygun bir şekilde görüntülemek için sadece JavaScript ve HTML5 teknolojilerinden yararlanarak bir yazılım modülü geliştirmişlerdir. Gerçekleştirdikleri çalışmada sunucu taraflı gösterim ile aşamalı ızgara akışı yaklaşımları konu bağlamında karşılaştırılmıştır [14].

2. MATERYAL, METOT VE YÖNTEM

2.1. Uzaktan Öğretim Sistemlerinin Çalışması

Uzaktan öğretimde kullanılan yöntemler teknolojinin gelişimine paralel olarak zaman içerisinde değişmiştir. Kullanılan yöntemler arasında mektupla iletişim, telekonferans, bilgisayarlı öğretim, çevrim içi öğretim, kitlesel açık çevrimiçi kurslar sayılabilir. Kullanılan her bir yöntemin elbette dezavantaj ve avantajları mevcut olabilir. Mektupla öğretimde zaman kaybı, telekonferans sisteminde etkileşim eksikliği, bilgisayarlı öğretimde yetersiz bilgisayar okuryazarlığı, çevrim içi öğretim kullanımında yeteri kadar çevrim içi olma imkânı bulunamaması ihtimali, kitlesel açık çevrim içi kurslarda ise rehber görevi gören eğitici personelin azlığı ve/veya yetersizliği gibi dezavantajların olduğu söylenebilir. Avantaj olarak bakıldığında mektupla öğretimde maliyet düşüklüğü, telekonferans yönteminde görsel olarak zengin içerik sunulabilmesi olanağı, bilgisayar sistemlerinin kullanıldığı yöntemlerde farklı medya türlerinin kullanılabilmesi, çevrim içi öğretim yöntemlerinde farklı sosyal ortamların öğretim faaliyeti içerisinde kullanılabilmesi gibi avantajlar göze çarpmaktadır.

Tarih boyunca yürütülen uzaktan öğretim faaliyetlerini iki şekilde gruplayabiliriz. Bu gruplardan biri faaliyet içerisinde kullanılan alt yapı ve teknolojik gelişmelere bağlı kalınan iletişim türü diğer ise söz konusu uzaktan öğretim faaliyetinin işleniş şekli olarak söylenebilir.

2.1.1. Kullanılan İletişim Teknolojisi

Tarihte her zaman için öğretim ortamları gerek formal gerekse informal olmak üzere her zaman var olmuştur. Bu öğretim ortamlarında süregelen öğretim faaliyetleri neticesinde bilginin sentezi ve sonucunda da yeni bilgi dağarcıklarının oluşumu gerçekleşmiştir. Zaman geçtikçe toplumların gerek kültürel gerekse siyasi açıdan bilgiye verdikleri önemin artması nedeniyle hazırlanan öğretim ortamlarında artış gözlemlenmesi normaldir. Bu gereksinimi en verimli şekilde karşılamak içinse toplumlar sürekli öğretimde teknoloji kullanımını arttırmaya çalışmışlardır. Bunun sonucu öğretim ortamları sürekli gelişerek günümüzde ki halini almıştır.

2.1.1.1. Mektupla Öğretim

Toplumların öğretim ihtiyaçlarını gidermek için geliştirdikleri ortamlara, teknolojinin ilk etkileri uzaktan öğretim faaliyeti açısından 18. yüzyılda gerçekleşmiştir. Bu yönetime öncülük eden Calep PHILIPS isimli stenografi öğretmenin 1728'de Boston gazetesine vermiş olduğu ilan mektupla öğretim döneminin ilk adımı sayılabilir. Bu öğretimde öğrencilere mektupla haftalık derslerin ve alıştırmaların gönderilmesi ve bunun sonucunda öğrenci bireylerin stenografi öğrenebilmesi planlanmıştır. Bu yöntemin kullanımı 19. yüzyılda İngiltere'de de Isaac PITMAN tarafından yine stenografi dersi verilmesi yoluyla devam etmiştir.

Mektupla öğretimde kurumsal olarak en büyük adımsa Londra Üniversitesinin 1858 yılında “harici sistem” adıyla başlattığı uzaktan öğretim faaliyeti sayılmaktadır. Mektupla uzaktan öğretim veren kuruluşların çoğunluğu yirminci yüzyılın ortalarında kurulmuştur. Bunların çoğu ticari atılımlar olsa da aralarında birkaç kâr amacı gütmeyen organizasyon ve dernek de mevcuttur. Bu kuruluşların hepsi olmasa da birkaçı gerçekten güzel öğretim faaliyetleri gerçekleştirmiştir.

Mektupla uzaktan öğretim faaliyetleri yürüten dünyaca tanınmış pek çok kuruluş gerek kurs hazırlığı gerekse pazar arayışı gibi sebeplerle işbirliği yapmışlardır. Özellikle dil problemleri Avrupa’da var olan kuruluşların işbirliğini sınırlandırmıştır. Bugün de var olan Avrupa Uzaktan Eğitim Birliği (EADL) kuruluşu Avrupa’da süre gelen bu işbirliği ürünü olarak ortaya çıkmıştır.

2.1.1.2. Telekonferansla Uzaktan Öğretim

Uzaktan öğretim faaliyetlerinde televizyonun da rol almasıyla teknoloji tabanlı uzaktan öğretim faaliyetlerinin ilk adımı atılmış oldu. Bu sayede öğrencilere, sadece yazılı materyaller değil ayrıca ses ve görüş içerikli materyaller de kullanılarak kurslar verilmeye başlandı.

Öğretim amaçlı ilk film 1910 yılında ortaya çıkmıştır. Thomas Edison filmin icadının ardından öğretim sisteminin ileriki on yıl içerisinde büyük ölçüde değişeceği görüşünü ifade etmiştir. Öğretim aracı olarak televizyonun var oluşu öğretimde çok kanallı medya kullanımı için teknolojiyenin yararlanılması adına önemli bir başlangıç olmuştur. Ancak potansiyeli

hakkında birçok fikir sür lm   olsa da televizyonun   retime ortamlarına etkisi sınırlı olmu tur. Amerika Birle ik Devletleri'nde y ksek  retim  zerine  alı malar yapan Carnegie komisyonunu 1972 yılında yayınladı ı raporda 2000 yılında yerle ke dı ı   retimin y zde 80, yerle ke i i   retimin de y zde 10-20 arası bir oranın telekom nikasyon teknolojileri  zerine kurulaca ı tahminine yer vermi tir.

2.1.1.3. Bilgisayar ve WWW Destekli Uzaktan   retim

Daha  nceki uzaktan   retim faaliyetlerinin materyal da ıtımı yazılı materyaller veya zahmeti  ok olan filmler  eklinde yapılmı tır. Bilgisayarın   retim faaliyetlerinde kullanılmaya ba lanmasıyla   retim materyalleri dijitalle mi tir. Kitap, mektup, slayt gibi basılı materyallerin dijital ortama aktarılması CD (Compact Disk – Kompakt Disk), disket gibi ortamlarla da ıtımını sa lamı tır. Ancak ne var ki mektupla   retim ve telekonferansla   retim y ntemlerinde oldu u gibi bilgisayarın   retimde ilk kullanım adımlarında da   retim faaliyeti sadece   renici ve   retici arasında   retici g d ml  yapılmaktadır. Bu durum   retim faaliyetlerinin kimi zaman gerekli verimlili e ula amamasına neden olmu tur. Bilgisayar teknolojilerinin geli mesiyle   retim faaliyetleri i in gerekli olan sosyal bazı gereksinimlerin giderilmesi, sistem takibi ve istatistik  bilgilerin toplanması gibi   retim faaliyeti i in  nemli i lemleri yerine getirmek amacıyla “  retim Y netim Sistemi” isimli yazılımlar geli tirilmi tir.

Böylece uzaktan öğretim faaliyetlerinin öğrencisi konumunda olan bireyler bu yazılımlar sayesinde öğrenim için farklı medya kaynaklarını, sosyal ortamları ve diğer öğrencilerle işbirliği gibi deneyimleri edinebilmiştir. Bu tür yazılımlar sayesinde öğrenci sınıf ortamını oluşturulan sanal sınıflar ile deneyimleyeceklerdir. Bir öğrenci birey diğer öğrencilerle yazılı ve sesli sohbet ortamlarıyla işbirlikçi öğrenme faaliyetleri gerçekleştirebilmektedir. Birçok ders materyalline rahatlıkla ulaşabilmekte ve fizikî ortamlardan bağımsız olarak çalışmasını yapabilmektedir.

Elektronik teknolojilerin bu denli gelişimiyle öğretim dünyasında öğretim ve öğretim adına temel değişikliklere yol açmıştır [15]. Bu değişikliklere adapte olmak için öğretici birey ve öğretim faaliyetini gerçekleştiren kuruluşlar kendilerini revize etmeye başlamışlardır. Öğretim Yönetim Sistemlerinin yaygın kullanımına başlanmasıyla üniversiteler sanal kampüsler kurmaya başlamışlardır. Öğretici konumundaki bireyler fizikî öğretim ortamlarında kullandıkları materyalleri dijitalleştirmişlerdir. Dijital ortamın bu kadar yoğun şekilde kullanılması elbette bu alanda öğreticinin de ayrı bir okuryazarlık seviyesine erişmesini gerektirmiştir.

Her öğretici birey, kendisini geliştirmeye çalışmış olsa da dijital ortama diğer meslektaşları seviyesinde adaptasyon sağlayamaması olası bir durumdur. Dijital ortamın özellikle uzaktan öğretimin içerisinde vazgeçilmez bir şekilde bu derece kullanılır olması ufak ölçekli de olsa bir sektör halini almıştır. Uzaktan öğretim faaliyetlerinin sürdürülebilir olması için gerekli olan neredeyse her aşama için ticari bir girişimle karşılaşmak mümkündür. Bu aşamalar için ayrıca kimi için başarılı olarak sayabileceğimiz kimi zaman da yeterince

başarılı sayamayacağımız genel kullanım lisansına sahip yazılımlar da mevcuttur.

Dijital ortamın yoğun kullanılmasının etkisi sadece bireyler üzerinde olmayıp aynı zamanda kurumların da gerek öğretim teknikleri gerekse öğretim alt yapıları üzerinde etkide bulunmuştur. Dünya genelinde birçok eğitim kurumu tarafından internet alt yapısı da kullanılarak web üzerinden sanal kampüs ortamları oluşturulmuştur.

2.1.2. Zamanlama Çeşidi

Uzaktan öğretim faaliyetlerinin ilk ortaya çıkışından günümüze kadar gerçekleştirilmesinde teknolojiden bağımsız olduğu düşünülemez. Öyle ki teknolojik gelişmeler uzaktan öğretim faaliyetlerinin işlenişi üzerinde her zaman etkili olmuştur. Uzaktan öğretim faaliyetlerinin mektupla öğretim döneminin ardından telekomünikasyon teknolojisinin kullanılması sayesinde faaliyetlerin işlenişi adına yeni bir boyut kazandırılmıştır. Telekomünikasyonun ardından bilgisayar ve sonrasında da internet alt yapısının gelişmesiyle uzaktan öğretim faaliyetleri zenginlik kazanmıştır. Teknolojik gelişmelerin uzaktan öğretim faaliyetlerine etki ettiği yönlerden bir tanesi de şüphesiz öğretimin sürekliliği içerisinde ayarlanan zamanlamadır. Mektupla öğretim döneminde teknolojik kısıtlılıklar nedeniyle öğretim muhakkak eş zamansız olarak gerçekleştirilmiştir. Telekomünikasyon teknolojilerinin kullanımıyla birlikte uzaktan öğretim içerisinde eşzamanlı olarak faaliyet gerçekleştirme

imkânı doğmuştur. Bilgisayarın kullanımıyla uzaktan öğretim faaliyetlerinin geneli eş zamansız olarak sürdürülmüştür. İnternet teknolojisinin gücünün uzaktan öğretim faaliyetlerinde kullanılmaya başlanmasıyla öğretici kurumlar söz konusu öğretim faaliyetlerini eş zamanlı, eş zamansız veya her iki şekilde birden sürdürme konusunda esnek olma imkânı yakalamışlardır.

2.1.2.1. Eş Zamanlı Öğretim

Öğretici ve öğrencinin eş zamanlı olarak farklı mekânlarda bulunduğu uzaktan öğretim faaliyetlerine eş zamanlı uzaktan öğretim faaliyeti denilmektedir. Bu tip öğretim faaliyetlerinde tekil bir birey ve/veya grubun bilgisayar veya farklı bir teknoloji kullanarak bağlandığı, ses ve görüntünün iletiminin çift taraflı olarak gerçekleştiği video konferans sistemi üzerinden öğretim gerçekleşmektedir.

Günümüzde internet temelli uzaktan öğretim faaliyetlerinin artmasıyla eş zamanlı uzaktan öğretim faaliyetleri için daha rahat bir ortam sağlanmıştır. Aynı zaman da gelişen teknolojiyle birlikte eş zamanlı uzaktan öğretim faaliyetleri sadece ses ve görüntü aktarımının çok önüne geçmiştir. Eş zamanlı öğretim faaliyetlerinde kullanılan yeni teknolojilerle öğretim sırasında uygulama paylaşımı, kaynak paylaşımı, uygulama gösterimi, beyaz tahta paylaşımı, yine farklı bir mekânda bulunan konuyla ilgili bir kişinin davet edilmesi gibi imkânlar kullanılabilir hâle gelmiştir. Katılımda bulunan öğrenciler her hangi bir soru sorma ihtiyacı duyduklarında o anki dersin tamamına hâkim olduklarından daha net bir cevap alabileceklerdir. Aynı zamandan o an

yürütölmekte olan eş zamanlı öđretim faaliyetlerinde katılımcı olan diđer bireylerin de konuyla ilgili görüřlerine anlık olarak ulaşılabilirlerdir. Öđretim için gerekli olan motivasyon faktörünü de eş zamanlı öđretim faaliyetlerinde öđrenici bireyler daha rahat elde edebilmektedirler.

Bunların yanında uzaktan öđretim faaliyetlerinde eş zamanlı öđretim kullanılması öđrenici bireyler için kimi zaman, zaman eşleşmesinin mecburi olması nedeniyle dezavantaj olarak ortaya çıkmaktadır. Günümüzde öđretim yönetim sistemleri ve beraberinde kullanılan yazılımlarla bu dezavantajın önüne geçebilecek birkaç çözüm kullanılabilirlerdir.

2.1.2.2. Eş Zamansız Öđretim

Öđretici ve öđrenen birey ve/veya bireylerin farklı zaman ve mekânlarda öđretim faaliyetlerine katılımının gerçekleştiđi uzaktan öđretim faaliyetleridir. Bu tür öđretim faaliyetlerinde öđretici bireyin daha çok öđretim ortamında rehberlik etmesi söz konusudur. Öđretimin sürekliliđi ve motivasyonun devamı öđrenici bireyler tarafından sağlanır. Eş zamansız öđretim faaliyetlerinde öđrenici ve faaliyetin başarısının izlenmesi için daha geniş çaplı istatistiksel verilere ihtiyaç duyulmaktadır. Hissedilen bu ihtiyacı karşılamak için günümüzde gelişmiş öđretim yönetim sistemleri kullanılmaktadır. Eş zamansız öđretimlerde de eş zamanlı öđretimlerde olduđu gibi kaynak paylaşımı, öđrenciler arasında işbirliđi, takımlarla çalışma, konunun uzmanından bilgi edinme, forumlar üzerinden öđretici-öđrenici

ve/veya öğrenici-öğrenici arasında fikir paylaşımları gibi etkinlikler gerçekleştirilebilmektedir.

Eş zamansız öğretimlerin öğrenci açısından göze çarpan en büyük avantajı bireyin öğretim faaliyeti için yapması gereken tüm etkinliklerin zamanlamasını ve miktarını kendisinin planlamasıdır. Bu avantaj özellikle iş dünyasında aktif olarak rol alan bireylerin öğretim faaliyetlerinden geri kalmamaları için büyük bir fırsat oluşturmaktadır. Diğer taraftan etkinliklerin zamanlamasının ve miktarının planlamasını bireyin kendisinin yapması, bireye kendisini ve amacını çok iyi şekilde tanıması sorumluluğunu eş zamanlı öğretim faaliyetlerinde olduğundan daha çok yüklemektedir.

2.2. Öğretim Alt Yapısı

Uzaktan öğretim faaliyetlerinde kullanılan teknolojiler elbette önem arz etmektedir. Zira bu teknolojilerin kalitesi ve kullanım etkinliği uzaktan öğretim faaliyetinin kalitesini ve geleceğini belirlemektedir. Bunun yanında uzaktan öğretim faaliyetinin kalitesini ve geleceğini etkileyen diğer bir etkense faaliyetin öğretim alt yapısının uygun şekilde tasarlanmasıdır. Burada sözü edilen öğretim faaliyeti için tasarlanmış ders materyallerinin ve sosyal ortamı yansıtmak için kullanılan imkânların kalitesidir. Uzaktan öğretim faaliyetinin alt yapısı denilince birçok değişken bu konu içerisinde değerlendirilebilir, kullanılan teknolojik imkânlar, eğitici personelin teknoloji kullanım yeterliliği, öğrencilerin bağlantısı için gereksinim duyulan bant genişliği miktarının

düşünülmesi, sanal materyallerin dağıtım yöntemlerinin ayarlanması bu değişkenler arasında sayılabilir.

Bu çalışmanın amacının uzaktan öğretim faaliyeti için materyal geliştirme konusu üzerinde olması nedeniyle bu başlık altında öğretim alt yapısının materyal geliştirmeyeyle ilgili kısmı söz konusu olacaktır.

2.2.1. İçerik Referans Modelleri

Etkin bir uzaktan öğretim faaliyet sunabilmek için yayınlanan içerik materyalleri üzerinde günümüze kadar farklı çalışmalar yürütülmüştür. Genel olarak bir ÖYS içerisinde öğrencinin faaliyetlerinin mümkün olduğunca detaylı şekilde takip edilmesi uzaktan öğretim faaliyetinin kalitesi açısından önem taşımaktadır. Öğrencinin ders materyalini kullanırken gerçekleştirdiği kullanım hareketlerinin tutulabilmesi de bu noktada önem taşır. ÖYS içerisinde bu kabiliyetin sağlanması için hazırlanan dijital ders materyallerinin akış şeması üzerinde standartlaştırma faaliyetleri yürütülmüştür. Bu çalışmalar kapsamında günümüze kadar çeşitli içerik referans modelleri tasarlanmış ve kullanılmıştır. Bu noktada şunu belirtmek gerekir ki bir ÖYS mutlaka bir içerik referans modelini desteklemek zorunda değildir, ancak ÖYS'nin sağladığı bu öğretim alt yapısı sayesinde uzaktan öğretim faaliyetinin etkinliğini ölçme ve artırma çalışmaları yapılabilmesi için bilgiler edinilebilmektedir.

Uzaktan öğretim faaliyetlerinde öğretim içeriklerinin akışlarını standartlaştırmak için kullanılan içerik referans modelleri AICC (Aviation

Industry CBT (Computer-Based Training) Committee), SCORM 1.1, SCORM 1.2, SCORM 2004 sürümleri ve son olarak da Tin Can API şeklinde sayılabilir.

Şekil 2.1’de örnek SCORM manifest dosyası içeriği mevcuttur.



```
- <content>
  - <globalProperties>
    - <externalMetadata>
      <source>ADL</source>
      <model>ADL SCORM 1.1</model>
      <location>metadata(1.1).xml</location>
    </externalMetadata>
  </globalProperties>
  - <block id="B1">
    - <identification>
      <title>TITLE</title>
      <description>DESCRIPTION</description>
    - <labels>
      <curricular>COURSE</curricular>
    </labels>
    </identification>
    - <sco id="S1">
      - <identification>
        <title>TITLE</title>
        <description>DESCRIPTION</description>
      - <labels>
        <curricular>SCO</curricular>
      </labels>
      </identification>
      - <launch>
        <location>index.html</location>
      </launch>
    </sco>
  </block>
</content>
```

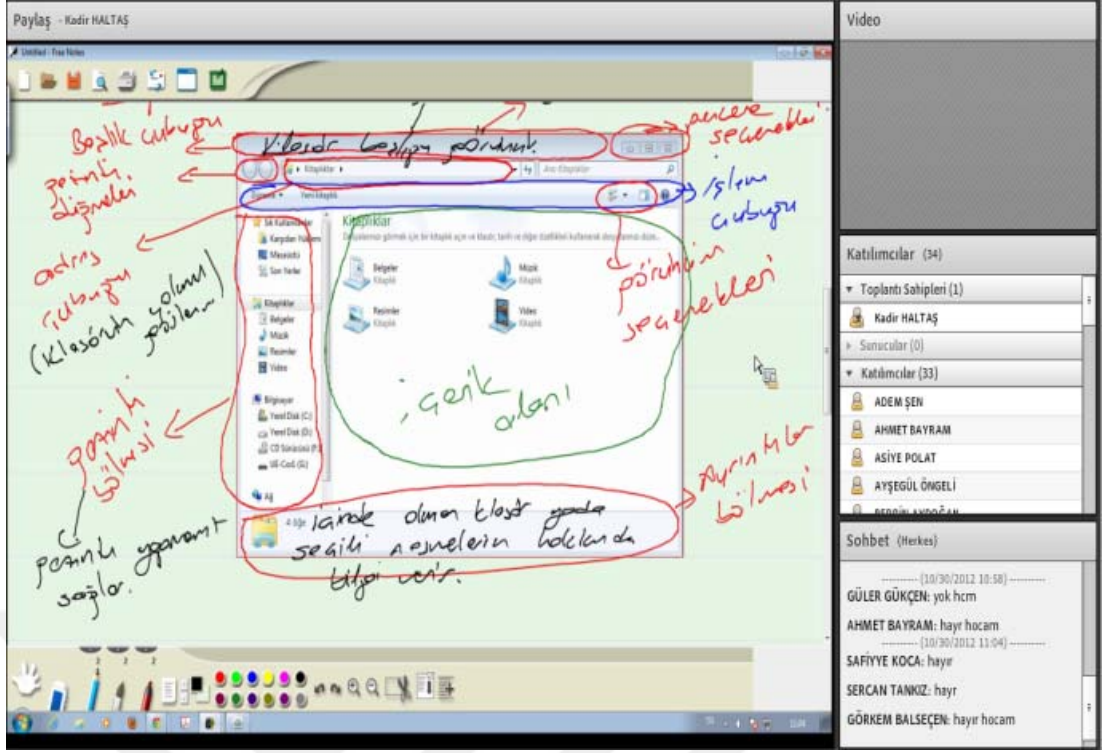
Şekil 2.1. Örnek SCORM paketi manifesto dosyası [16].

2.2.2. Sanal Sınıf ve Diğer Sosyal Ortamlar

Uzaktan öğretim faaliyetleri geçmişinde eş zamansız faaliyetler olarak başlasa da teknolojinin gelişimiyle birlikte bugün eş zamanlı öğretim

faaliyetleri gerekleřtirme imkânına sahip olunmuřtur. Gnmzde ister eř zamanlı ister eř zamansız uzaktan ğretim faaliyetlerinde eř zamanlı ğretim araları kullanılmaktadır. Bu eř zamanlı ğretim aracı sanal sınıflar olarak karřımıza ıkmaktadır. Temel olarak bir sanal sınıfın amacı farklı ortamlarda bulunan ğretici ve ğrencilerin eř zamanlı olarak ađ teknolojileri alt yapısının kullanılmasıyla tıpkı fiziki sınıf ortamından olduđu řekliyle internet zerinde buluşmalarını sađlamaktır.

Gnmzde pek ok sanal sınıf aracı mevcuttur. Bu sanal sınıf araları yazılım olarak elde edilebilmektedir ve uzaktan ğretim faaliyetinde bir YS kullanılıyorsa sistemle btnleřtirilmelidirler. Genel olarak sanal sınıf uygulamaları alıřma prensipleri nedeniyle YS sunucularından ayrı sunucu kurulumları gerektirmektedir. Bir sanal sınıfın standart olarak ğretici ve ğrencileri evrim ii olarak bir araya getirmesi beklenir. Ancak her yazılım bu standart zelliđin yanında farklı zellikler de tařımaktadır. Bu zelliklerden bazıları grnt paylařımı, ekran paylařımı, sesli soru sorma ve yanıtlama imkânı, fiziksel sınıf ortamında bulunan tahta aracının yerini tutabilecek beyaz tahta uygulaması, sanal sınıf zerinden uygulama paylařımı ve gsterimi, misafir ğretici ve/veya ğrencinin sanal sınıf oturumuna davet edilebilmesi, gerekleřen sanal sınıf oturumunun tm etkinlikleriyle kayıt altına alınabilmesi řeklinde sıralanabilir.



Şekil 2.2. Örnek sanal sınıf oturumu

Şekil 2.2’de örnek bir sanal sınıf oturum örneği görülmektedir.

Uzaktan öğretim faaliyetlerinde yer alan sosyal ortamlar sadece sanal sınıflardan oluşmamaktadır. Kullanılan ÖYS alt yapısına göre değişmekle birlikte genel olarak kullanılan sosyal ortamlar; öğretici – öğrenci veya öğrenici – öğrenici arası mesajlaşma, çalışma grupları oluşturma ve takibi, forum oluşturma, tartışma grupları oluşturma imkânları şeklinde örneklenebilir.

2.2.3. Kullanılan Altyapı ve Teknolojiler

Uzaktan öğretim faaliyetleri, dijital teknolojinin yoğun şekilde kullanılmaya başlanmasıyla daha da karmaşık bir yapıya sahip olmuştur. Bilgisayar destekli bir uzaktan öğretim faaliyeti eş zamanlı veya eş zamansız nasıl yürütülürse yürütülsün faaliyet yapısının, ders içeriklerinin ve kaynak materyallerin hazırlanması, görsel anlatım için gerekli animasyonları hazırlanışı, öğretim içeriğiyle ilgili soru ve soru bankalarının hazırlanışı, öğretici ve öğrenci için gerekli teknolojik alt yapının hazırlanması ve bakımı gibi farklı bölümlerinde farklı teknolojilerin kullanılması gerekmektedir. Alt yapının hazırlanması için gerekli olan bu farklı teknolojilerin kullanımı için geniş bilgi birikimine ve aynı ölçüde uygun zaman dilimine ihtiyaç duyulmaktadır.

Bu bölümde internet temelli bilgisayar destekli uzaktan öğretim faaliyetlerinde kullanılmakta olan başlıca alt yapı teknolojilerinden bahsedilecektir. Söz konusu teknolojilerin her biri için uzman bir personele ihtiyaç duyulmaktadır.

2.2.3.1. Öğretim Yönetim Sistemleri (ÖYS)

ÖYS'nin temel amacı farklı ortam ve/veya zamanda bulunan öğrenci ve öğreticinin uzaktan öğretim süreci boyunca gerçekleştirdiği faaliyetlerin idare, takip ve düzenleme, kaynak dağıtımı ve yönetsel işlemlerin yapımının gerçekleştirilmesi gibi öğretim için hayati önemi bulunan aktivitelerin takibini gerçekleştirmektir. Bu sistemde öğretim materyalleri olarak

kullanmak istediğimiz her hangi bir nesneyi sistem üzerinde kullanmak için uygun formata dönüştürülmesi gerekmektedir. Dünya üzerinde birçok farklı türde ÖYS'ler kullanılmaktadır, bunlardan birkaçını şu şekilde sıralayabiliriz: eFront, Moodle, Sakai. ÖYS'ler arasında hem açık kaynak kodlu (Moodle) hem de ticari amaçlı hazırlanmış (Blackboard) olanlar mevcuttur.

Burada göze çarpan nokta ÖYS üzerinde kullanılan öğretim içeriğinin farklı bir ÖYS'ye aktarıldığında da aynı davranışı ve kullanım olanaklarını sağlayıp sağlamayacağıdır. Bu sorunun giderilmesi için genel anlamda standartlar belirlenmiştir. Bunların başlıca olanları şu şekilde sıralanabilir: IMS (Information Management System – Bilgi Yönetim Sistemi), AICC, SCORM. Bu standartlardan en yaygın olarak kullanılanlardan bir tanesi SCORM (Sharable Content Object Reference Model) olarak bilinir.

Öğretim yönetim sistemlerinin incelenmesi ve karşılaştırılması adına yapılan bazı araştırmalar şu şekildedir: Sabine ve Beate (2005)'in 9 ÖYS; Ilias, openUSS, Lon-capa, dotLRN, ATutor, Dokeos, Spaghetti learning, Sakai, Moodle arasında uyarlanabilirlikleri açısından gerçekleştirdikleri karşılaştırma araştırması sonucunda diğerleri arasından Moodle ÖYS'nin en yüksek dereceyi aldığı gözlemlenmiştir [17]. Al-Ajlan tarafından 2012 yılında 10 adet ÖYS arasında yapılan araştırması sonucunda Moodle ÖYS'nin diğerleri arasında en iyi sonucu almıştır. Yapılan bu çalışmada ÖYS'ler ticari ve açık kaynak kodlu olmak üzere iki grupta ayrılarak incelenmiştir. Al-Ajlan, ÖYS'ler arasında gerçekleştirdiği karşılaştırmaları teknik araçlar, destek araçları ve öğrenci araçları özelliklerini göz önünde bulundurarak gerçekleştirmiştir. Yapılan bu çalışmada Moodle sisteminin sadece SCORM

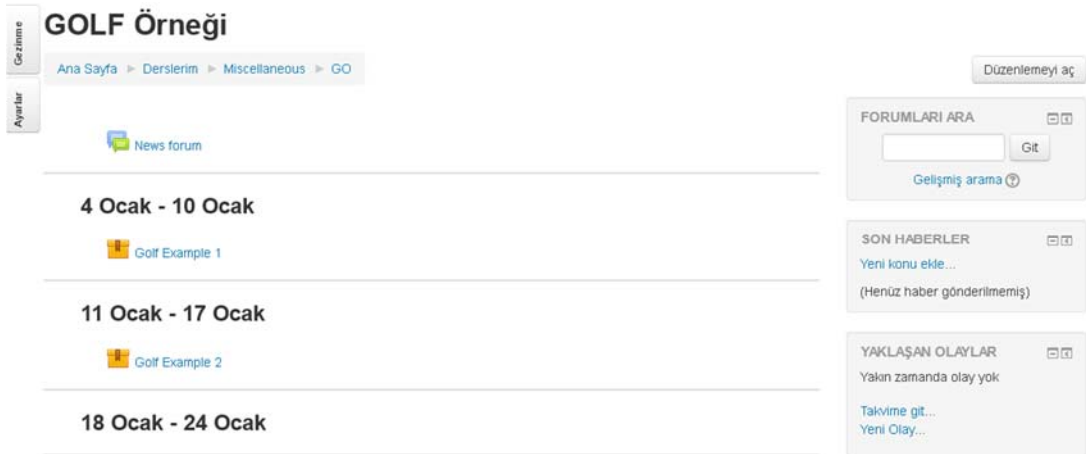
paketlerinin birkaç sürümünü desteklemekte problem yaşadığı gözlemlenmiştir [18].

Uzaktan öğretim faaliyetlerinde ÖYS kullanımıyla, öğretim organizasyonunun yönetimi, öğretim faaliyetlerinin gerçekleşmesi için gerekli sanal ortamın sağlanması, sürdürülen uzaktan öğretim faaliyetinin istatistiksel verilerinin toplanması ve raporlanması gibi işlevler yerine getirilebilmektedir. Günümüzde hızla artan farklı isim ve kabiliyetlerde ÖYS'ler mevcuttur. Bir öğretim yönetim sisteminden beklenen temel işlevler şu şekilde sıralanabilir:

- Kurumsal yapının sanal ortama tam anlamıyla taşınabilmesi için esnek organizasyon yapısının desteklenmesi,
- Direk mesajlaşma, tartışma forumları, duyurular, çalışma grupları, canlı mesajlaşma gibi iletişim araçlarını barındırması,
- Kendisine ait sanal sınıf uygulamasının olması ve/veya popüler sanal sınıf uygulamaları ile bütünleşmiş şekilde çalışabilmesi,
- Sanal ders materyallerinin barındırılmasının ve kullanımının kolay bir şekilde gerçekleştirilebilir olması,
- Kendisine ait bir içerik referans modeli olması ve/veya bilinen en az bir içerik referans modelini destekliyor olması,
- Organizasyon, öğretim faaliyeti, sanal sınıf oturumları ve kullanım verileri gibi öğretim için önem taşıyan verileri saklayabilmesi ve istenildiğinde kolay bir şekilde rapor verebilmesi,

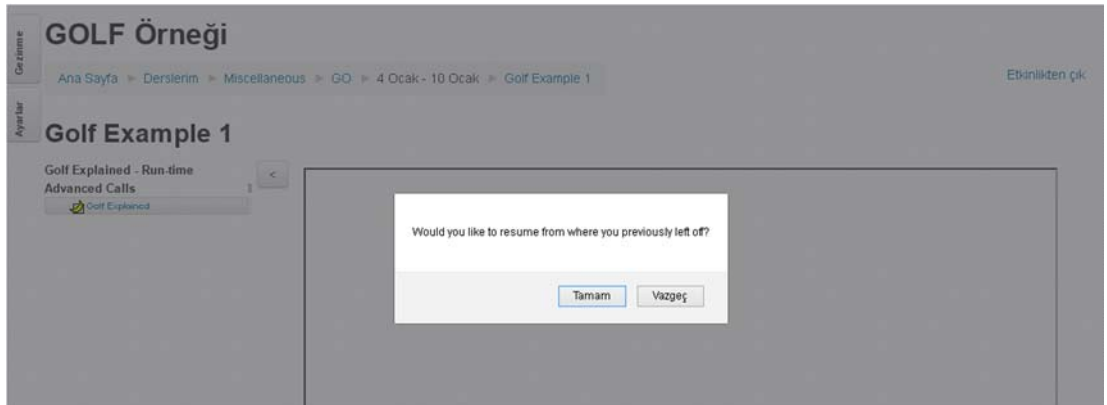
- Kimi zaman öğretim faaliyeti için gerekli olan üçüncü parti yazılımları desteklemesi ve/veya kolayca bütünleşebilir olabilmesi.

Sözü geçen bu özelliklerden üzerinde en çok durulması gereken işlevler, içerik referans modeli desteklemesi ve sanal sınıf yazılımlarıyla bütünleşebilir olmasıdır. Her ÖYS'nin alt yapı olarak sanal sınıf desteği değişiklik gösterse de genel olarak ÖYS'ler tarafından sanal sınıf uygulamaları desteklenmektedir. Gerek eş zamanlı öğretim gerekse eş zamansız öğretim faaliyetlerinde sanal sınıf uygulamaları kullanımı sayesinde gerçekleşen oturumlar kayıt altına alınabilmektedir. Bu sayede öğrenciler katılamadıkları sanal sınıf oturumlarını tekrarlayabilmektedir. Uzaktan öğretim yönteminin en önemli artılarından biri de öğreticiye bağlı kalmadan gerçekleşebilen sınırsız tekrar imkânıdır. Şekil 2.3'de örnek bir ÖYS ekran görüntüsü görülmektedir.



Şekil 2.3. Moodle ÖYS web ara yüzünden örnek bir kesit.

Öğretim yönetim sistemlerinde içerik referans modellerinin desteklenmesi büyük önem taşımaktadır. Bu sayede öğrencinin ders materyali kullanımı sırasında tüm hareketleri içerik referans modelinin desteklediği derecede takip edilebilmektedir. Aynı zamanda içerik referans modellerinin desteklenmesi sayesinde daha önceden hazırlanmış içerik referans modeli destekli öğretim materyalinin kullanımı rahatlıkla gerçekleştirilebilmektedir. Bu konuda en sık kullanılan içerik referans modeli SCORM olarak bilinmektedir. ÖYS'ler içerisinde içerik referans modellerinin kullanımıyla birlikte öğrencinin bireysel olarak takibi daha rahatlıkla gerçekleştirilebilmektedir. Ayrıca diğer bir artısı ise ders materyallerinin kullanımı tam olarak gerçekleşmediği durumlarda öğrenci kaldığı yerden devam edebilmekte böylece zaman kaybı ve tekrar nedeniyle oluşan motivasyon kaybının önüne geçilebilmektedir. Şekil 2.4'de Moodle ÖYS'de dersin kaldığı yerden devam edilmesine ilişkin görsel mevcuttur.



Şekil 2.4. Moodle ÖYS'de yarım kalan dersin bırakılan yerden devam edilmesine ait ekran görüntüsü

2.2.3.2. Flash

Günümüzde her ne kadar Adobe firması tarafından geliştirilmesi ve dağıtımı gerçekleşse de Flash temelinde Macromedia firması tarafından 1996 yılında piyasaya sürülmüştür. Bugün Flash uygulaması kullanılarak web ortamında kullanılmak için etkileşimli animasyonlar, iki yönlü video aktarımı, öğretim materyali hazırlama işlemleri yapılmaktadır.

Aslında Flash uygulamasının Macromedia şirketi tarafından yayınlanmadan öncesine ait bir geçmişi de bulunmaktadır. Temelde kalem bilgisayarlar için PenPoint OS işletim sistemi üzerinden çalışan SmartSketch uygulamasına dayanır. PenPoint işletim sisteminin pazarda beklenen başarıyı yakalayamamasının ardından SmartSketch uygulaması Windows ve Mac OS işletim sistemi üzerinde çalıştırılmaya başlanmıştır. Daha sonra uygulamaya kare-kare animasyon oluşturma özelliği eklenmiş ve pazara FutureSplash ismiyle yeniden sürülmüştür. Macromedia şirketinin bu uygulamayı geliştirmesi sonrası “Future” ve “Splash” kelimelerini birleştirerek “Flash” ismi kullanılmaya başlanmıştır.

Flash uygulaması ile oluşturulan dosyaların uzantısı “.swf” olarak karşımıza çıkmaktadır. Bu dosyalar bilgisayar üzerinde tekil olarak çalıştırılabilirler gibi web ortamında bulunan temel kodlama dili HTML kullanılarak web sayfaları içerisine yerleştirilebilir. Flash uygulamasına ait bu dosyaların içerisinde firmanın kendisi tarafından hazırlanmış olan betik dili ActionScript ile kodlama yapılabilmekte ve bu sayede dosya dışındaki ortamlarla da etkileşime girilebilmektedir. Şekil 2.5’de örnek ActionScript kodu görülmektedir.

```
var greet:TextField = new TextField();  
greet.Text="Hello World!";  
this.addChild(greet);
```

Şekil 2.5. Örnek ActionScript kodları

Flash uygulamasının üçüncü parti uygulamalar kullanılarak içeriğinin objeler ve kod bazında görülebilir ve değiştirilebilir olması işletim sistemi ve mobil cihaz üreticileri tarafından güvenilirliğinin yitirilmesine sebep olmuştur. Günümüzde bazı gelişmiş mobil cihaz ve işletim sistemleri Flash uygulamasını desteklememektedir. Bunun sebebi ise üreticiler tarafından Flash uygulamasıyla hazırlanmış dosyaların güvensiz olarak görülmesidir. Zamanla Flash uygulamasının yerini alabilecek alternatiflerin doğması bu platformlar tarafından Flash'ın tamamen kalkmasına neden olmuştur.

2.2.3.3. HTML

Türkçesi Hiper Metin İşaretleme Dili (Hyper Text Markup Language) olan HTML günümüzde sıklıkla kullandığımız, sıradan bir ihtiyaç hâlini almış olan Internet ortamında yayınlanan web sayfalarının temel yapı taşı olarak düşünülebilir. Bizler web ortamında gerek mobil gerekse masaüstü cihazlar ile gezinirken web sayfalarını görüntülemek için web tarayıcıları kullanmaktayız. Web tarayıcıların burada üstlendiği görev HTML kullanılarak yayınlanmış olan

web sayfasının kullanıcının isteđi dođrultusunda, düzgün formatta görüntüleme işlemidir. Bunun anlamı HTML bir programlama dili olarak sayılmamasıdır. Zira HTML kullanılarak kendi başına çalışan bir yazılım geliştirilemez.

İnternet ortamında yayınlanan web sayfalarının temel yapı taşı olan HTML, W3C (World Wide Web Consortium – WWW Konsorsiyumu) organizasyonu tarafından standartlaştırılmaktadır.

HTML kullanılarak farklı medya türlerinde bilgiler barındıran dosyalar hazırlanabilir ve/veya hazırlanan bu dosyalar arasında bağlantı kurulabilir.Şekil 2.6’da HTML sayfasına resim eklemeye ilgi örnek ve çıktısı görölmektedir. Diđer bir kullanım amacı ise web sayfalarının web tarayıcılar tarafından düzgün görüntülenebilmesi için gerekli kuralları tanımlamasıdır.

```

```



Şekil 2.6 Bir HTML sayfasına resim eklemek için kullanılan “” etiketine ve sayfa çıktısına bir örnek.

Avrupa Nükleer Araştırma Merkezinde çalışan Tim BERNERS-LEE’nin HTML kodlamayı geliştirmesi ardından, internet ortamında sayfaların

artmasıyla internet popülerlik kazanmıştır. Buna bağlı olarak web kavramı zamanla gelişmiş ve bugün dilimize yerleşmiştir. İlk adımlarının atılmasından bu zamana kadar HTML zamanla geliştirilerek farklı sürümlerle karşımıza çıkmıştır. Bu sürümler; HTML 1.0, HTML 2.0, HTML 3.2, HTML 4.0, HTML 4.01, XHTML (eXtensible Hyper Text Markup Language – Geliştirilebilir Yüksek Seviyeli Metin İşaretleme Dili) ve son olarak HTML 5.0 şeklinde sayılabilir.



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "  
http://www.w3.org/TR/html4/strict.dtd">  
<html>  
  <head>  
    <title></title>  
  </head>  
  <body>  
  </body>  
</html>
```

Şekil 2.7. HTML 4.01 standardı kullanılarak hazırlanmış HTML sayfa kodlaması.

Şekil 2.7’de örnek bir HTML 4.01 standardındaki sayfasının kodlaması görülmektedir.

HTML 5.0; temel olarak HTML 4.0 sürümü etiketleri üzerine geliştirilmiştir. Genel amaç olarak günümüz web ortamında hızla artan zengin medya formatlarının ek bir bileşen olmadan sayfa içerisine dâhil edilebilir ve yürütülebilir olmasını sağlamak şeklinde söylenebilir. HTML 5.0 sürümünün

sağladığı diğer bir özellikse ister kullanıcı arabirimi geliştiricisi isterse sunucu taraflı uygulama geliştiricisi olsun tek bir kod yapısı kullanılması sayesinde rahat bir geliştirme ortamı sunmasıdır. HTML 5.0, metin biçimleme amaçlı kullanılabildiği gibi web uygulamaları da geliştirilebilir. CSS3 ve JavaScript teknolojilerinin de birleştirilmesiyle kararlı ve birçok platform uyumluluğu sağlanmış uygulamalar geliştirilebilmektedir.

HTML 5.0 sürümüyle birlikte özellikle semantik açıdan geliştirmelerin yapılabilmesi için HTML 4.01 sürümüne ek olarak bazı etiketler eklenmiştir. Eklenen bu etiketler sayesinde görsel yapısı ile mantıksal yapısı bir bütün oluşturmaktadır. HTML 5.0 ile birlikte gelen yeni etiketler Çizelge 2.1'de gösterilmiştir.

Çizelge 2.1. HTML5 ile gelen yeni etiketler

<canvas>	Sayfada bir tuval alanı oluşturur. Tuvale çizim JavaScript ile yapılabilir.
<audio>	Sayfaya ses oynatıcı bir modül ekler.
<video>	Video oynatıcı bir modül ekler.
<progress>	İşlem süreci göstergesi ekler.
<caption>	Başlık olarak düşünülen metinleri düzenler.
<header>	Sitenin başlık ve açıklama içeriğini alır.
<nav>	Menüleri ve bir takım zorunlu işlevleri içine alır.
<footer>	Sitelerin en alt kısmını içine alır.
<section>	Sitelerin ana içerik kısmını içine alır.
<aside>	Ana içerikten ayrı yazılan kısımdır.
<article>	Makale, deneme tarzı yazıları kapsar.
<embed>	Dışarıdan eklenen bileşenler için kullanılır. (Örn: swf uzantılı dosyalar.)
<details>	Detay bilgisi içerir.
<summary>	Yazının başlığını belirler.
<time>	Tarih ve saat verilerini kapsar.

Çizelge 2.1. (Devam)

<mark>	Yazı içerisinde özellikle üstünde durulan kelimeleri belirler.
<figcaption>	<figure> elementinin başlığını belirler.
<figure>	Çeşitli medya içeriği gruplarını belirler.
<hgroup>	Başlık grubunu belirtir. H1, H2 gibi başlık elementleri burada tanımlanabilir.
<datalist>	Düzenlenebilir elementlere otomatik tamamlama özelliği verilmesini sağlar.

Öğretim ortamının sanal olarak öğrenciye ulaştırılması konusu düşünüldüğünde içeriği aktarmak için şüphesiz ki en iyi ortamın web teknolojileri olduğu görülecektir. Web teknolojileri kullanılmak istendiğinde ise HTML kullanımı neredeyse zorunluluk halini almıştır.

Web teknolojileri kullanılarak herhangi bir içeriğin dağıtımı işleminde HTML, temel olarak içeriğin tarayıcı ortamında istenilen düzende görünmesini sağlamaktadır. Öğretim içeriklerinin ses, görüntü ve video gibi birden fazla medya ortamına ihtiyaç duyduğu düşünüldüğünde HTML5 öncesi sürümler yetersiz kalmıştır. Bu açığı gidermek için her web tarayıcı üreticisi kendi tarayıcılarına uyumlu eklentiler geliştirmek zorunda kalmıştır. Ayrıca tüm bunlara alternatif olarak Adobe firmasının ürünü olan Flash nesneleri ve oynatıcısı kullanılmıştır. Tüm bunlar günümüzde hala alternatif imkân olarak kullanılmaktadır. Ancak bu durum hem kullanıcıların kullanım tecrübelerini etkilemekte hem de güvenlik açıklarına neden olmaktadır. Buna bağlı olarak HTML5 temelli içerik ortamlarının artmasıyla birçok web tarayıcı üreticisi ve

hatta mobil işletim sistemi yayımcıları bu ek uygulamaları artık desteklemediklerini veya sorumluluğu kullanıcılara bıraktıklarını açıklamıştır.

HTML5 kullanımının yaygınlaşmasıyla kullanıcı arabirimi tasarımcıları ve sunucu taraflı yazılım geliştiricilerin ortak bir alt yapıda buluşmalarını sağlamıştır. HTML5 beraberinde CSS3 ve JavaScript kullanımıyla çok daha güçlü ve kararlı uygulamalar çıkabilmekte ve sanal ortamda var olabilen hemen hemen her makinede sıkıntısız çalışmaktadır [19].

2.2.3.4. .NET Framework

.NET Framework, CLR (Common Language Runtime - Ortak Dil Çalışma Zamanı) ile birlikte programların ve oluşturulmuş kütüphanelerin çalıştırılabilmesi için gerekli kütüphane kümelerini içerir. Bu kütüphane kümesinde Framework çekirdeğine ait kütüphaneler ve bu kütüphaneler ile oluşturulmuş, .NET Framework çatısı altında kullanılan teknolojilere ait kütüphaneler bulunmaktadır. Uzun adı “Common Language Runtime (Ortak Dil Çalışma Zamanı)” olan CLR sayesinde “managed code (yönetimli kod)” yapısında hazırlanmış yürütülebilir “.exe” veya kütüphane olarak kullanılabilir “.dll” dosyalarına dönüştürülür.

Bilindiği üzere bir yazılımın veya bir kod parçasının bir donanıma anlamlandırılması için işlemci üzerinde işlenmesi gerekmektedir. Dolayısı ile hazırlanmış olan program kodları makine diline çevrilmelidir. Günümüzde her işlemci aynı komut kümesini kullanmadığı için bir kod parçası ya da bir yazılımın kullanılması düşünülen platform için özel olarak derlenmesi

gerekmektedir. Şüphesiz ki her platform için bu işlemi gerçekleştirmek büyük emek ve zaman kaybı olacaktır. Bu durumun üstesinden gelebilmek için .NET Framework yapısı içerisinde hazırlanan yürütülebilir dosya (.exe) veya kütüphane dosyası (.dll) MSIL (Microsoft Intermediate Language – Microsoft Ara Dil) ismi verilen ara bir dile dönüştürülür. MSIL diline dönüştürülen kod bloklarının oluşturduğu dosya içerisinde program komutlarının yanı sıra metadata bilgileri de mevcuttur. Hazırlanan bu paket dosya CLR tarafından yürütülmek üzere yüklendiğinde CLR içerisinde bulunan JIT (Just In Time – Çalışma Anında Derleme) derleyici ile makine diline çevrilir.

Bu yapı sayesinde .NET Framework tarafından desteklenen farklı dillerde (C#, VB , F# vb.) yazılmış yazılımlar ,kütüphaneler ve/veya program parçaları sorunsuzca birlikte çalışabilmektedirler. Aynı zamanda makine diline çevrim işlemi tam çalışma anında yapıldığı için CLR tarafından desteklenen tüm platformlarda uyum içerisinde çalışabilmektedir.

CLR aynı zamanda birçok servisin çalışmasına ev sahipliği yapar. Bu servislerden birkaç tanesi; bellek yönetimi, kütüphane yükleme ve güvenlik hizmetleri olarak sıralanabilir. Günümüze kadar .NET Framework yapısının 1, 2, 3, 3.5, 4 ve 4.5 olmak üzere çeşitli sürümleri geliştirilmiştir.

2.2.3.5. JavaScript

Sadece biçimleme dili olarak kullanılabilen HTML zaman zaman kullanım alanlarına göre basit ölçüde esnekliğe ve daha çok işlevselliğe ihtiyaç

duyabilmektedir. Günümüzde HTML ile hazırlanmış hemen hemen her dosyada bu ihtiyacı karşılamak için JavaScript kullanılmaktadır.

İlk olarak Netscape firması tarafından Netscape Navigator 2.0 web tarayıcı ile internet dünyasına tanıtılmıştır. HTML sayfalarına dinamizm katan bu programlama dili başlangıçta LiveScript olarak isimlendirilmiştir. Netscape 2.0 Beta3 sürüm web tarayıcı ile birlikte ismi JavaScript olarak değiştirilmiştir.

JavaScript dilinin en büyük avantajlarından biri istemci taraflı olarak çalışmasıdır. Yani bir kullanıcı siteyi ziyaret ettiğinde web tarayıcı ile görüntülenen sayfada yer alan JavaScript kodları çalışırken siteyi ziyaret eden kullanıcının sistem kaynaklarını kullanmaktadır. Bu sayede web sitesinin barındırıldığı sunucu bilgisayarın kaynak kullanımında aşırı yüklenme gerçekleşmemiş olur.

JavaScript nesne yönelimli bir betik dilidir. Farklı uygulamalar içinde ve/veya farklı uygulamalarla haberleşmek için kullanılabilmesinin yanı sıra sıklıkla tasarlanan HTML sayfalarına esneklik kazandırmak için kullanılır.

2.3. SCORM (Paylaşılabilir İçerik Nesne Referans Modeli)

SCORM uzaktan öğretim sistemlerinin idaresi için kullanılan ÖYS'ler içine aktarılan içeriklerin standartlarını, özelliklerini ve aktarım için gerekli olan paketleme şeklini tanımlamak için kullanılan yapıdır. SCORM, Amerika Birleşik Devletleri Savunma Sekreterliği Ofisince oluşturulmuş ADL (Advanced Distributed Learning – Gelişmiş Dağıtık Öğrenme) girişimi tarafından Ocak 1999'da ortaya çıkartılmıştır [20].

SCORM'un tarihsel süreci içerisinde SCORM 1.1, SCORM 1.2, SCORM 2004 2.Sürüm, SCORM 2004 3.Sürüm, SCORM 2004 4.Sürüm ve TIN Can API olmak üzere versiyonları bulunmaktadır. Tüm bu sürümlerin temel amacı aynı olmakla birlikte sürümler arasında standartlar açısından geliştirmeler yapılmıştır. Günümüzde en çok kullanılan sürüm SCORM 2004 olarak bilinmektedir.

SCORM standartlarının temel amacı, ÖYS ve ders içerik paketinin nasıl birbiri arasında haberleşmeyi sağlayacağı, bir içerik hazırlayıcı veya programcı personelin görevlerini gerçekleştirirken uyması gereken standartları ve hedefleri belirler. Buradan da anlaşılacağı gibi SCORM öğretim içeriğinin pedagojik veya tasarımsal ara yüzü ile ilgilenmez. Sadece arka planda olan sistem-sistem ve içerik-sistem arası iletişimi organize etmektedir.

Bir uzaktan öğretim faaliyeti içerisinde SCORM destekli materyallerin bulunmasının faydaları şu şekilde sıralanabilir;

- Yeniden kullanılabilirlik: SCORM standartları dikkate alınarak hazırlanan öğretim materyali farklı öğretim faaliyetlerinde, farklı ÖYS'ler içerisinde kullanılabilirler.
- Erişilebilirlik: belirli uzak bir kaynaktan öğretimsel bileşenin birçok farklı noktaya rahatça ulaştırılması durumunu ifade eder.
- Özelleştirilebilir olma: SCORM uyumlu bir ders materyali bireysel ve kurumsal ihtiyaçlara göre özelleştirilebilmektedir.
- Karşılanabilirlik: SCORM uyumlu içerik paketleri belirli standartlara uyumlu olarak hazırlandıkları için farklı ortamlarda da rahatlıkla kullanılabilir. Bu sayede her ders içeriği için harcanan zaman ve maliyet düşürülebilmektedir.

- Dayanıklılık: teknolojik hızlı değişimi ve karşılaşılan gelişmelere rahatlıkla uyum sağlayabilir ve tekrar tasarlanması, ayarlanması veya kaydedilmesine gerek yoktur.
- Birlikte kullanılabilirlik: bir merkezde belirli araçlarla hazırlanan SCORM içerik paketleri barındırdığı standartlar sayesinde farklı ortamlarda farklı platformlarda rahatlıkla kullanılabilir.

Bir SCORM paketinin içeriğini ve hazırlanışını anlatmadan önce birkaç tanımlamanın yapılması faydalı olacaktır.

İçerik, bir uzaktan öğretim faaliyetinde bilginin anlamlı küçük parçalara ayrılmış ve metin, ses, grafik, video, animasyon, benzetim vb. gibi dijitalleştirilmiş şekilde kullanıma hazır hâlidir.

Öğrenme nesneleri, medya ortamında bağımsız şekilde bilgi topluluğundan oluşan tekrar kullanılabilir şekilde hazırlanmış uzaktan öğretim içeriğinin temel yapı taşıdır. Paylaşılabilir öğrenme nesneleri ise tek bir öğrenme hedefi bulunan, diğer öğrenme nesnelerinden bağımsız şekilde var olan, uygun şekilde birleştirilebilen küçük parçalardır [21].

Bir ders içeriğinin uzaktan öğretim faaliyeti içinde içerik olarak kullanılması adına SCORM içerik tanımlama standartları içinde yapılması gereken 3 adet temel adımdan bahsedilebilir. Bunlar:

İçerik birleşim modeli; SCORM uyumlu bir içerik paketinin tanımlanmasını ve barındırdığı varlıkların birleşimini konu alır. XML tabanlı bir yapı söz konusudur.

Çalışma ortamı; SCORM uyumlu içerik paketinin ÖYS ile haberleşmesinin nasıl gerçekleşeceği konu alır. Burada JavaScript betik dili temelli bir yapı mevcuttur.

Sıralama ve dolaşım; SCORM paketi içerisinde yer alan varlık ve paylaşılabılır içerik nesnelerinin sıralaması ve öğrencinin bu paylaşılabılır içerik nesneleri arasındaki dolaşım kurallarını tanımlar. Daha çok paket içerisinde bulunan XML tabanlı metadata dosyasıyla ilgilidir.

2.3.1. İçerik Birleşim Modeli

İçerik birleşim modeli öğrenme etkinliğinde kaynak olarak kullanılabilecek varlıkların birleşerek öğrenme nesnesi haline gelmesi durumuyla ilgilidir. İçerik oluşturmak için kullanılan her bir yapıtaşının birleşim şemasını ve bu birleşim sonucu ortaya çıkan nesneyi tanımlama üzerine kuruludur. Bu modelin kuruluşunda karşımıza çıkan bazı bileşenler şu şekildedir:

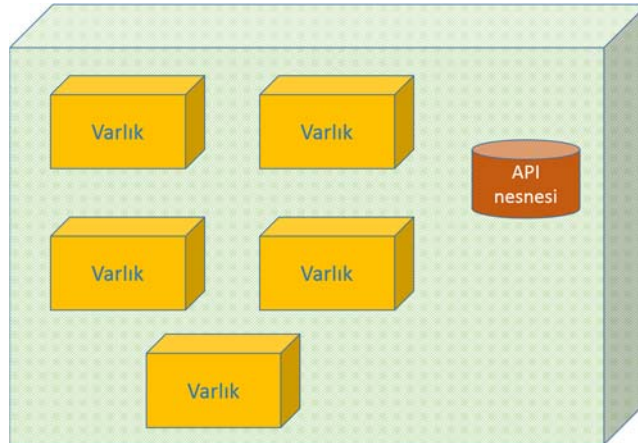
Varlıklar (Asset): içeriğin yapı taşı olarak kullanılan malzemelerin her biri olarak düşünülebilir. Bu varlıklar bir metin belgesi, resim, video, ses dosyası, web sayfası ve hatta arka planda kullanılan program kütüphane dosyalarından oluşabilir. Şekil 2.8’de varlıklara örnekler gösterilmiştir. Kısacası öğrenme faaliyetinde kullanılmak üzere görev alan her türlü kaynağın dijitalleştirilmiş halidir. Birkaç varlık bir araya gelerek yine bir varlığı oluşturabilmektedir. Aynı zamanda bir varlık XML dosya tanımlamaları sayesinde tekrar kullanım için ayarlanabilir ve arama sonuçlarına dâhil edilebilir.



Şekil 2.8. Örnek varlık çeşitleri.

Paylaşılabilir içerik nesnesi (SCO): bir veya birkaç varlıktan oluşan tek başına başlatılabilir en küçük öğrenme nesnesidir. ÖYS sistemi ile iletişimi SCORM çalışma ortamı tarafından sağlanır. ÖYS tarafından takip edilebilir en küçük SCORM nesnesidir. Bir paylaşılabilir içerik nesnesi temel olarak ders içeriğinin bütününden ayrı olarak düşünülmelidir. Bunun sebebi içerik için hazırlanan SCO(Sharable Content Object – Paylaşılabilir İçerik Nesnesi) paketinin farklı öğrenim ortamlarında farklı öğrenim hedefleri için tekrar kullanılmak istenebilir olmasıdır.

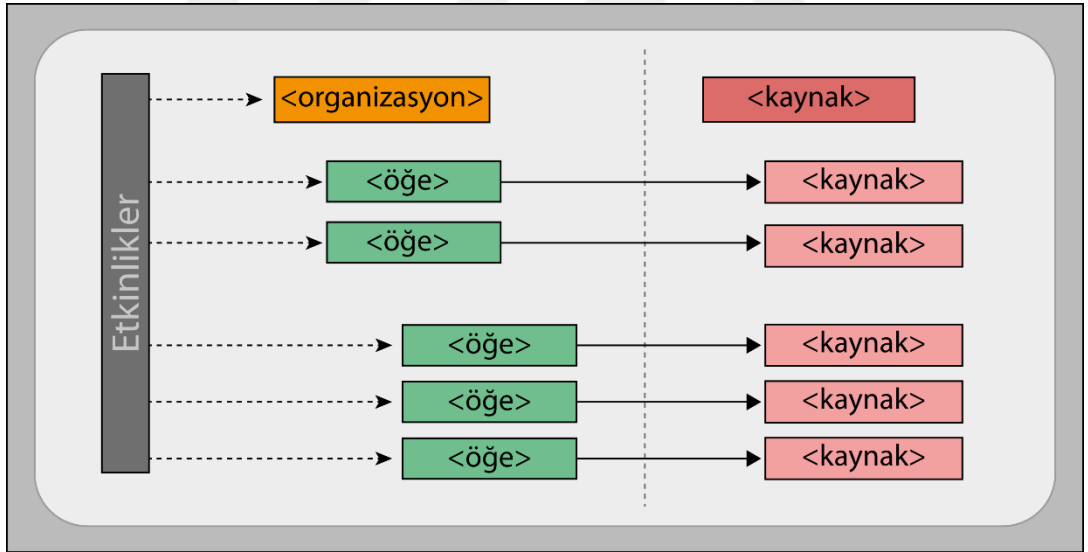
Bir SCO nesnesinin içeriği Şekil 2.9’da görselleştirilmiştir.



Şekil 2.9. Bir SCO nesnesi örnek gösterimi.

Herhangi bir SCO nesnesi SCORM RTE (Run Time Environment – Çalışma Zamanı Ortamı) tarafından belirlenen gereksinimlere bağlı kalmalıdır. Bunun anlamı minimum olarak SCORM çalıştırma ortamı tarafından sağlanan API içerisinde bulunan “Initialize(“”)” ve “Terminate(“”)” metotlarını barındırmalıdır.

Aktiviteler: bir içeriğin öğretimsel açıdan anlamlı en küçük parçası olarak ifade edilebilir. Bir aktivite birkaç SCO ve/veya varlıktan oluşabileceği gibi alt aktivitelerden de oluşabilir.

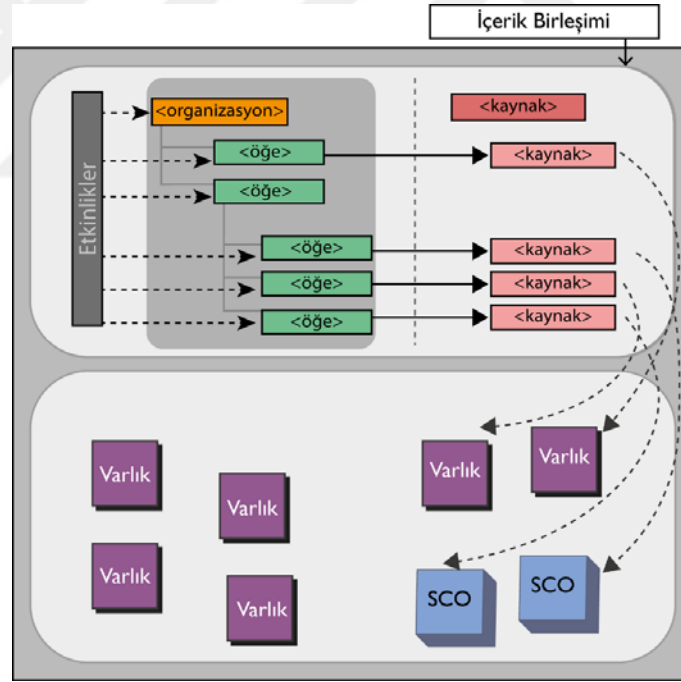


Şekil 2.10. Bir SCO nesnesinin hiyerarşik yapısı [22].

Şekil 2.10'dan anlaşılabileceği gibi bir aktivite bir çok alt öğeden oluşmaktadır. Öyle ki bir aktivitenin içerisinde bir alt aktivite de yer alabilir.

İçerik organizasyonu; öğrenme nesnelerinin yani aktivitelerin birbirleriyle olan ilişkilerini tanımlayan harita olarak düşünülebilir. İçerik organizasyonun metadata bilgisi olarak tutulmasıyla öğrenme nesnelerinin tekrar kullanılabilir olması sağlanmaktadır. Aynı zamanda bu sayede öğrenme nesnesinin bakım işlemini kolaylaşmaktadır. Sıralama sadece aktiviteler arasında yapılmaktadır. Burada sıralamadan kasıt aktivitelerin uygulanmasında izlenecek sıralamadır. Sıralamanın değerlendirilmesi de içerik organizasyonunun dâhilinde kabul edilmektedir.

İçerik birleşimi; SCORM paketi içerisinde aktiviteler ve varlıklar arasındaki fonksiyonel ilişkiyi tanımlamaktadır. Bu durum Şekil Şekil 2.11’de görsel olarak ifade edilmiştir.



Şekil 2.11. İçerik birleşiminin şematik gösterimi [22].

3. GELİŞTİRİLEN YAZILIM

MS PowerPoint Microsoft firmasının paket programlar serisinde yer alan sunum hazırlama yazılımıdır. MS PowerPoint gerek iş çevresinde gerekse öğrencilerce oldukça sık kullanım oranına sahiptir. Bunun anlamı oldukça geniş bir kitle tarafından kullanılmasıdır. Buna bağlı olarak günümüz modern öğretim ortamlarında öğretim materyali hazırlanmak amacıyla sıklıkla kullanılmaktadır. Bu çalışmada MS PowerPoint uygulamasına eklenti olacak şekilde bir yazılım geliştirilmiştir. Geliştirilmiş olan bu yazılım sayesinde hazırlanan herhangi bir temel düzeyde ki MS PowerPoint sunum dosyası web ortamında kullanılabilecek SCORM uyumlu bir öğretim materyaline dönüştürülebilmektedir [23].

3.1. Geliştirilen Yazılımın İşleyişi

Geliştirilen yazılımda .Net Framework kullanılmıştır. Programlama dili olarak C# tercih edilmiştir. Yazılım içerisinde SCORM uyumluluğunun eklenmesi isteğe bağlı olarak ayarlanabilmektedir. Ancak herhangi bir sunum dosyası eğitim materyali olarak hazırlanmak istendiğinde öncelikle HTML çevrim işlemi gerçekleştirilmelidir.

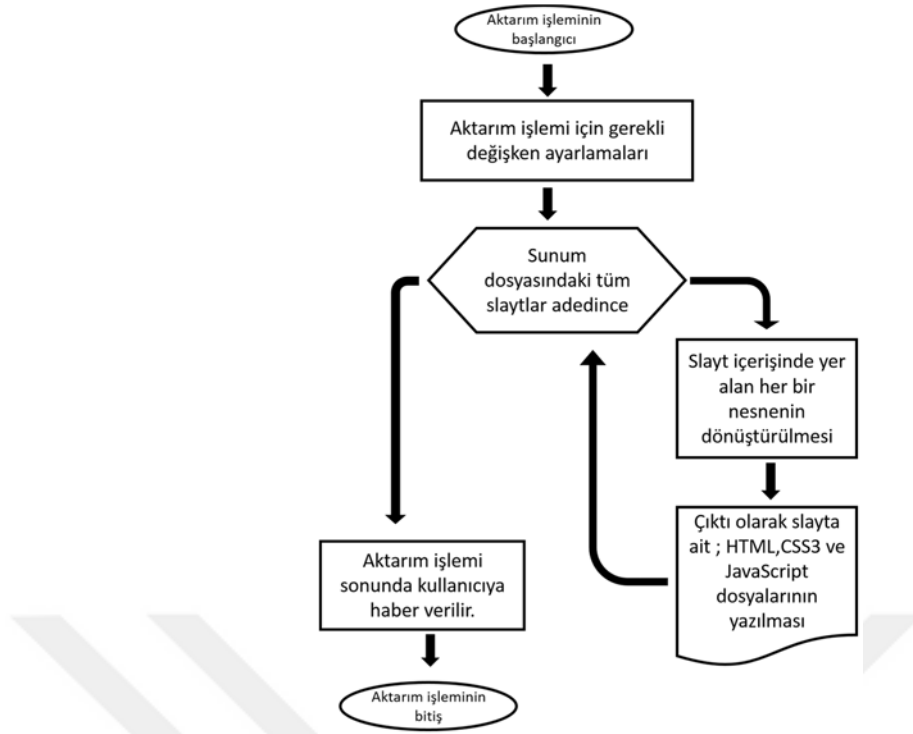
Yazılım temelinde .NET Framework yapısı kullanılması sayesinde PowerPoint sunusunda yer alan her bir elamanın özellikleri Office SDK (Software Development Kit – Yazılım Geliştirme Araç Kutusu) kullanılarak

rahatlıkla incelenebilmektedir. Bu sayede görsel zenginlik katan animasyonlar da incelenerek CSS3 ve JavaScript olarak dışa aktarılmaktadır.

Geliştirilen yazılımın kullanılabilmesi için sistem üzerinde Windows tabanlı bir işletim sistemi ve bunun yanında .NET Framework 4.0 ve MS Office paket programlarından en az PowerPoint versiyon 2007 yazılımının yüklü olması gerekmektedir. Yazılım işletim sistemine yüklendikten sonra PowerPoint paket programına eklenti olarak yerini almaktadır. Yazılımın genelinde 1500 üzeri satır kod yazılmıştır. Yazılım içerisinde 20 üzeri sınıf, struct ve enum yapısı oluşturulmuştur.

3.1.1. HTML Aktarım Modülü

Bu modül kullanılarak sunum dosyası içerisinde yer alan her bir slayt incelenir (Bkz. Ek1). İnceleme sonucu slayt içerisinde yer alan her bir nesneye uygun dönüşüm kodları hazırlanır ve slayt için hazırlanmış klasör içerisine HTML, JavaScript ve CSS çıktı dosyaları kaydedilir. Şekil 3.1'de aktarım işlemine ait akış şeması görülmektedir.



Şekil 3.1. HTML, CSS ve JavaScript *aktarım algoritması akış şeması*.

PowerPoint SDK kullanılarak incelenen her bir nesne “Shape” olarak ele alınır ve HTML içerisinde “<div>” etiketi kullanılarak temsil edilir. Şekil Şekil 3.2’de örnek HTML aktarım sonucu görülmektedir.

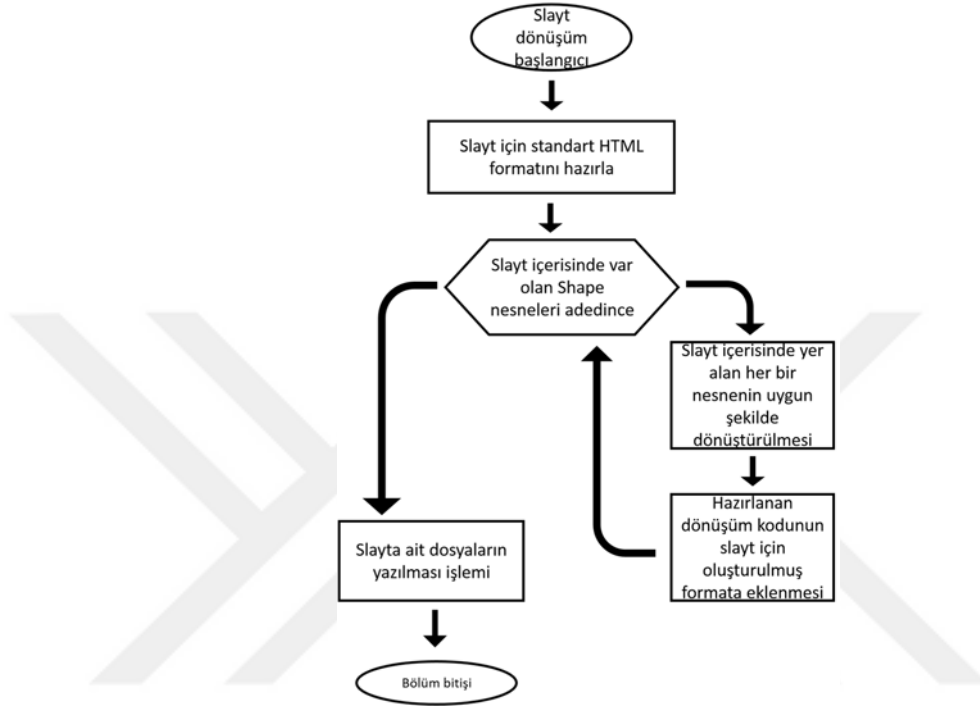
```

<div id="contentWrapper">
  <div id="shape2">
    <div id="shape2_text">
      <p id="shape2_text_parag1">
        <span id="shape2_text_parag1_run1">Örnek Eğitim Materyali</span><br>
      </p>
    </div>
  </div>
</div>

```

Şekil 3.2. Örnek HTML aktarım sonucu oluşan HTML kodlama

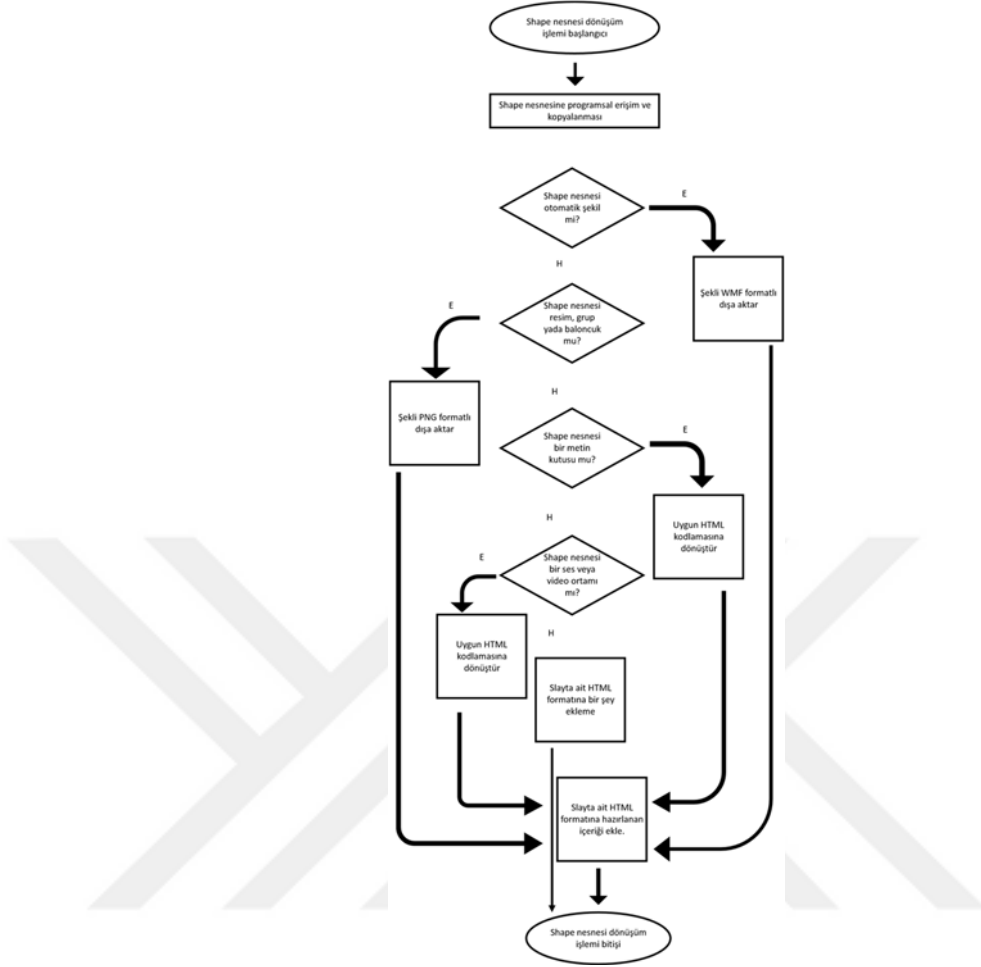
Şekil 3.3’de bir slaytın dönüştürölme işlemine ait algoritma akış şeması yer almaktadır.



Şekil 3.3. Slayt dönüştürölme işlemi akış şeması.

3.1.2. Ortam Aktarım Modülü

Bilindiği üzere PowerPoint sunum dosyalarına birçok ortam türü eklenebilmektedir. Bunların önde gelenleri; resim, ses, video ve şekil olarak sıralanabilir. Ancak ortam türü ne olursa olsun geliştirilen yazılım içerisinde “Shape” nesnesi olarak ele alınır (Bkz. Ek1). Şekil 3.4’de bir “Shape” nesnesinin dönüştürölmesine ait akış şeması görölmektedir.



Şekil 3.4. Shape nesnesi dönüşüm algoritması akış şeması

Şekil Şekil 3.4'den anlaşılacağı gibi farklı Shape nesnesi türleri için farklı yöntemler kullanılmaktadır. Eğer nesne bir otomatik şekilse WMF olarak belirli bir klasöre kayıt edilir. Şekil 3.5'de WMF olarak dışa aktarmak için kullanılan kod bloğu görülmektedir.

```

private void ExportShapeAsWMF(Shape shapeToExport, string pathToExport)
{
    shapeToExport.Select();

    object[] args = new object[] { pathToExport, PpShapeFormat.ppShapeFormatWMF, 1, 1,
        PpExportMode.ppRelativeToSlide };
    shapeToExport.GetType().InvokeMember("Export", System.Reflection.BindingFlags.InvokeMethod, null,
        shapeToExport, args);
}

```

Şekil 3.5. Nesnenin WMF formatlı dışa aktarım için kullanılan kod bloğu

Eğer nesne resim, grup veya bir konuşma baloncuğu ise PNG formatlı dışa aktarım gerçekleştirilmektedir. Şekil Şekil 3.6'de dönüşüm için kullanılan kodlar görülmektedir.

```

private void ExportShapeAsPNG(Shape shapeToExport, string pathToExport)
{
    shapeToExport.Select();
    shapeToExport.Rotation = 0F;
    object[] args = new object[] { pathToExport, PpShapeFormat.ppShapeFormatPNG, 0, 0,
        PpExportMode.ppClipRelativeToSlide };
    shapeToExport.GetType().InvokeMember("Export", System.Reflection.BindingFlags.InvokeMethod, null,
        shapeToExport, args);
    shapeToExport.Delete();
}

```

Şekil 3.6. Nesnenin PNG formatlı dışa aktarımı için kullanılan kod bloğu

Eğer nesne bir metin kutusu ise HTML dosyasında temsili için uygun dönüşüm gerçekleştirilir. Şekil Şekil 3.6'da dönüşüm için kullanılan kodlar görülmektedir.

```

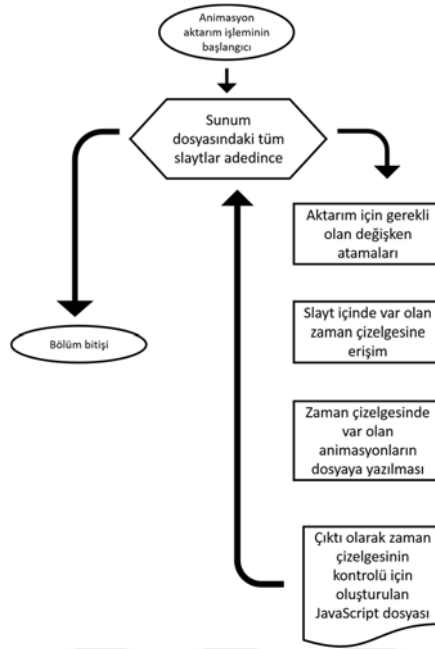
private H.HtmlNode PrepareTextBoxNode(Shape sp, Slide sld)
{
    H.HtmlDocument hd = new H.HtmlDocument();
    H.HtmlNode shapeNode = ShapeHelper.PrepareShapeCover(sp, sld);
    string shapeId = shapeNode.Attributes["id"].Value;
    if (sp.HasTextFrame == C.MsoTriState.msoTrue)
    {
        string subCss = "";
        H.HtmlNode divTextFrame = ShapeHelper.PrepareTextFrameNode(sp, shapeId, out subCss);
        shapeNode.AppendChild(divTextFrame);
        listShapeSubCsses.Add(subCss);
    }
    return shapeNode;
}

```

Şekil 3.7. Metin kutusunu HTML olarak dışa aktarmak için kullanılan kod bloğu

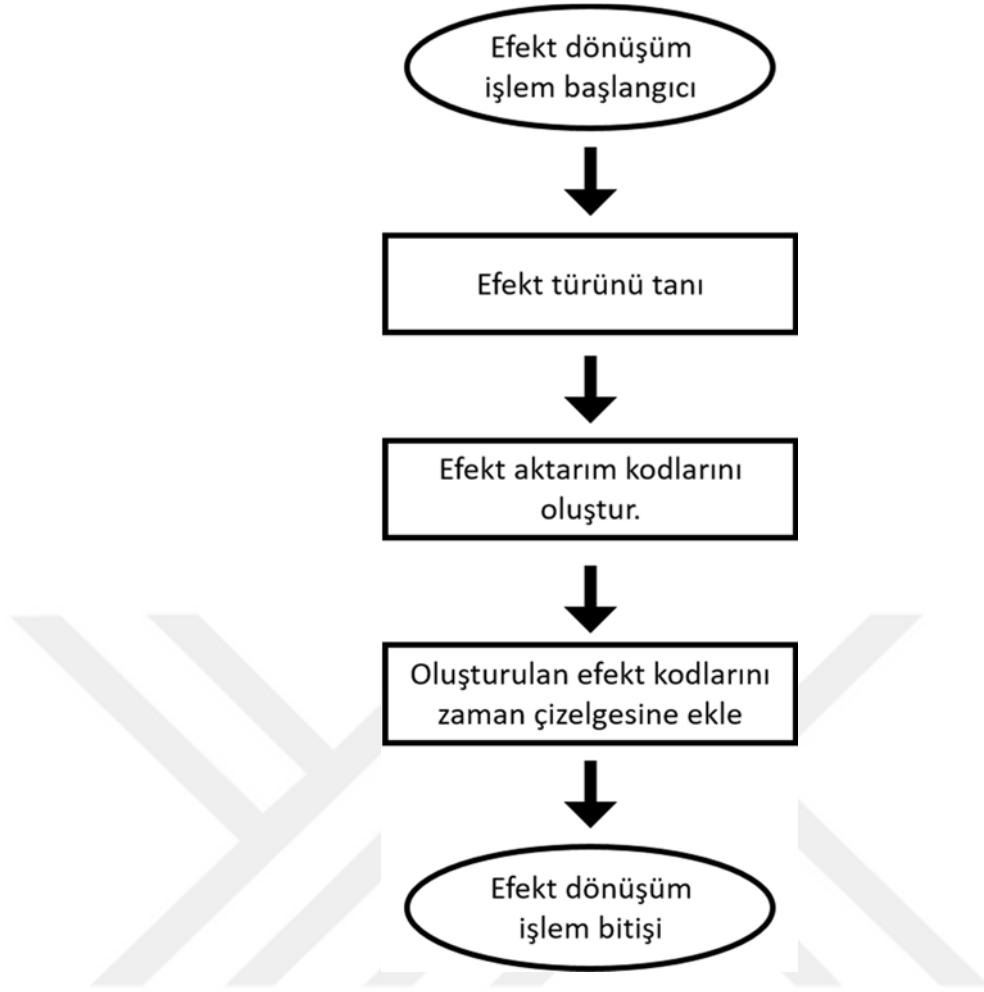
3.1.3. Animasyon Aktarım Modülü

Animasyon aktarımı için yazılım içerisinde oluşturulan “Animation” sınıfı kullanılmaktadır (Bkz. Ek2). Şekil **Şekil 3.8**'de animasyonların oluşturulmasına ilişkin akış şeması görülmektedir.



Şekil 3.8. Animasyonların oluşturulmasına ait algoritma akış şeması

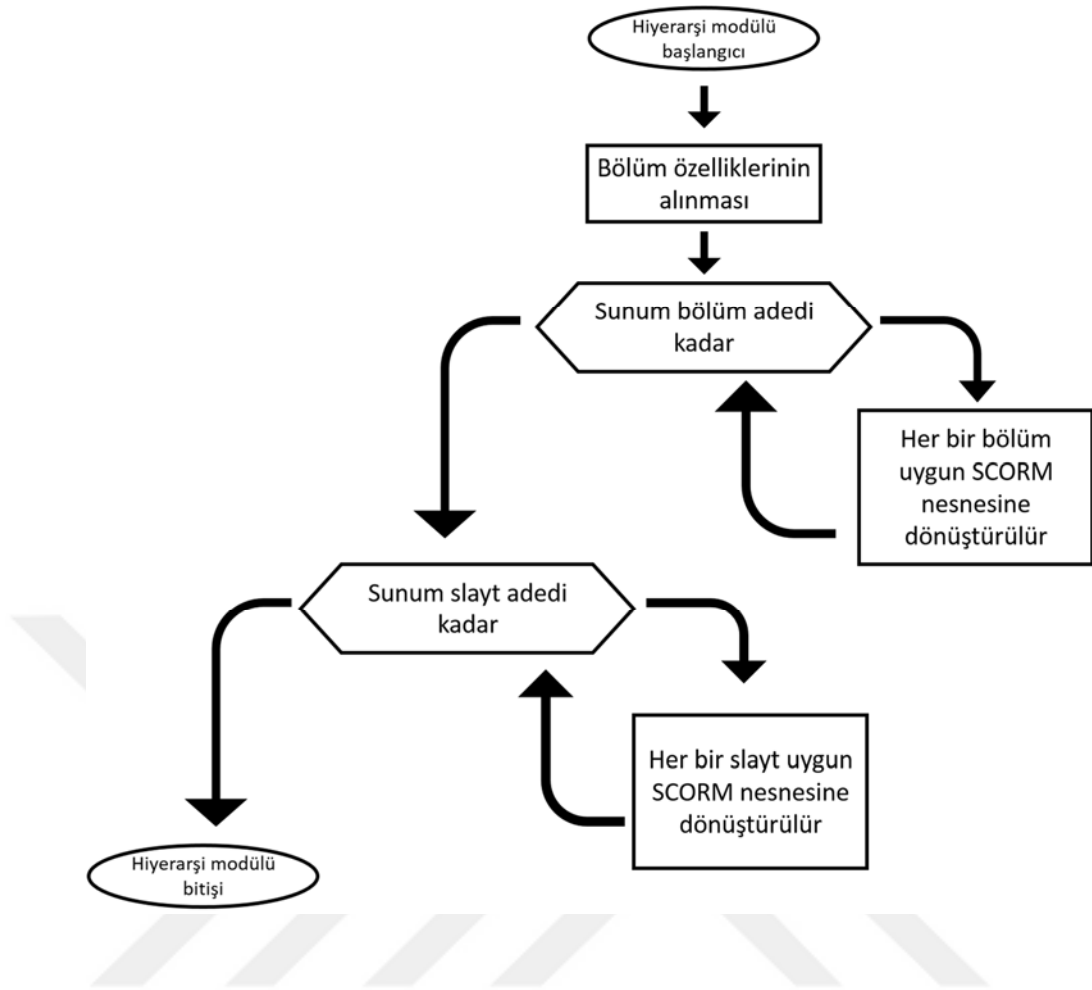
Bazı animasyonların oluşturulması için sadece CSS3 yeterken bazıları içinse CSS3 ve beraberinden JavaScript kodları gerekmektedir. Animasyon aktarım işlemlerinde en önemli nokta her slayt için var olan zaman çizelgesini dönüştürebilmektedir. Zaman çizelgesini oluşturmak için de yazılım içerisinde oluşturulan “Animation” sınıfından yararlanılır. Şekil 3.9’da efektlerin dönüştürülmesi ve zaman çizelgesine ilişkin akış şeması görülmektedir.



Şekil 3.9. Efekt dönüşüm akış şeması

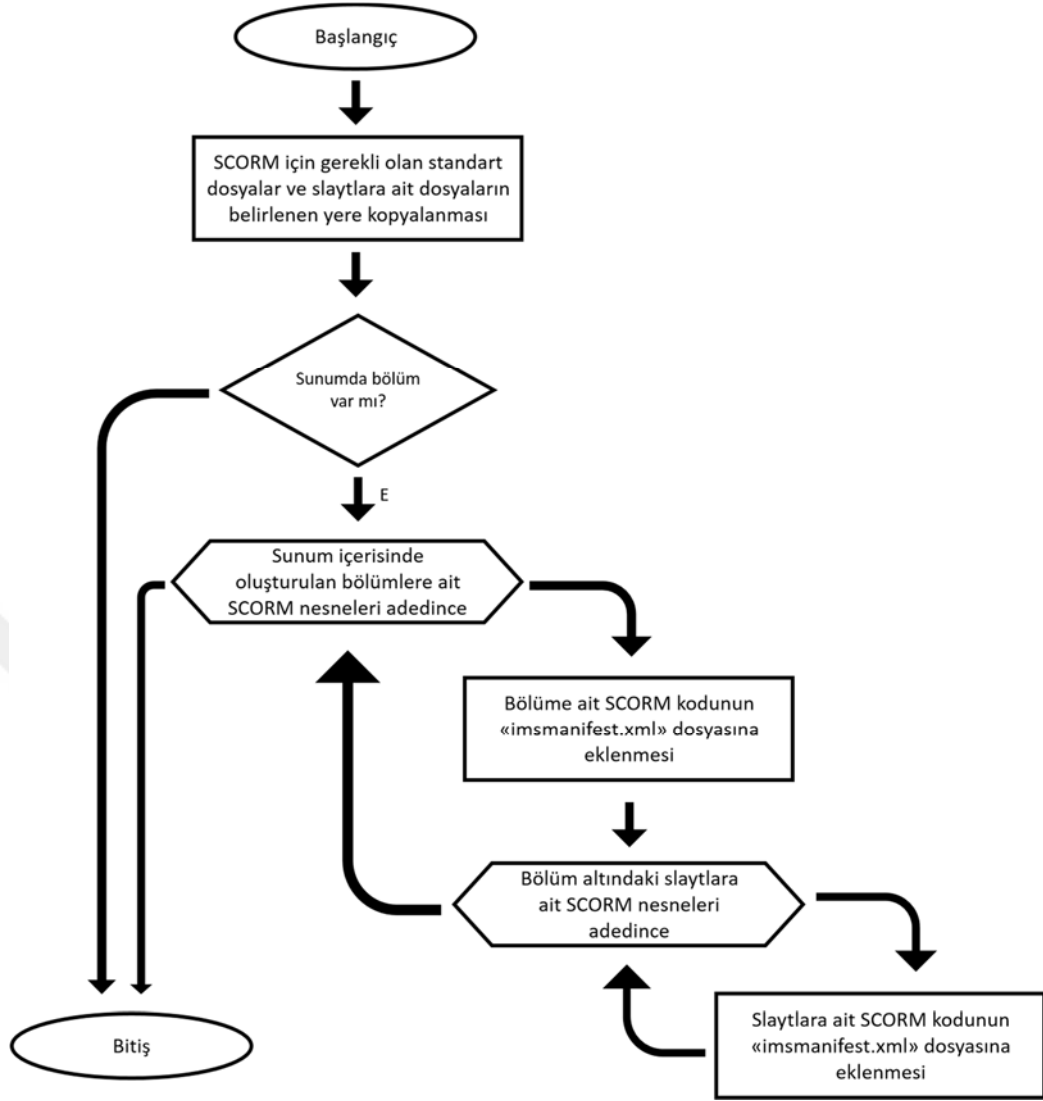
3.1.4. SCORM Uyumluluk Modülü

SCORM uyumluluğu eklemekten önce öğretim içeriği için de önemli olan hiyerarşi düzenlemesi yapılmalı (Bkz. Ek3). Şekil Şekil **3.10**'da hiyerarşi düzenleme işlemiyle ilgili akış şeması mevcuttur.



Şekil 3.10. Hiyerarşi düzenleme algoritması akış şeması.

Hiyerarşi düzenleme işlemi ardından oluşturulan ve program içerisinde kayıtlı tutulan SCORM nesneleri, nesne türüne uygun şekilde “imsmanifest.xml” dosyasına yazılır (Bkz. Ek1). Şekil Şekil 3.11’de “imsmanifest.xml” dosyasına yazılmasına ilişkin akış şeması görülmektedir.



Şekil 3.11. SCORM aktarım işleminin akış şeması.

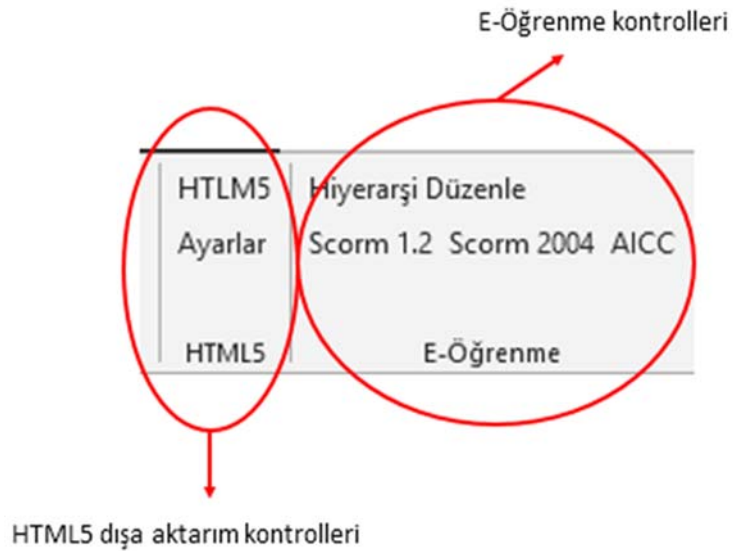
3.2. Geliştirilen Yazılımın Kullanımı

Kurulan yazılım, Office PowerPoint paket programında menü şeridine “E-Öğrenme” etiketli yeni bir sekme eklemektedir. Şekil Şekil 3.12’de yazılım tarafından eklenen menü görülmektedir.



Şekil 3.12. Yazılımın menü görüntüsü

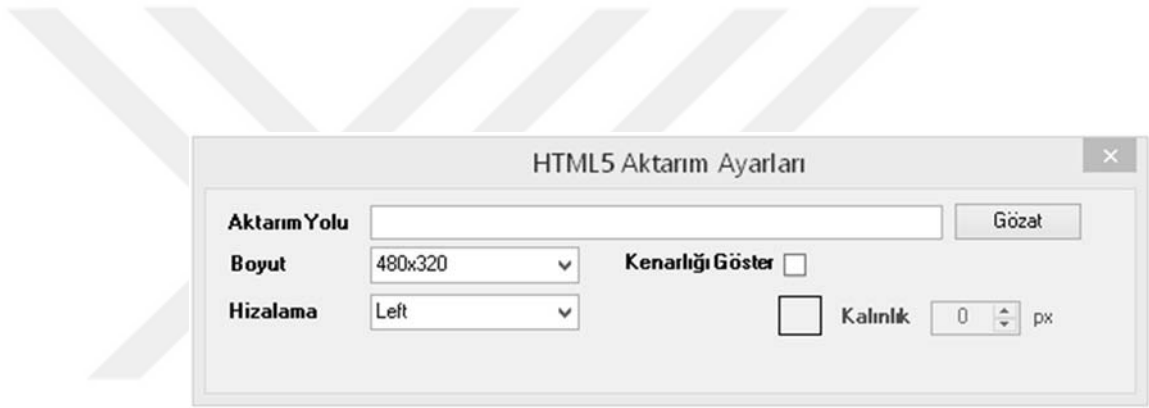
Geliştirilen yazılım esnek kullanım imkânı sağlanması için tasarlanmıştır. Buna göre yazılım içerisinde aktarma işlemleri için kullanılacak ayarla ve HTML5 formatlı aktarma işlemleri için kullanılan kontroller bir menü grubunda, öğretim amaçlı kullanılacak “SCORM” uyumlu paketlerin oluşturulması için kullanılan kontroller bir grupta yer almaktadır. Şekil Şekil 3.13’de yazılım kontrollerinin bulunduğu gruplar görünmektedir.



Şekil 3.13. Yazılım kontrol grupları

Sunum programı içerisinde yer alan animasyonlar ele alındığı kadarıyla HTML5 ve CSS3 teknolojileri kullanılarak JavaScript yardımıyla dışa aktarımı gerçekleştirilmiştir.

Yazılım içerisinde aktarım ayarlarına “HTML5” grubu içerisinde yer alan “Ayarlar” kontrolü ile erişilmektedir. Bu ayarlar “aktarım yolu”, “içerik boyutu”, “içerik hizalama”, “kenarlık gösterimi”, “kenarlık kalınlığı” gibi düzenlemeleri içermektedir. Şekil Şekil 3.14’de yazılıma ait HTML aktarım ayarları diyalog penceresi görülmektedir.



Şekil 3.14. HTML5 aktarım ayarları diyalog formu

Herhangi bir MS PowerPoint sunum dosyası geliştirilen yazılım kullanılarak HTML5 dönüşümü ve/veya SCORM paket dönüşümü yapılmak istendiğinde öncelikle ilgili sunum dosyasının MS PowerPoint programıyla açılması gereklidir. HTML5 aktarım işlemi başlatıldığında sunum dosyası içerisinde bulunan tüm slaytlar sırayla tek tek işlem görmekte ve slaytlarda yer alan her türlü nesne yazılım içerisinde “PowerPoint shape” nesnesi olarak işlem görmektedir.

Bir slayt işlem görmeye başladığında yazılım içerisinde sırasıyla;

- i. Öncelikle yazılım aktif sunum dosyasının kayıtlı olup olmadığını kontrol eder ve kayıtlı değilse kayıt edilmesi gerektiğini bildirir ve işlemi sonlandırır. Eğer kayıtlıysa sıradaki işlemlere devam eder.
- ii. Slayt arka planı belirlenen dizin yapısı içerisinde slayt numarasıyla resim olarak kaydedilir.
- iii. Slayt üzerinde bulunan tüm elamanlar “shape” nesnesi olarak ele alınır ve türlerine göre işlem görmeye başlar; temel olarak eğer nesne bir şekilse Windows Meta File dosya formatıyla dışarıya aktarılır, eğer bir medya nesnesiyse uygun formatta dışa aktarılır.
- iv. Windows Meta File dosyası olarak dışa aktarılmış olan görüntü nesneleri HTML5 belgesi içerisinde rahatlıkla kullanılabilmesi için Scalable Vector Graphics formatlı dosyalara dönüştürülür ve yazılım içerisinde daha önceden belirlenmiş olan yere kaydedilir. Şekil 3.15’de SVG(Scalable Vector Graphics – Ölçeklenebilir Vektör Grafikleri) formatına örnek kodlar görülmektedir.

```
<svg id="svg2" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:cc="http://creativecommons.org/ns#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:svg="http://www.w3.org/2000/svg" xmlns="http://www.w3.org/2000/svg" version="1.0" width="69.139999mm" height="37.119999mm" style="margin: 20px;">
<g id="g6">
<path id="path8" d="m 0.44978749,0.74901798 0,127.33305202 240.78623251,0 0,-127.33305202 -240.78623251,0 z" inkscape:connector-curvature="0" style="fill:#4f81bd;fill-opacity:1;fill-rule:evenodd;stroke:none">
<path id="path10" d="m 0.44978749,0.74901798 240.78623251,0 0,127.33305202 -240.78623251,0 z" inkscape:connector-curvature="0" style="fill:none;stroke:#385d8a;stroke-width:2.5487957px;stroke-linecap:round;stroke-linejoin:round;stroke-opacity:1;stroke-dasharray:none">
</g>
</svg>
```

Şekil 3.15. Örnek SVG formatı kodları

- v. Görüntü tabanlı aktarım işlemleri tamamlandıktan sonra her slayt için bir HTML dosyası oluşturulur. Şekil 3.16'da oluşturulan HTML dosyasından bir bölüm görülmektedir.

```
<head>
  <link rel="stylesheet" href="css\Slide2-style.css" />
  <link rel="stylesheet" href="css\cssKeyframes.css" />
  <script src="js/jquery-2.1.0.js"></script>
  <script src="js/jsAnimations.js"></script>
  <script src="js/Slide2-animations.js"></script>
  <title></title>
</head>
```

Şekil 3.16. Yazılım kullanılarak oluşturulmuş HTML sayfasından örnek

- vi. Oluşturulan HTML dosyası içerisine slayt yapısını yansıtacak şekilde HTML elemanları yerleştirilir. İçerik alanının boyutu HTML aktarım ayarlarında belirlendiği şekilde düzenlenir. HTML yapısının oluşturulması aşamasında ayrıca harici bir CSS3 dosyası da içerik biçimleme işlemi için hazırlanır ve ilgili HTML dosyasına bağlanır. Şekil 3.17'de örnek CSS3 dosyası içeriği görülmektedir.

```

@-webkit-keyframes msoAnimEffectBlinds_shape4
{
    from    {          height: 0px;    }
    to      {          height: 32.757px;  }
}

@-moz-keyframes msoAnimEffectBlinds_shape4
{
    from    {          height: 0px;    }
    to      {          height: 32.757px;  }
}

@-o-keyframes msoAnimEffectBlinds_shape4
{
    from    {          height: 0px;    }
    to      {          height: 32.757px;  }
}

@keyframes msoAnimEffectBlinds_shape4
{
    from    {          height: 0px;    }
    to      {          height: 32.757px;  }
}

```

Şekil 3.17. Animasyonu aktarmak için kullanılan stil kodlarına örnek

- vii. HTML dosyası oluşturulurken “shape” nesnesi olarak ele alınan nesnenin üzerinde animasyon varsa ilgili animasyonun CSS3 ile kodlanmış hali HTML dosyası ile bağlı olan CSS3 dosyasına yazılır. Ayrıca slayt üzerinde yer alan “shape” nesnesinde animasyon yer aldığı için zamanlama ve tetikleme işlemlerinin yönetimini yapmak üzere bir JavaScript dosyası oluşturulur. Slayt animasyon zaman çizelgesi bu dosyaya daha önce yazılım içerisinde belirlenen biçimde yazılır ve bu dosya da ilgili HTML dosyasına bağlanır.

Yapılmış Çalışmalar

- Konuk kullanıcı paketlerinin tünellemesi
Sastry et al., 2007
- İzinsiz giriş tespiti
Takahashi et al., 2010
- Kullanıcı doğrulaması
Heer et al., 2007
- AP kapsama alanı çakışması ve genişliği
Amaldi et al. 2004,
- Kablosuz kapsama alanını genişletmek için var olan geniş bant hizmetinin kullanımı
Solarski, M., 2006

Şekil 3.18. Aktarımı gerçekleştirilen slayta ait HTML sayfa görüntüsü

Şekil 3.18’de aktarımı gerçekleştirilen slayta ait HTML sayfası görseli bulunmaktadır.

MS Office PowerPoint paket programının öğretim faaliyetlerinde kullanımının tercih edilmesinin sebeplerinden biri de katılımcılara içeriği daha dikkat çekici şekilde sunulmasını sağlayan animasyon(lar) eklenebilmesidir.

MS Office PowerPoint yazılımı içerisinde herhangi bir nesneye uygulanabilecek 100 üzerinde animasyon bulunmaktadır. Bu animasyonlardan “Görünüş”, “Perdeler”, “İçeri Kaydır” ve “Şekil (Kutu)” isimli animasyonlar yazılım içerisinde dışa aktarım için seçilmişlerdir. Elbette diğer animasyon çeşitleri de uygun dönüşüm kodlarıyla dışa aktarılabilir hale getirilebilirler. PowerPoint paket programında yer alan animasyonların dışa aktarımı için

CSS3 teknolojisi içerisinde yer alan “@keyframes” anahtar kelimesiyle oluşturulan kurallar bütünü ve “animation” anahtar kelimesiyle gerçekleştirilen animasyonlar kullanılmaktadır. Şekil 3.19’da “@keyframes” anahtar kelimesinin kullanımına ait örnek görülmektedir.

```
@keyframes animasyon_adi
{
  from
  {
    background-color: red;
  }
  to
  {
    background-color: black;
  }
}
```

Şekil 3.19. “@keyframes” kullanımına bir örnek

“@keyframes” anahtar kelimesi standart olarak ayarlanmak istenen bir animasyonun başlangıç kurallar kümesi (“from” bloğu) ve bitiş kurallar (“to” bloğu) kümesini tanımlar. Eğer istenirse bu kural kümeleri yüzdelik dilim olarak da kullanılabilir. Şekil 3.20’de “@keyframes” anahtar kelimesinin yüzdelik dilimlerle kullanımına örnek görülmektedir.

```
@keyframes animasyon_adi
{
  10%    {top:10px;background-color: red;}
  20%    {top:20px;background-color: yellow;}
  50%    {top:30px;background-color: black;}
  90%    {top:40px;background-color: white;}
  100%   {top:60px;background-color: navy;}
}
```

Şekil 3.20. "@keyframes" anahtar kelimesinin yüzdelerle kullanımı.

"@keyframes" anahtar kelimesi ile kuralları belirlenen animasyon "animation" CSS3 anahtar kelimesiyle istenilen bir HTML elemanına atanabilmektedir. Burada dikkat edilmesi gereken CSS3 versiyonu ile gelen bazı yeni stil özellikleri web tarayıcısı üreticilerinin kullandıkları HTML motoruna göre farklılık göstermektedir.

Çizelge 3.1'de görüldüğü üzere web tarayıcı üreticileri yazılımlarında farklı HTML yorum motorları kullanmaktadır. Dolayısıyla tüm CSS3 stil özelliklerinin doğru şekilde kullanılabilmesi ihtiyaç duyulan anahtar kelimelerin önlerine üreticilerin kullandıkları HTML yorum motorlarına ait ön eklerin eklenmesi gerekir.

Çizelge 3.1. Popüler web tarayıcıların kullandıkları HTML yorum motorları ve CSS3 ön ekleri.

Web Tarayıcı	HTML Yorum Motoru	CSS3 Ön Eki
Google Chrome	WebKit	-webkit-
Internet Explorer	Trident	-ms-
Mozilla Firefox	Gecko	-moz-
Opera	WebKit	-webkit-
Safari	WebKit	-webkit-

Şekil 3.21’de görüldüğü üzere “animation” anahtar kelimesi kullanılarak oluşturulan CSS kuralının Google Chrome ve Safari web tarayıcılarda geçerli olabilmesi için “-webkit-animation” şeklinde yazılmalıdır. Web tarayıcı üreticilerine göre ön eklerin kullanımı Şekil Şekil **3.21**’de örnekle gösterilmiştir.

```
.appear
{
    /*Opera web tarayıcı*/
    -o-animation: msoAnimEffectAppear 2s infinite;
    /*Mozilla Firefox web tarayıcı*/
    -moz-animation: msoAnimEffectAppear 2s infinite;
    /* Safari and Chrome web tarayıcı*/
    -webkit-animation: msoAnimEffectAppear 2s infinite;
    /*Standart anahtar kelime*/
    animation: msoAnimEffectAppear 2s infinite;
}
```

Şekil 3.21. Web tarayıcı üreticilerine göre stil etiketlerinin ön ek ile kullanımına örnek.

Geliştirilen yazılım içerisinde CSS3 kodlarına çevrilen animasyonlar programın kendisine has olarak JavaScript kullanılarak hazırlanan “EffectObject”, “TimingObject” ve “TimeLineObject” yapıları kullanılarak zamanla ve tetikleme işlemlerinin yönetimi gerçekleştirilmektedir. Bunlarda Şekil 3.24’de yer alan “TimeLineObject” yapısı bir slayt için geçerli olan zaman çizelgesini oluşturmak, Şekil 3.22’de gösterilen “EffectObject” yapısı herhangi bir nesneye uygulanmış animasyonu temsil etmek ve Şekil 3.23’de gösterilen

"TimingObject" yapısı ise bir animasyonun zaman çizelgesindeki yaşamını ifade etmek için kullanılmaktadır.

```
function EffectObject()
{
    //jquery ile kapsüllenmiş hedef nesne
    this.Shape = null;
    //Çıkış animasyonu için true
    this.Exit = false;
    //effect sıra numarası
    this.Index = 0;
    this.Paragraph = 0; //effect uygulanacak paragraf numarası
    this.TextRangeLength = 0; //effect uygulanacak metin uzunluğu
    this.TextRangeStart = 0; //effect uygulanacak metnin başlangıç noktası
    this.Timing = new TimingObject();
    this.EffectFunction = null
    this.EffectName = "";
}
```

Şekil 3.22. JavaScript ile geliştirilen "EffectObject" yapısı.

```
function TimingObject() {
    this.Accelerate = 0; //float
    this.AutoReverse = false; //geri sarma olsun mu?
    this.Duration = 0; //float , seconds
    this.RepeatCount = 0; //tekrar sayısı
    this.RepeatDuration = 0; //tekrar durumunda maksimum saniye , float
    this.RewindAtEnd = false; //bittğinde başlangıca dön ve dur.
    this.TriggerDelayTime = 0; //float, tetikleyici etkinse geçikme zamanı.
    this.TriggerType = MsoAnimTriggerType.msoAnimTriggerNone; //tetikleyici tipi
}
```

Şekil 3.23. JavaScript kullanılarak geliştirilen "TimingObject" yapısı

```
function TimeLineObject() {
    this.que = new Array();
    this.addEffectToQue = function (eff) {
        this.que.push(eff);
    }
    this.processQue = new Array();
    this.startTimeLine = function ()...
```

Şekil 3.24. JavaScript kullanılarak geliştirilen "TimeLineObject" yapısı.

Bir MS PowerPoint sunum dosyası geliştirilen yazılım kullanılarak SCORM uyumlu bir ders materyaline dönüştürülmek isteniyorsa öncelikle HTML aktarım işlemi gerçekleştirilmelidir. Bu işlem gerçekleştirilmişse SCORM aktarımı işleminde adımlar şu şekildedir;

- i. Daha önce HTML formatlı aktarım yapılan klasör konumu içerisine "scorm" isimli bir klasör açılır.
- ii. HTML aktarım sonucu oluşan dosyalar bu klasörün içerisine kopyalanır.
- iii. İstenilen SCORM aktarım formatına uygun XML ve XML şema tanımlama dosyaları oluşturulan "scorm" isimli klasör içerisine yerleştirilir.
- iv. HTML dosyalarının kullanılacağı SCORM destekli ÖYS üzerinde etkileşime girmesi için gerekli olan JavaScript komutlarının bulunduğu dosya da ilgili HTML dosyalarına bağlanır.

- v. SCORM uyumlu aktarım işleminin son aşmasında ise paket içeriğinin hiyerarşi düzeninin bulunduğu ve her SCORM paketinde bulunması gerekli olan “imsmanifest.xml” dosyası paket içerisine yerleştirilir.

Bilindiği üzere MS PowerPoint programında, temel yapı taşları olan slaytlar sadece öncelik ve sonralık durumuna göre sıralanmışlardır. Ancak SCORM paketleri içerisinde asıl yapı hiyerarşik düzen gerektirmektedir. Bu düzenlemeye de geliştirilen yazılımda bulunan e-öğrenme kontrolleri içerisinde yer alan “Hiyerarşi Düzenle” kontrolü kullanılarak erişilebilir. Bu kontrol kullanıldığında Şekil 3.25’de görünen form açılmaktadır.

Şekil 3.25. Hiyerarşi düzenleme formu

Şekil 3.25’de gösterilen form sayesinde SCORM yapısı içerisindeki içerik ağacına MS PowerPoint slayt isimleri yerine ayarlanan ünite ve konu

başlıkları olması gerektiği gibi görünecektir. Böylece öğretim altyapısı için gereken uygun SCORM yapısı sağlanmış olmaktadır.

Geliştirilen yazılım şuan için sadece SCORM 1.2 versiyonuna uygun öğretim materyali oluşturabilmektedir. Hiyerarşi düzenlemesi yapıldıktan sonra oluşturulan içerik ağacına uygun şekilde “imsmanifest.xml” dosyası oluşturulup paket içine yerleştirilir.



4. SONUÇ

Gerçekleştirilen çalışmada öğretim ortamlarında sıklıkla kullanılan, dolayısıyla ders materyali olarak da hazırlanan MS Office PowerPoint sunum dosyalarının kolaylıkla bir ÖYS içerisine atılabilmesi ve istenirse SCORM uyumlu olabilmesi için bir yazılım geliştirilmiştir. Geliştirilen yazılımda sunum içeriklerinden slaytlar, şekiller, metin alanları, tanımlanan 5 adet animasyon başarılı bir şekilde dönüştürülebilmektedir. Ayrıca sunum dosyaları içerisinde yer alan video ve ses dosyaları da başarılı bir şekilde aktarılmıştır. Ancak video ve ses dosyalarının dışa aktarımı için sunum dosyasının minimum MS Office PowerPoint 2007 kullanılarak hazırlanmış olması gerekmektedir.

Geliştirilen yazılım farklı MS Office PowerPoint sunum dosyaları üzerinde çalıştırılmıştır. Bu denemeler sonucunda programın temel işleyiş olarak başarılı olduğu gözlemlenmiştir. Ancak aktarım ortamında biçimsel bir dil kullanıldığından çok ufak farklılıklar gözlemlenmiştir.

Ortaya çıkan yazılıma benzer olarak gösterilebilecek yazılımlar arasında yer alan iSpring Solutions şirketine ait iSpring Pro 7 yazılımıyla karşılaştırıldığında, geliştirdiğimiz yazılımda sunum dosyası içerisinde yer alan şekil, metin kutuları gibi ortamları aktarımında HTML5 öğelerinin kullanılması en önemli artı olarak gösterilmektedir. Bu sayede dönüştürme işlemi sonucu ortaya çıkan eğitim materyali üzerinde daha çok kontrol imkânı sağlanmış durumdadır.

Gerçekleştirilen yazılım çalışmasında elbette ki geliştirilebilir bir takım noktalar mevcuttur. Bunlar; yazılım içerisinde işleyişe yardımcı olunması

iin kullanılan farklı aık kaynak kodlu yazılımların ve kütüphanelerin yazılım boyutunu ve işlem zamanını arttırması, MS Office PowerPoint programı içerisinde uygulanabilen animasyonların henüz tümünün dışa aktarılmamış olması, sunucu destekli kontrollerin eklenebilir olması gibi sıralanabilir. İleride bu konuda yapılabilecek akademik alıřmalarda bu eksikliklerin giderilmesi iin amalanabilir.

Tüm bu geliştirilebilir noktaların yanında üzerinde alıřılan yazılım, öğretici personelin kaynak olarak bulundurduğu MS Office PowerPoint sunum dosyalarının minimum uğrař verilerek SCORM destekli bir uzaktan öğretim materyali haline dönüřtürölmesi iin başlangı seviyesinde yeterli olarak görülebilir. Yapılan bu alıřma, bu konuda gerekleřtirilecek akademik faaliyetlere ışık tutması iin bir başlangı adımı olarak görülebilir.

KAYNAKLAR

- [1] Anonim. Adult Learning Activities | California Distance Learning Project. [Online]. <http://www.cdiponline.org/index.cfm?fuseaction=whatis>
- [2] M. Moore and G. Kearsley, Distance Education : A System View Of Online Learning.
- [3] Karl Andersson and Dan Johansson, "Mobile e-Services Using HTML5," 6th IEEE Workshop On User MObility and VEhicular Networks, pp. 814-819, 2012.
- [4] Masahiro Araki and Daisuke Takegoshi, "A Rapid Development Framework for Multilingual Spoken Dialogue Systems," IEEE 36th International Conference on Computer Software and Applications, pp. 195-202, 2012.
- [5] Michael Nebeling, Fabrice Matulic, Lucas Streit, and Moira C. Norrie, "Adaptive Layout Template for Effective Web Content Presentation in Large-Screen Contexts," DocEng '11 Proceedings of the 11th ACM Symposium on Document Engineering, pp. 219-228, 2011.
- [6] Nelson Baloian, Francisco Gutierrez, and Gustavo Zurita, "An Architecture for Developing Distributed Collaborative Applications Using HTML5," Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design, pp. 213-220, 2013.
- [7] Hiroki Mori, Hisaharu Tanaka, Yoshiaki Hori, Makoto Otani, and Kenzi Watanabe, "Development of Lecture Videos Delivery System using HTML5 Video Element," 2013 Eighth International Conference on Broadband, Wireless Computing, Communication and Applications, pp. 557-559, 2013.
- [8] Guolei Zhu, Fang Zhang, Wei Zhu, and Yayu Zheng, "HTML5 Based Media Player for Real-Time Video Surveillance," 2012 5th International Congress on Image and Signal Processing, pp. 245-248, 2012.
- [9] Fabien Cazenave, Vincent Quint, and Cécile Roisin, "Timesheets.js: When SMIL Meets HTML5 and CSS3," DOCENG 2011: PROCEEDINGS OF THE 2011 ACM SYMPOSIUM ON DOCUMENT ENGINEERING, pp. 43-52, 2011.

- [10] Martin Stříbný and Pavel Smutný, "Using HTML5 Web Interface for Visualization and Control System," Carpathian Control Conference (ICCC), 2013 14th International, pp. 363-366, 2013.
- [11] Hwa-Young Jeong and Bong-Hwa Hong, "A service component based CAT system with SCORM for advanced learning effects," Multimedia Tools and Applications , vol. 1, no. 63, pp. 217-226, 2013.
- [12] Ana María Feroso GARCIA and Alberto Pedrero ESTEBAN, "ADAPTING SCORM TO M-LEARNING CONTEXT," in eLearning & Software for Education, Bucharest, 2011.
- [13] Dragoş CAZACU, Anatol OPREA, and Aurelian JURCOANE, "MLMS – ENABLING SCORM COMPATIBLE LEARNING MANAGEMENT SYSTEMS," in The 8th International Scientific Conference eLearning and software for Education, Bucharest, 2012.
- [14] Bartosz Sawicki and Bartosz Chaber, "Efficient visualization of 3D models by web browser," Computing, vol. 95, no. 1, pp. 661-673, 2013.
- [15] Michael Grahame Moore and William G. Anderson, HANDBOOK OF DISTANCE EDUCATION. London: LAWRENCE ERLBAUM ASSOCIATES, 2003.
- [16] RUSTICI Software. Scorm - Golf Examples. <http://scorm.com/wp-content/assets/SchemaDefinitionFiles/SCORM%201.1%20Schema%20Definition/CSF.xml>
- [17] Sabine Graf and Beate List, "An Evaluation of Open Source E-Learning Platforms Stressing Adaptation Issues," In Proceedings of 5th International Conference on Advanced Learning Technologies, pp. 163-165, 2005.
- [18] Ajlan S. Al-Ajlan. (2012) A Comparative Study Between E-Learning Features. [Online]. <http://cdn.intechopen.com/pdfs-wm/27926.pdf>
- [19] Peter Lubbers, Brian Albers, and Frank Salim, Pro HTML5 Programming.: Apress, 2010.
- [20] WIKIPEDIA , the free encyclopedia. http://en.wikipedia.org/wiki/Sharable_Content_Object_Reference_Model (Erişim tarihi : 2014, Nisan)

- [21] Luis Anido-Rifon Victor Gonzalez-Barbone, "Creating the first SCORM object," Computers & Education, vol. 51, pp. 1634-1637, 2008.
- [22] Angelo Panar, SCORM® 2004 4th Edition Content Aggregation Model (CAM) Version 1.1.: Advanced Distributed Learning, 2009.
- [23] Kadir HALTAŞ, Murat LÜY, Atilla ERGÜZEN, Ertuğrul ÇAM, and F.Mustafa AKALP, "UZAKTAN EĞİTİMDE MODERN DERS İÇERİKLERİ GELİŞTİRME YAZILIM UYGULAMASI," in International Science and Technology Conference, St. Petersburg, 2015, pp. 588-595.



EKLER

EK-1. Geliştirilen yazılımın genel kodlarının bulunduğu “HTML5.cs” dosyası.

```
using Microsoft.Office.Interop.PowerPoint;
using Microsoft.Office.Tools.Ribbon;
using System;
using System.Linq;
using System.Collections.Generic;
using System.Diagnostics;
using System.IO;
using System.Text;
using System.Windows.Forms;
using System.Xml;
using ToHtml5AddIn.PptHelper;
using ToHtml5AddIn.ScormHelper;
using C = Microsoft.Office.Core;
using H = HtmlAgilityPack;
namespace ToHtml5AddIn
{
    public partial class HTML5
    {
        List<string> listShapeSubCsses = new List<string>();
        protected string presentationPath = string.Empty;
        protected string svgPath = string.Empty;
        protected string wmfPath = string.Empty;
        protected string imgPath = string.Empty;
        UnicodeEncoding enc = new UnicodeEncoding();
        Presentation p = null;
        private void HTML5_Load(object sender, RibbonUIEventArgs e)
        {
            try
            {
                string pPath = Globals.ThisAddIn.Application.ActivePresentation.Path;
                presentationPath = System.IO.Path.GetDirectoryName(pPath);
            }
            catch (Exception)
            {
                presentationPath = string.Empty;
            }
        }

        private void btnPublish_Click(object sender, RibbonControlEventArgs e)
        {
            try
            {
                p = Globals.ThisAddIn.Application.ActivePresentation;
                if (p.Saved != C.MsoTriState.msoTrue)
                {
                    throw new Exception();
                }
            }
            catch
            {
                MessageBox.Show("Lütfen önce kayıtlı bir sunum dosyası açınız veya bu sunum dosyasını kaydediniz.");
                return;
            }
            if (Settings.Main.HTML5ExportPath.IsNullOrEmpty())
            {
                DialogResult dr = fbdChooseDir.ShowDialog(ThisAddIn.NW);
                if (dr == DialogResult.OK)
                {
                    Settings.Main.HTML5ExportPath = fbdChooseDir.SelectedPath;
                }
                else
                {
                    MessageBox.Show("Aktarım yolu seçilmediği için işlem bitirildi.");
                    return;
                }
            }
        }
    }
}
```

```

Settings.Main.HScaleRatio = Settings.Main.ContentSize.H / p.SlideMaster.Height;
Settings.Main.WScaleRatio = Settings.Main.ContentSize.W / p.SlideMaster.Width;
string dirName = p.Name.Replace(".pptx", "").Replace(".ppt", "");
DirectoryInfo exportDir = Directory.CreateDirectory(Settings.Main.HTML5ExportPath + @"\" + dirName);
DirectoryInfo wmfDir = Directory.CreateDirectory(exportDir.FullName + @"wmf");
DirectoryInfo svgDir = Directory.CreateDirectory(exportDir.FullName + @"\svg");
DirectoryInfo imgDir = Directory.CreateDirectory(exportDir.FullName + @"\img");
svgPath = svgDir.FullName;
wmfPath = wmfDir.FullName;
imgPath = imgDir.FullName;
int shapeIndex = 1;
int slideIndex = 1;
foreach (Slide sld in p.Slides)
{
    sld.Select();
    var frm = SurecFormuGetir(slideIndex++);
    string slideImageDirectoryPath = string.Format(@"{0}\{1}", imgDir.FullName, sld.Name);
    DirectoryInfo diSlideImages = Directory.CreateDirectory(slideImageDirectoryPath);
    H.HtmlDocument sldDoc = HtmlHelper.HDoc(true);
    H.HtmlNode container = sldDoc.DocumentNode.SelectSingleNode(@"//div[@id='contentWrapper']");
    string slideCss = "";
    Dictionary<string, string> cssWrapper = new Dictionary<string, string>();
    cssWrapper.Add("position", "relative");
    cssWrapper.Add("height", Settings.Main.ContentSize.H + "pt");
    cssWrapper.Add("width", Settings.Main.ContentSize.W + "pt");
    if (Settings.Main.ShowBorder)
    {
        cssWrapper.Add("border", string.Format(@"solid {0}px {1}", Settings.Main.BorderWidth,
Settings.Main.BorderColor));
    }
    string backgroundImage = ExportSlideBackground(sld, diSlideImages.FullName);
    cssWrapper.Add("background-image", "url(" + backgroundImage + ")");
    cssWrapper.Add("background-size", string.Format("{0}pt {1}pt", Settings.Main.ContentSize.W,
Settings.Main.ContentSize.H));
    switch (Settings.Main.Alignment)
    {
        case ToHtml5AddIn.Settings.ContentAlignment.Left:
            cssWrapper.Add("float", "left");
            cssWrapper.Add("margin-left", "0px");
            cssWrapper.Add("margin-right", "auto");
            break;
        case ToHtml5AddIn.Settings.ContentAlignment.Right:
            cssWrapper.Add("float", "right");
            cssWrapper.Add("margin-left", "auto");
            cssWrapper.Add("margin-right", "0px");
            break;
        case ToHtml5AddIn.Settings.ContentAlignment.Center:
            cssWrapper.Add("margin-left", "auto");
            cssWrapper.Add("margin-right", "auto");
            break;
        default:
            break;
    }
    string htmlTagCss = Environment.NewLine
        + HtmlHelper.PrepareGeneralStyle(cssWrapper, CssElementType.id,
container.Attributes["id"].Value)
        + Environment.NewLine;
    slideCss += htmlTagCss;
    listShapeSubCses.Clear();
    int progressShapeIndex = 1;
    foreach (Shape sh in sld.Shapes)
    {
        sld.Select();
        frm.FormYazdir(string.Format(@"Şekil işleniyor ... {0}/{1}", progressShapeIndex, sld.Shapes.Count));
        var sonuc = (int)(progressShapeIndex * 100) / sld.Shapes.Count;
        frm.pbSurec.Value = sonuc > 100 ? 100 : sonuc;
        progressShapeIndex++;
        H.HtmlNode shapeNode = null;
        if (sh.Type == C.MsoShapeType.msoAutoShape)
        {
            string wmfFileName = string.Format("shape_{0}.wmf", sh.Id);
            string fileName = string.Format(@"{0}\{1}", wmfDir.FullName, wmfFileName);
            float newShapeWidth = sh.Width * Settings.Main.WScaleRatio;
            float newShapeHeight = sh.Height * Settings.Main.HScaleRatio;

```

```

        if (newShapeHeight < 0.5F)
        {
            newShapeHeight = 0.5F;
        }
        try
        {
            if (sh.Connector != C.MsoTriState.msoTrue)
            {
                Shape newShape = sld.Shapes.AddShape(sh.AutoShapeType, sh.Left, sh.Top,
newShapeWidth, newShapeHeight);
                ExportShapeAsWMF(newShape, fileName);
                newShape.Delete();
            }
            else
            {
                ExportShapeAsWMF(sh, fileName);
            }
        }
        catch (Exception)
        {
            sh.Select();
            MessageBox.Show(sh.Name);
        }
        shapeNode = PrepareAutoShapeNode(sh, sld);
        shapeNode.Attributes.Add("wmf", wmfFileName);
    }
    if (sh.Type == C.MsoShapeType.msoPicture || sh.Type == C.MsoShapeType.msoGroup)
    {
        string pictureName = string.Format(@"{0}.png", sh.Name.Replace(" ", ""));
        string picturePath = string.Format(@"{0}\{1}", diSlideImages.FullName, pictureName);
        picturePath = picturePath.Replace(@"\", "/");
        ShapeRange sr = sh.Duplicate();
        Shape shx = sr[1];
        shx.Width = sh.Width * Settings.Main.WScaleRatio;
        shx.Height = sh.Height * Settings.Main.HScaleRatio;
        ExportShapeAsPNG(shx, picturePath);
        shapeNode = PreparePictureShapeNode(sh, sld);
    }
    if (sh.Type == C.MsoShapeType.msoPlaceholder)
    {
        PlaceholderFormat pf = sh.PlaceholderFormat;
        switch (pf.ContainedType)
        {
            case Microsoft.Office.Core.MsoShapeType.msoAutoShape:
                shapeNode = PrepareTextBoxNode(sh, sld);
                break;
            case Microsoft.Office.Core.MsoShapeType.msoCallout:
                break;
            case Microsoft.Office.Core.MsoShapeType.msoCanvas:
                break;
            case Microsoft.Office.Core.MsoShapeType.msoChart:
                break;
            case Microsoft.Office.Core.MsoShapeType.msoComment:
                break;
            case Microsoft.Office.Core.MsoShapeType.msoDiagram:
                break;
            case Microsoft.Office.Core.MsoShapeType.msoEmbeddedOLEObject:
                break;
            case Microsoft.Office.Core.MsoShapeType.msoFormControl:
                break;
            case Microsoft.Office.Core.MsoShapeType.msoFreeform:
                break;
            case Microsoft.Office.Core.MsoShapeType.msoGroup:
                break;
            case Microsoft.Office.Core.MsoShapeType.msoInk:
                break;
            case Microsoft.Office.Core.MsoShapeType.msoInkComment:
                break;
            case Microsoft.Office.Core.MsoShapeType.msoLine:
                break;
            case Microsoft.Office.Core.MsoShapeType.msoLinkedOLEObject:
                break;
            case Microsoft.Office.Core.MsoShapeType.msoLinkedPicture:
                break;
        }
    }

```

```

        case Microsoft.Office.Core.MsoShapeType.msoMedia:
            break;
        case Microsoft.Office.Core.MsoShapeType.msoOLEControlObject:
            break;
        case Microsoft.Office.Core.MsoShapeType.msoPicture:
            string pictureName = string.Format(@"{0}.png", sh.Name.Replace(" ", ""));
            string picturePath = string.Format(@"{0}\{1}", diSlideImages.FullName, pictureName);
            picturePath = picturePath.Replace(@"\", "/");
            ShapeRange sr = sh.Duplicate();
            Shape shx = sr[1];
            shx.Width = sh.Width * Settings.Main.WScaleRatio;
            shx.Height = sh.Height * Settings.Main.HScaleRatio;
            ExportShapeAsPNG(shx, picturePath);
            shapeNode = PreparePictureShapeNode(sh, sld);
            break;
        case Microsoft.Office.Core.MsoShapeType.msoPlaceholder:
            break;
        case Microsoft.Office.Core.MsoShapeType.msoScriptAnchor:
            break;
        case Microsoft.Office.Core.MsoShapeType.msoShapeTypeMixed:
            break;
        case Microsoft.Office.Core.MsoShapeType.msoSlicer:
            break;
        case Microsoft.Office.Core.MsoShapeType.msoSmartArt:
            break;
        case Microsoft.Office.Core.MsoShapeType.msoTable:
            break;
        case Microsoft.Office.Core.MsoShapeType.msoTextBox:
            break;
        case Microsoft.Office.Core.MsoShapeType.msoTextEffect:
            break;
        default:
            break;
    }
}
if (sh.Type == C.MsoShapeType.msoTextBox)
{
    shapeNode = PrepareTextBoxNode(sh, sld);
}
container.AppendChild(shapeNode);
Dictionary<string, string> cssShapeProps = new Dictionary<string, string>();
cssShapeProps.Add("position", "absolute");
cssShapeProps.Add("top", HtmlHelper.PtToCss(sh.Top * Settings.Main.HScaleRatio) + "pt");
cssShapeProps.Add("left", HtmlHelper.PtToCss((sh.Left) * Settings.Main.WScaleRatio) + "pt");
cssShapeProps.Add("width", HtmlHelper.PtToCss(sh.Width * Settings.Main.WScaleRatio) + "pt");
cssShapeProps.Add("height", HtmlHelper.PtToCss(sh.Height * Settings.Main.HScaleRatio) + "pt");
cssShapeProps.Add("display", "table");
Dictionary<string, string> cssShapeComplexProps = new Dictionary<string, string>();
cssShapeComplexProps.Add("-moz-transform", @"rotate(" + Convert.ToInt32(sh.Rotation) + "deg)");
cssShapeComplexProps.Add("-webkit-transform", @"rotate(" + Convert.ToInt32(sh.Rotation) + "deg)");
cssShapeComplexProps.Add("-o-transform", @"rotate(" + Convert.ToInt32(sh.Rotation) + "deg)");
cssShapeComplexProps.Add("transform", @"rotate(" + Convert.ToInt32(sh.Rotation) + "deg)");
string shapeCss = HtmlHelper.PrepareGeneralStyle(cssShapeProps, CssElementType.id,
shapeNode.Attributes["id"].Value);
var complexName = string.Format(@"#{0} > *", shapeNode.Attributes["id"].Value);
shapeCss += HtmlHelper.PrepareGeneralStyle(cssShapeComplexProps, CssElementType.complex, "",
complexName);
foreach (string str in listShapeSubCsses)
{
    slideCss += str;
}
listShapeSubCsses = new List<string>();
slideCss += shapeCss;
shapeIndex++;
}
DirectoryInfo cssDir = Directory.CreateDirectory(string.Format(@"{0}\css", exportDir.FullName));
string cssFilePath = string.Format(@"{0}\{1}-style.css", cssDir.FullName, sld.Name);
File.WriteAllText(cssFilePath, slideCss);
DirectoryInfo jsDir = Directory.CreateDirectory(string.Format(@"{0}\js", exportDir.FullName));
sldDoc.DocumentNode.SelectSingleNode("/link").SetAttributeValue("href", string.Format(@"{0}\{1}-
style.css", cssDir.Name, sld.Name));
WriteSVGtoHTML(sldDoc);
if (Settings.Main.ShowContentTree)
{

```

```

        WriteContentTreeToHtml(sldDoc);
    }
    H.HtmlNode scriptAnimation =
sldDoc.DocumentNode.SelectSingleNode(@"//script[@scripttype=""self""]");
    scriptAnimation.SetAttributeValue("src", @"js\" + sld.Name + @"-animations.js");
    string docName = string.Format(@"{0}\{1}", exportDir.FullName, sld.Name);
    sldDoc.Save(docName + ".html", Encoding.UTF8);
    frm.Close();
}
string strHideClass = @".hide{display:none !important;}";
string strShowClass = @".show{display:normal;}";
string cssKeyframeUrl = string.Format(@"{0}\css\{1}.css", exportDir.FullName, "cssKeyframes");
File.CreateText(cssKeyframeUrl).Close();
StreamWriter swAnimationKeyframes = File.AppendText(cssKeyframeUrl);
swAnimationKeyframes.WriteLine(strHideClass);
swAnimationKeyframes.WriteLine(strShowClass);
swAnimationKeyframes.Close();
shapelIndex = 1;
#region javascript olayları
foreach (Slide s in p.Slides)
{
    Animations.listSlideFunction.Clear();
    string slideHtmlUrl = string.Format(@"{0}\{1}.html", exportDir.FullName, s.Name);
    string slideJsUrl = string.Format(@"{0}\js\{1}-animations.js", exportDir.FullName, s.Name);
    string slideCssUrl = string.Format(@"{0}\css\{1}-style.css", exportDir.FullName, s.Name);
    File.CreateText(slideJsUrl).Close();
    StreamWriter fsJs = File.AppendText(slideJsUrl);
    string strTimeline = "var timeline= new TimeLineObject();";
    fsJs.WriteLine(strTimeline);
    Timeline tLine = s.TimeLine;
    Sequence sMain = tLine.MainSequence;
    for (int i = 1; i <= sMain.Count; i++)
    {
        Effect eff = (Effect)sMain._Index(i);
        Shape sh = eff.Shape;
        string writeContent = "";
        writeContent = Animations.CreatAnimationString(eff, slideCssUrl);
        fsJs.WriteLine(writeContent);
    }
    string startJs = Animations.StartAnimationString();
    fsJs.WriteLine(startJs);
    fsJs.Close();
}
#endregion
var sourcePath = "";
var destinationPath = "";
sourcePath = AppDomain.CurrentDomain.BaseDirectory + @"js\jquery-2.1.0.js";
destinationPath = exportDir.FullName + @"js\jquery-2.1.0.js";
File.Copy(sourcePath, destinationPath, true);
sourcePath = AppDomain.CurrentDomain.BaseDirectory + @"js\jsAnimations.js";
destinationPath = exportDir.FullName + @"js\jsAnimations.js";
File.Copy(sourcePath, destinationPath, true);
sourcePath = AppDomain.CurrentDomain.BaseDirectory + @"css\cssKeyframes.css";
destinationPath = exportDir.FullName + @"css\cssKeyframes.css";
File.Copy(sourcePath, destinationPath, true);
}

private string ExportSlideBackground(Slide sld, string exportPath)
{
    string url = "";
    Slide copySlide = sld.Duplicate()[1];
    while (copySlide.Shapes.Count > 0)
    {
        copySlide.Shapes[1].Delete();
    }
    string imgName = sld.Name.Trim() + ".png";
    string fullPath = string.Format(@"{0}\{1}", exportPath, imgName);
    copySlide.Export(fullPath, "png", Settings.Main.ContentSize.W, Settings.Main.ContentSize.H);
    copySlide.Delete();
    url = string.Format(@"../{0}\{1}\{2}", "img", sld.Name.Trim(), imgName);
    return url;
}

private H.HtmlNode PrepareAutoShapeNode(Shape sp, Slide sld)
{

```

```

H.HtmlDocument hd = new H.HtmlDocument();
H.HtmlNode shapeNode = ShapeHelper.PrepareShapeCover(sp, sld);
string shapeld = shapeNode.Attributes["id"].Value;
H.HtmlNode svgDiv = hd.CreateElement("div");
svgDiv.Attributes.Add("id", "svg" + shapeNode.Attributes["id"].Value);
if (sp.HasTextFrame == C.MsoTriState.msoTrue)
{
    string subCss = "";
    H.HtmlNode divTextFrame = ShapeHelper.PrepareTextFrameNode(sp, shapeld, out subCss);
    shapeNode.AppendChild(divTextFrame);
    listShapeSubCsses.Add(subCss);
}
return shapeNode;
}
private H.HtmlNode PrepareTextBoxNode(Shape sp, Slide sld)
{
    H.HtmlDocument hd = new H.HtmlDocument();
    H.HtmlNode shapeNode = ShapeHelper.PrepareShapeCover(sp, sld);
    string shapeld = shapeNode.Attributes["id"].Value;
    if (sp.HasTextFrame == C.MsoTriState.msoTrue)
    {
        string subCss = "";
        H.HtmlNode divTextFrame = ShapeHelper.PrepareTextFrameNode(sp, shapeld, out subCss);
        shapeNode.AppendChild(divTextFrame);
        listShapeSubCsses.Add(subCss);
    }
    return shapeNode;
}
private H.HtmlNode PreparePictureShapeNode(Shape sp, Slide sld)
{
    H.HtmlNode node = ShapeHelper.PrepareShapeCover(sp, sld);
    H.HtmlDocument doc = new H.HtmlDocument();
    H.HtmlNode imgNode = new H.HtmlNode(H.HtmlNodeType.Element, doc, 0);
    imgNode.Name = "img";
    imgNode.Attributes.Add("id", "img" + sp.Id);
    imgNode.Attributes.Add("src", string.Format(@"{0}\{1}\{2}.png", "img", sld.Name, sp.Name.Replace(" ", "")));
    imgNode.Attributes.Add("width", "100%");
    imgNode.Attributes.Add("height", "100%");
    node.AppendChild(imgNode);
    return node;
}
private void ExportShapeAsWMF(Shape shapeToExport, string pathToExport)
{
    shapeToExport.Select();
    object[] args = new object[] { pathToExport, PpShapeFormat.ppShapeFormatWMF, 0, 0,
PpExportMode.ppClipRelativeToSlide };
    shapeToExport.GetType().InvokeMember("Export", System.Reflection.BindingFlags.InvokeMethod, null,
shapeToExport, args);
}
private void ExportShapeAsPNG(Shape shapeToExport, string pathToExport)
{
    shapeToExport.Select();
    shapeToExport.Rotation = 0F;
    object[] args = new object[] { pathToExport, PpShapeFormat.ppShapeFormatPNG, 0, 0,
PpExportMode.ppRelativeToSlide };
    shapeToExport.GetType().InvokeMember("Export", System.Reflection.BindingFlags.InvokeMethod, null,
shapeToExport, args);
    shapeToExport.Delete();
}
private void WriteSVGtoHTML(H.HtmlDocument slideHTML)
{
    H.HtmlNodeCollection targetNodes = slideHTML.DocumentNode.SelectNodes(@"//div[@wmf]");
    if (targetNodes == null)
    {
        return;
    }
    string baseDir = AppDomain.CurrentDomain.BaseDirectory;
    string exePath = @"SvgExporter\Inkscape.exe";
    Process process = new Process();
    foreach (H.HtmlNode hn in targetNodes)
    {
        string wmfFileName = hn.Attributes["wmf"].Value;
        string fnWmfFilePath = wmfPath + @"\" + wmfFileName;
        string fnSvgFilePath = svgPath + @"\" + wmfFileName.Replace("wmf", "svg");
    }
}

```

```

1000);
    string fullCommand = string.Format(@" inkscape -f ""{0}"" -l ""{1}""", fnWmfFilePath, fnSvgFilePath, 1000,
    ProcessStartInfo psi = new ProcessStartInfo();
    psi.FileName = baseDir + exePath;
    psi.Arguments = fullCommand;
    psi.RedirectStandardInput = true;
    psi.CreateNoWindow = false;
    psi.UseShellExecute = false;
    process.StartInfo = psi;
    process.Start();
    process.WaitForExit();
}
foreach (H.HtmlNode hn in targetNodes)
{
    H.HtmlNode hnSvgDiv = slideHTML.CreateElement("div");
    hnSvgDiv.Attributes.Add("style", "position:absolute;");
    string nodeWmf = hn.Attributes["wmf"].Value;
    string svgName = nodeWmf.Replace("wmf", "svg");
    XmlDocument svgDoc = new XmlDocument();
    svgDoc.Load(svgPath + @"\" + svgName);
    XmlNode svgNode = svgDoc.LastChild;
    H.HtmlTextNode txSvgText = slideHTML.CreateTextNode(svgNode.OuterXml);
    hnSvgDiv.ChildNodes.Prepend(txSvgText);
    hn.ChildNodes.Prepend(hnSvgDiv);
}
}
private void WriteContentTreeToHTMl(H.HtmlDocument slideHTML)
{
    H.HtmlNode tree = HtmlHelper.PrepareContentTree();
    H.HtmlNode ul = tree.OwnerDocument.CreateElement("ul");
    ul.Attributes.Add("style", "list-style-type:none;");
    foreach (Slide s in p.Slides)
    {
        H.HtmlNode link = tree.OwnerDocument.CreateElement("a");
        link.Attributes.Add("target", "_self");
        link.Attributes.Add("href", s.Name + ".html");
        H.HtmlNode text = tree.OwnerDocument.CreateTextNode(s.Name + ".html");
        link.AppendChild(text);
        H.HtmlNode li = tree.OwnerDocument.CreateElement("li");
        li.AppendChild(link);
        ul.AppendChild(li);
    }
    tree.AppendChild(ul);
    H.HtmlNode container = slideHTML.DocumentNode.SelectSingleNode(@"//div[@id='contentWrapper']");
    slideHTML.DocumentNode.InsertBefore(tree, container);
}
private void btnExportAnimations_Click(object sender, RibbonControlEventArgs e)
{
    Array EntryEffectValues = Enum.GetValues(typeof(MsoAnimEffect));
    Presentation p = Globals.ThisAddIn.Application.ActivePresentation;
    string pPath = p.Path;
    string dirName = p.Name.Replace(".pptx", "");
    DirectoryInfo exportDir = Directory.CreateDirectory(pPath + @"\" + dirName);
    FileStream fs = File.Create(exportDir.FullName + "/Effects.txt");
    foreach (var item in EntryEffectValues)
    {
        string satir = string.Format(@"{0}={1}{2}", item, (int)Enum.Parse(typeof(MsoAnimEffect), item.ToString()),
Environment.NewLine);
        fs.Write(enc.GetBytes(satir), 0, enc.GetByteCount(satir));
    }
    fs.Close();
}
private bool IsEntryEffect(Effect e)
{
    if (((int)e.EffectType) >= 1 && ((int)e.EffectType) <= 52 && e.Exit == C.MsoTriState.msoFalse)
    {
        return true;
    }
    return false;
}
private void button1_Click(object sender, RibbonControlEventArgs e)
{
    if (p == null)

```

```

    {
        p = Globals.ThisAddIn.Application.ActivePresentation;
    }
    SectionForm sf = new SectionForm();
    sf.Sunu = p;
    sf.Show();
}
private XmlDocument Scorm12Manifest(Presentation p)
{
    var pName = Path.GetFileNameWithoutExtension(p.Name);
    Sco12 scoHelper = new Sco12();
    var rootDir = Path.GetFileNameWithoutExtension(pName);
    var scormDir = new DirectoryInfo(string.Format(@"{0}\{1}\scorm", presentationPath, rootDir));
    XmlNode nodeManifest = scoHelper.CreateManifest(pName);
    XmlNode nodeMeta = scoHelper.CreateManifestMetadata("ADL SCORM", "1.2");
    nodeManifest.AppendChild(nodeMeta);
    #region Resource
    XmlNode nodeRes = scoHelper.CreateResources();

    XmlNode nodeRes = scoHelper.CreateResource("common_res", ResourceType.webcontent,
ScormObjectType.asset);
    XmlNode nodeFilejs = scoHelper.CreateFile(@"js\jsAnimations.js");
    XmlNode nodeFilejs1 = scoHelper.CreateFile(@"js\jquery-2.1.0.js");
    XmlNode nodeFilejs2 = scoHelper.CreateFile(@"js\scorm.js");
    nodeRes.AppendChild(nodeFilejs);
    nodeRes.AppendChild(nodeFilejs1);
    nodeRes.AppendChild(nodeFilejs2);
    nodeRess.AppendChild(nodeRes);
    foreach (Scoltem si in ThisAddIn.Scoltems)
    {
        if (si.Item == null || si.Type == ScoltemType.Section)
        {
            continue;
        }
        Slide sld = si.Item as Slide;
        var n = scoHelper.CreateResource(sld.Name + "_res", ResourceType.webcontent, ScormObjectType.sco,
sld.Name + ".html");
        var f1 = scoHelper.CreateFile(string.Format(@"css\{0}-style.css", sld.Name));
        var f2 = scoHelper.CreateFile(string.Format(@"js\{0}-animations.js", sld.Name));
        var f3 = scoHelper.CreateFile(string.Format(@"{0}.html", sld.Name));
        n.AppendChild(f1);
        n.AppendChild(f2);
        n.AppendChild(f3);
        var d = scoHelper.CreateDependency("common_res");
        n.AppendChild(d);
        nodeRess.AppendChild(n);
    }
    #endregion

    XmlNode nodeOrg = scoHelper.CreateOrganization(pName + "_org", pName);
    List<XmlNode> sectionList = new List<XmlNode>();
    var listNonSectionedItems = new List<XmlNode>();
    if (ThisAddIn.Scoltems.Count(x => x.Type == ScoltemType.Section) > 0)
    {
        var sections = ThisAddIn.Scoltems.Where(x => x.Type == ScoltemType.Section).ToList();
        for (int i = 1; i <= sections.Count; i++)
        {
            var sectionName = Utility.RemoveSpaceAndNumerics(sections[i - 1].Name);
            var n = scoHelper.CreateItem(sections[i - 1].Index.ToString(), sectionName + "_ref", sectionName);
            sectionList.Add(n);
        }
        var slideScoltems = ThisAddIn.Scoltems.Where(x => x.Type == ScoltemType.Slide).ToList();
        foreach (Scoltem si in slideScoltems)
        {
            var sld = si.Item as Slide;
            var slideName = sld.Name;
            var n = scoHelper.CreateItem(si.Index.ToString(), slideName + "_ref", slideName, slideName + "_res");
            if (si.SectionIndex != -1)
            {
                sectionList.FirstOrDefault(x => x.Attributes["uniq"].Value ==
si.SectionIndex.ToString()).AppendChild(n);
            }
            else
            {

```

```

        listNonSectionedItems.Add(n);
    }
}
foreach (XmlNode xn in sectionList)
{
    nodeOrg.AppendChild(xn);
}
foreach (XmlNode xn in listNonSectionedItems)
{
    nodeOrg.AppendChild(xn);
}
XmlNode nodeOrgs = scoHelper.CreateOrganizations(nodeOrg.Attributes["identifier"].Value);
nodeOrgs.AppendChild(nodeOrg);
nodeManifest.AppendChild(nodeOrgs);
nodeManifest.AppendChild(nodeRess);
return scoHelper.ScoDoc;
}
private void btnScorm2004_Click(object sender, RibbonControlEventArgs e)
{
    MessageBox.Show(btnScorm04.Tag.ToString());
}
private void btnAICC_Click(object sender, RibbonControlEventArgs e)
{
    MessageBox.Show(btnAICC.Tag.ToString());
}
private void btnPublishSettings_Click(object sender, RibbonControlEventArgs e)
{
    HTML5ExportSettingsForm _f = new HTML5ExportSettingsForm();
    _f.ShowIcon = false;
    _f.StartPosition = FormStartPosition.CenterParent;
    _f.ShowDialog(ThisAddIn.NW);
}

private void btnHiyerarsi_Click(object sender, RibbonControlEventArgs e)
{
    try
    {
        p = Globals.ThisAddIn.Application.ActivePresentation;
        if (p.Saved != C.MsoTriState.msoTrue)
        {
            throw new Exception();
        }
    }
    catch
    {
        MessageBox.Show("Lütfen önce kayıtlı bir sunum dosyası açınız veya bu sunum dosyasını kaydediniz.");
        return;
    }
    SectionForm sf = new SectionForm();
    sf.Sunu = p;
    sf.ShowDialog(ThisAddIn.NW);
}
Surec SurecFormuGetir(int slideIndex)
{
    Surec frmSurec = new Surec();
    frmSurec.StartPosition = FormStartPosition.CenterScreen;
    frmSurec.SlideIndex = slideIndex;
    return frmSurec;
}

private void btnScorm12_Click(object sender, RibbonControlEventArgs e)
{
    try
    {
        p = Globals.ThisAddIn.Application.ActivePresentation;
        presentationPath = p.Path;
    }
    catch
    {
        presentationPath = "";
        p = null;
    }
}

```

```

if (p == null)
{
    return;
}
string dirName = Settings.Main.HTML5ExportPath + @"\" + p.Name.Replace(".pptx", "");
DirectoryInfo diSco = Directory.CreateDirectory(dirName + @"\scorm");
DirectoryInfo baseInfo = new DirectoryInfo(dirName);
foreach (DirectoryInfo di in baseInfo.GetDirectories())
{
    if (di.Name.ToLower() == "wmf" || di.Name.ToLower() == "scorm")
    {
        continue;
    }
    DirectoryInfo diNew = Directory.CreateDirectory(diSco.FullName + @"\" + di.Name);
    foreach (FileInfo fi in di.GetFiles())
    {
        File.Copy(fi.FullName, diNew.FullName + @"\" + fi.Name, true);
    }
}
foreach (FileInfo fi in baseInfo.GetFiles("*.html"))
{
    string newHtmlFilePath = diSco.FullName + @"\" + Path.GetFileName(fi.FullName);
    File.Copy(fi.FullName, newHtmlFilePath, true);
    HtmlAgilityPack.HtmlDocument _new = new HtmlAgilityPack.HtmlDocument();
    _new.Load(newHtmlFilePath);

    H.HtmlNode _nodeScorm = new H.HtmlNode(H.HtmlNodeType.Element, _new, 0);
    _nodeScorm.Name = "script";
    _nodeScorm.Attributes.Add("src", "js/scorm.js");
    _new.DocumentNode.InsertBefore(_nodeScorm,
    _new.DocumentNode.SelectSingleNode(@"//div[@id='contentWrapper']"));
    _new.Save(newHtmlFilePath);
}
var listXsd = Directory.GetFiles(AppDomain.CurrentDomain.BaseDirectory + @"\Xsd");
foreach (string path in listXsd)
{
    File.Copy(path, diSco.FullName + @"\" + Path.GetFileName(path), true);
}
string pathScormSource = string.Format(@"{0}\js\scorm12.js", AppDomain.CurrentDomain.BaseDirectory);
string pathScormDestination = string.Format(@"{0}\js\scorm.js", diSco.FullName);
File.Copy(pathScormSource, pathScormDestination, true);
XmlDocument scoDoc = Scorm12Manifest(p);
scoDoc.Save(diSco.FullName + @"\imsmanifest.xml");
}
}
enum CoursePackageType : byte
{
    Scorm12 = 10,
    Scorm2004 = 20,
    AICC = 30
}
}

```

EK-2. Animasyon aktarımı ve zaman çizelgesi oluşturmak için kullanılan "Animation.cs" dosyası.

```
using Microsoft.Office.Interop.PowerPoint;
using System;
using System.Collections.Generic;
using System.IO;

namespace ToHtml5AddIn
{
    public static class Animations
    {
        private static string FolderPath
        {
            get
            {
                string p = "";
                p = (Globals.ThisAddIn.Application.ActivePresentation.Path + @"\" +
Globals.ThisAddIn.Application.ActivePresentation.Name);
                p = p.Replace(Path.GetExtension(p), "");
                return p;
            }
        }

        public static List<string> listSlideFunction = new List<string>();

        public static string CreatAnimationString(Effect eff, string slideCssUrl)
        {
            string strAnim = "";
            string effectObjectName = "shape_" + eff.Shape.Id + "_effect_" + eff.Index;
            string timingName = "effect_" + eff.Index + "_timing";
            string strTiming = "";
            string effectObject = "";
            var strFunc = "";
            var strKeyframe = "";
            var strKeyframeName = "";
            var strMainPart = "";
            Shape _s = null;
            switch (eff.EffectType)
            {
                case MsoAnimEffect.msoAnimEffectAppear:
                    #region msoAnimEffectAppear
                    strTiming = SetTimingObject(timingName, eff.Timing.Duration, eff.Timing.TriggerType.ToString(),
eff.Timing.TriggerDelayTime, eff.Timing.RepeatCount);
                    strAnim += strTiming;
                    strFunc = @"
                        function(){{
                            $(this.Shape).removeClass('hide');
                        }};
                    " + Environment.NewLine;
                    strAnim += SetEffectObject(effectObjectName, "shape" + eff.Shape.Id, eff.Exit ==
Microsoft.Office.Core.MsoTriState.msoTrue ? "true" : "false", eff.Index, timingName, strFunc);
                    strAnim += string.Format(@"{1} timeline.addEffectToQueue({0});{1}", effectObjectName,
Environment.NewLine);
                    #endregion
                    break;
                case MsoAnimEffect.msoAnimEffectAscend://içeri kaydır
                case MsoAnimEffect.msoAnimEffectDescend: // içeri kaydır
                    #region msoAnimEffectDescend
                    _s = eff.Shape;
                    strTiming = SetTimingObject(timingName, eff.Timing.Duration, eff.Timing.TriggerType.ToString(),
eff.Timing.TriggerDelayTime, eff.Timing.RepeatCount);
                    #region keyframe
                    int start = 0;
                    int stop = 0;
                    string keyframeName = "";
                    if (eff.EffectType == MsoAnimEffect.msoAnimEffectAscend)
                    {
```

```

start = (int)_s.Top + 20;
stop = (int)_s.Top;
keyframeName = MsoAnimEffect.msoAnimEffectAscend.ToString();
}
if (eff.EffectType == MsoAnimEffect.msoAnimEffectDescend)
{
start = (int)_s.Top - 20;
stop = (int)_s.Top;
keyframeName = MsoAnimEffect.msoAnimEffectDescend.ToString();
}
keyframeName += "_shape" + _s.Id;
string mainpart = string.Format(@"
    {{
        from {{
            -o-transition : top linear , opacity linear ;
            -webkit-transition : top linear , opacity linear ;
            -moz-transition : top linear , opacity linear ;
            transition : top linear , opacity linear ;
            opacity:0.1;
            top:{0}px;
        }}
        to {{
            -o-transition : top linear , opacity linear ;
            -webkit-transition : top linear , opacity linear ;
            -moz-transition : top linear , opacity linear ;
            transition : top linear , opacity linear ;
            opacity:1;
            top:{1}px;
        }}
    }}" + Environment.NewLine, start, stop);
string keyFrame = string.Format(@"
        @keyframes {0}{1}{2}
        @-webkit-keyframes {0}{1}{2}
        @-moz-keyframes {0}{1}{2}
        @-o-keyframes {0}{1}{2}
        ", keyframeName, mainpart, Environment.NewLine);
File.AppendAllText(slideCssUrl, keyFrame);
#endregion
string css = string.Format(@"
    {0} {{
        -webkit-animation : {1} {2}s {3} linear;
        -o-animation : {1} {2}s {3} linear;
        -moz-animation : {1} {2}s {3} linear;
        animation : {1} {2}s {3} linear;
    }}
    {4}
    ", keyframeName + "_class", keyframeName, (int)(eff.Timing.Duration),
eff.Timing.RepeatCount, Environment.NewLine);
File.AppendAllText(slideCssUrl, css);
strFunc = string.Format(@"
    function(){
        $(this.Shape).removeClass("hide");
        $(this.Shape).addClass("{0}");
    };
    ", keyframeName + "_class") + Environment.NewLine;
effectObject = SetEffectObject(effectObjectName, "shape" + _s.Id, eff.Exit ==
Microsoft.Office.Core.MsoTriState.msoTrue ? "true" : "false", eff.Index, timingName, strFunc);
strAnim += strTiming + Environment.NewLine;
strAnim += effectObject + Environment.NewLine;
strAnim += string.Format(@"{1} timeline.addEffectToQueue({0});{1}", effectObjectName,
Environment.NewLine);
#endregion
break;
case MsoAnimEffect.msoAnimEffectBlinds://perdeler
#region msoAnimEffectBlinds
_s = eff.Shape;
strTiming = SetTimingObject(timingName, eff.Timing.Duration, eff.Timing.TriggerType.ToString(),
eff.Timing.TriggerDelayTime, eff.Timing.RepeatCount);
strKeyframeName = "msoAnimEffectBlinds_shape" + _s.Id;
strMainPart = string.Format(@"{{
    from{{
        height:0px;
    }}
    to{{

```

```

        height:{0}px;
    }}
    }", HtmlHelper.PtToCss(_s.Height / 6));
strKeyframe = string.Format(@"
    @-webkit-keyframes {0}{1}{2}
    @-moz-keyframes {0}{1}{2}
    @-o-keyframes {0}{1}{2}
    @keyframes {0}{1}{2}
    ", strKeyframeName, strMainPart, Environment.NewLine);
File.AppendAllText(slideCssUrl, strKeyframe);
strFunc = string.Format(@"
function(){
    var mainObj=$(this.Shape);
    var overlayDiv=$("<div/>");
    overlayDiv.attr("id","{0}_overlay");
    overlayDiv.css({
        "position": "absolute",
        "top": mainObj.css("top"),
        "left": mainObj.css("left"),
        "width": mainObj.css("width"),
        "height": mainObj.css("height"),
        "z-index": mainObj.css("z-index")+1
    });
    var h = $("#{0}").height() / 6;
    $("#{0}").parent().append(overlayDiv);
    for (var i = 0; i < 6; i++) {
        var fakeDiv = $("<div/>");
        fakeDiv.attr("id", "fake_" + (i + 1));
        fakeDiv.attr("fake", "fake");
        fakeDiv.css({
            "position": "absolute",
            "height": h+"px",
            "width": mainObj.css("width"),
            "overflow": "hidden",
            "z-index": i + 10,
            "top": (i * h) + "px",
            "left": "0px",
            "clip": "rect(0px," + mainObj.css("width") + "px," + h+"px,0px)"
        });
        var mask = $("<div/>");
        mask.attr("mask", "mask");
        mask.css({
            "position": "absolute",
            "top": "0px" + (i * h) + "px",
            "left": "0px"
        });
        mask.html(mainObj.html());
        fakeDiv.append(mask);
        overlayDiv.append(fakeDiv);
    }
    $("div[fake]").css({
        "animation": "{2} linear {1}s"
    })
    setTimeout(function () {
        $("#{0}_overlay").remove();
        mainObj.removeClass("hide");
    }, {1}*999);
}
", "shape" + _s.Id, HtmlHelper.PtToCss(eff.Timing.Duration), strKeyframeName);
effectObject = SetEffectObject(effectObjectName, "shape" + _s.Id, eff.Exit ==
Microsoft.Office.Core.MsoTriState.msoFalse ? "false" : "true", eff.Index, timingName, strFunc);
strAnim += strTiming + Environment.NewLine;
strAnim += effectObject + Environment.NewLine;
strAnim += string.Format(@"{1} timeline.addEffectToQueue({0});{1}", effectObjectName,
Environment.NewLine);
#endregion
break;
case MsoAnimEffect.msoAnimEffectBox:
#region msoAnimEffectBox
_s = eff.Shape;
strMainPart = @"
{
    from{
        -webkit-transform: scale(1);

```

```

        -moz-transform: scale(1);
        -o-transform: scale(1);
        transform: scale(1);
    }
    to{
        -webkit-transform: scale(0.1);
        -moz-transform: scale(0.1);
        -o-transform: scale(0.1);
        transform: scale(0.1);
    }
}
",
strKeyName = "msoAnimEffectBox_shape_" + _s.Id;
strKeyframe = string.Format(@"
    @-moz-keyframes {0}{1}{2}
    @-webkit-keyframes {0}{1}{2}
    @-o-keyframes {0}{1}{2}
    @keyframes {0}{1}{2}
", strKeyName, strMainPart, Environment.NewLine);
File.AppendAllText(slideCssUrl, strKeyframe);
strTiming = SetTimingObject(timingName, eff.Timing.Duration, eff.Timing.TriggerType.ToString(),
eff.Timing.TriggerDelayTime, eff.Timing.RepeatCount);
strFunc = string.Format(@"
function(){
    $(this.Shape).removeClass("hide");
    var obj = $(this.Shape);
    var {1} = $("<div/>");
    {1}.attr("id", "{1}");
    var top = obj.css("top");
    var left = obj.css("left");
    var h = obj.css("height");
    var w = obj.css("width");
    {1}.css({
        "position": "absolute",
        "top": top,
        "left": left,
        "height": h,
        "width": w,
        "background-color": "white",
        "z-index": "100",
        "box-shadow": "0px 0px 15px 15px white"
    });
    obj.parent().append({1});
    {1}.css("animation", "{0} 2s linear");
    {1}.css("webkit-animation", "{0} 2s linear");
    {1}.css("o-animation", "{0} 2s linear");
    {1}.css("animation", "{0} 2s linear");
    setTimeout(function () {{
        {1}.remove();
    }}, 1900);
}
", strKeyName, "shape_" + _s.Id + "_box");
effectObject = SetEffectObject(effectObjectName, "shape_" + _s.Id, eff.Exit ==
Microsoft.Office.Core.MsoTriState.msoTrue ? "true" : "false", eff.Index, timingName, strFunc);
strAnim += strTiming + Environment.NewLine;
strAnim += effectObject + Environment.NewLine;
strAnim += string.Format(@"{1} timeline.addEffectToQueue({0});{1}", effectObjectName,
Environment.NewLine);
#endregion
break;
case MsoAnimEffect.msoAnimEffectVerticalGrow:
#region msoAnimEffectVerticalGrow

_s = eff.Shape;

mainpart = @"{
    0%{{transform: scaleY(1);}}
    50%{{transform: scaleY(1.5);}}
    100%{{transform: scaleY(1);}}
}";
keyframeName = "msoAnimEffectVerticalGrow_shape_" + _s.Id;
keyFrame = string.Format(@"
    @keyframes {0}{1}{2}
    @-moz-keyframes {0}{1}{2}

```

```

        @-webkit-keyframes {0}{1}{2}
        ", keyframeName, mainpart, Environment.NewLine);
File.AppendAllText(slideCssUrl, keyFrame);
strTiming = SetTimingObject(timingName, eff.Timing.Duration, eff.Timing.TriggerType.ToString(),
eff.Timing.TriggerDelayTime, eff.Timing.RepeatCount);
strFunc = string.Format(@"
        function(){
            var obj=$(this.Shape);
            obj.removeClass("hide");
            obj.css("animation","dikeybuyume linear 1s 1");
        }
    ");
effectObject = SetEffectObject(effectObjectName, "shape" + _s.Id, eff.Exit ==
Microsoft.Office.Core.MsoTriState.msoTrue ? "true" : "false", eff.Index, timingName, strFunc);
strAnim += strTiming + Environment.NewLine;
strAnim += effectObject + Environment.NewLine;
strAnim += string.Format(@"{1} timeline.addEffectToQueue({0});{1}", effectObjectName,
Environment.NewLine);
#endregion
break;
default:
break;
}
return strAnim;
}
private static string SetTimingObject(string name, float duration, string trigertype, float delay, int repeateCount
= 1)
{
    var r = "";
    r = string.Format(@"
    var {0}= new TimingObject();
    {0}.Duration={1};
    {0}.RepeatCount={2}; {3} ", name, duration, repeateCount, Environment.NewLine);
    r += string.Format(@"{0}.TriggerType={1}; {2}{0}.TriggerDelayTime={3}; {2}", name, "MsoAnimTriggerType."
+ trigertype, Environment.NewLine, delay);
    return r;
}
private static string SetEffectObject(string name, string shapeid, string exit, int index, string timingname, string
functionstr)
{
    var r = "";
    r = string.Format(@"
    var {0}= new EffectObject();
    {0}.Shape=""#{3}"";
    {0}.Exit={1};
    {0}.Index={2};
    {0}.Timing={5};
    {0}.EffectName=""{0}"";
    {0}.EffectFunction={6};
    {4}", name, exit, index, shapeid, Environment.NewLine, timingname, functionstr);
    return r;
}
public static string StartAnimationString()
{
    return "timeline.startTimeline()";
}
}
}

```

EK-3. Hiyerarşi düzenlemek için kullanılan “SectionForm.cs” dosyası içeriği

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Microsoft.Office.Interop.PowerPoint;
namespace ToHtml5AddIn
{
    public partial class SectionForm : Form
    {
        ScormHelper.Scoltem selectedScoltem = null;
        TreeNode selectedNode = null;
        public Presentation Sunu
        {
            get;
            set;
        }
        public SectionForm()
        {
            InitializeComponent();
        }
        private void SectionForm_Load(object sender, EventArgs e)
        {
            GetScoltems();
            AgacSifirla();
        }
        private void tvHiyer_NodeMouseClick(object sender, TreeNodeMouseClickEventArgs e)
        {
            txtNewName.Text = e.Node.Text;
            selectedScoltem = e.Node.Tag as ScormHelper.Scoltem;
            selectedNode = e.Node;
        }
        private void AgacSifirla()
        {
            tvHiyer.Nodes.Clear();
            if (ThisAddIn.Scoltems.Count == 0)
            {
                MessageBox.Show("Lütfen önce slide ekleyiniz !");
                return;
            }
            var groupedScoltems = ThisAddIn.Scoltems.GroupBy(x => x.SectionIndex).OrderBy(x => x.Key);
            foreach (var g in groupedScoltems)
            {
                var items = g.OrderBy(x => x.Order).ToList();
                foreach (ScormHelper.Scoltem item in items)
                {
                    TreeNode tn = new TreeNode(item.Title);
                    tn.Name = string.Format("tr_{0}_{1}", item.SectionIndex, item.Index);
                    tn.Tag = item;
                    if (item.SectionIndex == -1)
                    {
                        tvHiyer.Nodes.Add(tn);
                    }
                    else
                    {
                        var parentSco = ThisAddIn.Scoltems.FirstOrDefault(x => x.Index == item.SectionIndex);
                        var parentName = string.Format("tr_{0}_{1}", parentSco.SectionIndex, parentSco.Index);
                        tvHiyer.Nodes.Find(parentName, true)[0].Nodes.Add(tn);
                    }
                }
            }
        }
        private void GetScoltems()
        {
            SectionProperties sp = Sunu.SectionProperties;
            if (ThisAddIn.Scoltems.Count > 0)
            {
                return;
            }
        }
    }
}
```

```

    }
    if (sp.Count > 0)
    {
        for (int i = 1; i <= sp.Count; i++)
        {
            ScormHelper.Scoltem si = new ScormHelper.Scoltem()
            {
                Index = i,
                Item = null,
                Type = ScormHelper.ScoltemType.Section,
                Name = sp.Name(i),
                Title = sp.Name(i),
                SectionIndex = -1,
                Order = i
            };
            ThisAddIn.Scoltems.Add(si);
        }
    }
    for (int i = 1; i <= Sunu.Slides.Count; i++)
    {
        Slide sx = Sunu.Slides[i];
        ScormHelper.Scoltem slideSi = new ScormHelper.Scoltem
        {
            Index = sx.SlideID,
            Item = (object)sx,
            Name = sx.Name,
            Title = sx.Name,
            Type = ScormHelper.ScoltemType.Slide,
            SectionIndex = sx.sectionIndex,
            Order = i
        };
        ThisAddIn.Scoltems.Add(slideSi);
    }
}

private void btnAddAfter_Click(object sender, EventArgs e)
{
    if (selectedScoltem == null)
    {
        MessageBox.Show("Lütfen önce sol taraftan bir nesne seçiniz !");
        return;
    }
    if (txtNewName.Text.Trim() == "")
    {
        MessageBox.Show("Lütfen bir metin giriniz!");
        return;
    }
    ScormHelper.Scoltem si = new ScormHelper.Scoltem
    {
        Index = ThisAddIn.Scoltems.Max(x => x.Index) > 500 ? ThisAddIn.Scoltems.Max(x =>
x.Index) + 1 : ThisAddIn.Scoltems.Max(x => x.Index) + 501,
        Item = null,
        Name = txtNewName.Text,
        Title = txtNewName.Text,
        Type = ScormHelper.ScoltemType.Section,
        SectionIndex = selectedScoltem.SectionIndex,
        Order = selectedScoltem.Order + 1
    };
    ThisAddIn.Scoltems.Insert(ThisAddIn.Scoltems.IndexOf(this.selectedScoltem) + 1, si);
    foreach (ScormHelper.Scoltem sii in ThisAddIn.Scoltems.Where(x => x.SectionIndex ==
selectedScoltem.SectionIndex).ToList())
    {
        if (sii.Order <= si.Order)
        {
            continue;
        }
        else
        {
            sii.Order++;
        }
    }
    AgacSifirla();
}

private void btnChangeName_Click(object sender, EventArgs e)
{

```

```

        if (selectedScoltem == null)
        {
            MessageBox.Show("Lütfen önce sol taraftan bir nesne seçiniz !");
            return;
        }
        if (txtNewName.Text.Trim() == "")
        {
            MessageBox.Show("Lütfen bir metin giriniz!");
            return;
        }
        selectedScoltem.Title = txtNewName.Text;
        selectedNode.Text = txtNewName.Text;
        selectedNode.Tag = selectedScoltem;
    }
    private void btnAddBefore_Click(object sender, EventArgs e)
    {
        if (selectedScoltem == null)
        {
            MessageBox.Show("Lütfen önce sol taraftan bir nesne seçiniz !");
            return;
        }
        if (txtNewName.Text.Trim() == "")
        {
            MessageBox.Show("Lütfen bir metin giriniz!");
            return;
        }
        ScormHelper.Scoltem si = new ScormHelper.Scoltem
        {
            Index = ThisAddIn.Scoltems.Max(x => x.Index) > 500 ? ThisAddIn.Scoltems.Max(x => x.Index) + 1 :
            ThisAddIn.Scoltems.Max(x => x.Index) + 501,
            Item = null,
            Name = txtNewName.Text,
            Title = txtNewName.Text,
            Type = ScormHelper.ScoltemType.Section,
            SectionIndex = selectedScoltem.SectionIndex,
            Order = selectedScoltem.Order - 1
        };
        ThisAddIn.Scoltems.Insert(ThisAddIn.Scoltems.IndexOf(this.selectedScoltem), si);
        var ilgiler = ThisAddIn.Scoltems.Where(x => x.SectionIndex == selectedScoltem.SectionIndex).ToList();
        foreach (ScormHelper.Scoltem sii in ilgiler)
        {
            if (ilgiler.IndexOf(sii) <= ilgiler.IndexOf(si))
            {
                continue;
            }
            else
            {
                sii.Order++;
            }
        }
        AgacSifirla();
    }
    private void btnUp_Click(object sender, EventArgs e)
    {
        if (selectedScoltem == null)
        {
            MessageBox.Show("Lütfen önce sol taraftan bir nesne seçiniz !");
            return;
        }
        var ilgiler = ThisAddIn.Scoltems.Where(x => x.SectionIndex == selectedScoltem.SectionIndex).OrderBy(x
=> x.Order).ToList();
        if (ilgiler.Min(x=>x.Order)==selectedScoltem.Order)
        {
            return;
        }
        var onceki = ilgiler[ilgiler.IndexOf(selectedScoltem) - 1];
        var g = onceki.Order;
        onceki.Order = selectedScoltem.Order;
        selectedScoltem.Order = g;
        AgacSifirla();
    }
    private void btnDown_Click(object sender, EventArgs e)
    {
        if (selectedScoltem == null)

```

```

    {
        MessageBox.Show("Lütfen önce sol taraftan bir nesne seçiniz !");
        return;
    }
    var ilgiler = ThisAddIn.Scoltems.Where(x => x.SectionIndex == selectedScoltem.SectionIndex).OrderBy(x
=> x.Order).ToList();
    var sonraki = ilgiler[ilgiler.IndexOf(selectedScoltem) + 1];
    var g = sonraki.Order;
    sonraki.Order = selectedScoltem.Order;
    selectedScoltem.Order = g;
    AgacSifirla();
}
private void btnLevelDown_Click(object sender, EventArgs e)
{
    if (selectedScoltem.SectionIndex == -1 && selectedScoltem.Order == ThisAddIn.Scoltems.Where(x =>
x.SectionIndex == -1).Min(x => x.Order))
    {
        return;
    }
    var ilgiler = ThisAddIn.Scoltems.Where(x => x.SectionIndex == selectedScoltem.SectionIndex).ToList();
    if (selectedScoltem.Order == ilgiler.Min(a => a.Order))
    {
        return;
    }
    var onceki = ilgiler.FirstOrDefault(a => a.Order == selectedScoltem.Order - 1);
    if (onceki.Type == ScormHelper.ScoltemType.Slide)
    {
        return;
    }
    selectedScoltem.SectionIndex = onceki.Index;
    selectedScoltem.Order = ThisAddIn.Scoltems.Where(a => a.SectionIndex == onceki.Index).ToList().Count +
1;
    AgacSifirla();
}
private void btnLevelUp_Click(object sender, EventArgs e)
{
    if (selectedScoltem.SectionIndex == -1)
    {
        return;
    }
    var parent = ThisAddIn.Scoltems.FirstOrDefault(x => x.Index == selectedScoltem.SectionIndex);
    var parentIlgililer = ThisAddIn.Scoltems.Where(x=>x.SectionIndex==parent.SectionIndex);
    selectedScoltem.Order = parentIlgililer.Max(x => x.Order) + 1;
    selectedScoltem.SectionIndex = parent.SectionIndex;
    AgacSifirla();
}
}
}

```