

**T.C.
BALIKESİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MİMARLIK ANABİLİM DALI**

**BİLGİSAYAR DESTEKLİ MİMARİ TASARIM SİSTEMLERİNDEN BİR
HACİM AKUSTİĞİ ANALİZ YAZILIMINA ifcXML İLE VERİ AKTARIMI**

YÜKSEK LİSANS TEZİ

Sibel MACİT

Balıkesir, Haziran-2007

T.C.
BALIKESİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MİMARLIK ANABİLİM DALI

BİLGİSAYAR DESTEKLİ MİMARİ TASARIM SİSTEMLERİNDEN BİR
HACİM AKUSTİĞİ ANALİZ YAZILIMINA ifeXML İLE VERİ AKTARIMI

YÜKSEK LİSANS TEZİ

Sibel MACİT

Tez Danışmanı: Yard. Doç. Dr. Mustafa Emre İLAL

Sınav Tarihi: 29 Haziran 2007

Jüri Üyeleri: Yard. Doç. Dr. Mustafa Emre İLAL (Danışman-BAÜ)

Doç. Dr. Serdar KALE (BAÜ)

Yard. Doç. Dr. Şevket GÜMÜŞTEKİN (İYTE)

Balıkesir, Haziran-2007

ÖZET

BİLGİSAYAR DESTEKLİ MİMARİ TASARIM SİSTEMLERİNDEN BİR HACİM AKUSTİĞİ ANALİZ YAZILIMINA ifcXML İLE VERİ AKTARIMI

Sibel MACİT

Balıkesir Üniversitesi, Fen Bilimleri Enstitüsü,
Mimarlık Anabilim Dalı

(Yüksek Lisans Tezi / Tez Danışmanı : Yrd. Doç. Dr. M. Emre İLAL)

Balıkesir, 2007

Bilgisayar ve iletişim teknolojilerindeki gelişmelerin paralelinde “birlikte işlerlik” kavramı yapım sektöründe kullanılan bilgisayar destekli sistemlerin geliştirilmesine yönelik çalışmaların odak noktası olmuştur. Bilgisayar destekli sistemler arasında birlikte işlerlik kavramının gerçekleştirilmesinde veri tanımlama standartlarının önemli bir rolü vardır. Sunulan bu çalışma kapsamında birlikte işlerliği sağlamaya yönelik geliştirilen IFC standardı ile uyumlu bilgisayar destekli tasarım (BDT) programlarından, bir hacim akustiği analiz aracı olan ESTARA’ya otomatik veri transferi gerçekleştiren bir yazılım geliştirilmiştir. ESTARA öğrencilerin tasarım kararlarıyla hacim akustiği göstergeleri arasındaki etkileşimleri kavramaları için, parametrik analiz çalışmalarını destekleyecek bir araç olarak Balıkesir Üniversitesi’nde geliştirilmektedir. Bu çalışma kapsamında geliştirilen yazılım ile ESTARA için gereken bilgilerin öğrencilerin tasarımlarını oluşturdukları BDT ortamından ifcXML dosyası olarak alınması sağlanmıştır. Bu sayede ESTARA’nın üç boyutlu veri girişi için arayüz ihtiyacı ortadan kalkarken öğrencilerin tanımları gereken arayüz basitleşmiş ve hacim akustiği konularına odaklanmaları kolaylaşmıştır. Bu çalışmayla IFC platformunun, özellikle ifcXML ve Java ile ufak projelerde birlikte işlerliğin sağlanmasında kullanılabilirliğini ortaya koymak amaçlanmıştır.

ANAHTAR SÖZCÜKLER: Birlikte İşlerlik / IFC / ifcXML / Veri Transferi.

ABSTRACT

INTEROPERABILITY BETWEEN COMPUTER AIDED ARCHITECTURAL DESIGN ENVIRONMENTS AND A ROOM ACOUSTICS ANALYSIS APPLICATION USING ifcXML

Sibel MACİT

**Balıkesir University, Institute of Science,
Department of Architecture**

(MSc. Thesis/Supervisor: Asst. Prof. Dr. Mustafa Emre İLAL)

Balıkesir-Turkey, 2007

The interoperability concept in parallel with the developments in computer and communication technologies has become a focus for the studies directed to the development of computer aided systems in architecture, engineering, and construction (AEC) sector. Data description standards have a major role in achieving the desired level of interoperability among computer aided systems. Within the context of this study, interoperability based on the ifcXML standard is explored. Software was developed enabling data transfer from an IFC compliant computer aided architectural design tool to ESTARA (Educational Simulation Tool for Architectural Room Acoustics), a room acoustics analysis tool under development at Balıkesir University. ESTARA aims to provide students with a better understanding of the interactions between room acoustics indicators and design variables by supporting parametric studies. By means of the software developed for this thesis, the required design description for ESTARA is imported from a CAAD environment students are familiar with in the form of an ifcXML file. This makes a graphical user interface for three dimensional input unnecessary for ESTARA and helps students focus more on room acoustics concepts by simplifying the tool. The study seeks to provide information on the workability of the IFC platform especially in smaller development projects, specifically through ifcXML and Java.

KEYWORDS: Interoperability / IFC / ifcXML / Data Transfer.

İÇİNDEKİLER

ÖZET, ANAHTAR SÖZCÜKLER	ii
ABSTRACT, KEY WORDS	iii
İÇİNDEKİLER	iv
ŞEKİL LİSTESİ	vi
KISALTMALAR	vii
1. GİRİŞ	1
1.1 Amaç ve Kapsam	3
1.2 Çalışmanın Organizasyonu	4
2. BİRLİKTE İŞLERLİK, VERİ PAYLAŞIMI ve STANDARTLAR	5
2.1 Bilgisayar Destekli Faaliyetlerin Entegrasyonu	6
2.2 Veri Değişimi ve Sorunları	7
2.3 Geometrik Modelleme	9
2.4 Nesne Yönelimli Modelleme (Ürün Modeli)	10
2.5 Veri Tanımlanması ve Paylaşımı için Standartlar	12
2.5.1 IGES	12
2.5.2 SET	13
2.5.3 VDA-FS	13
2.5.4 PDES	13
2.5.5 DXF	13
2.5.6 STEP	14
2.5.7 IFC	15
2.6 STEP Metodolojisi ve Amacı	15
2.6.1 Tanımlama Metotları	16
2.6.2 Erişim Metotları	17
2.6.3 Uyumluluk Testi Metodolojisi	19
2.6.4 Bütünleşik Çekirdek Kaynaklar	19
2.6.5 Uygulama Protokolleri	19
2.7 IAI - IFC	20
2.7.1 IFC Sürümleri	20
2.7.2 IFC Model Mimarisi	23
2.7.2.1 Kaynak Katmanı	24
2.7.2.2 Çekirdek Katmanı	25
2.7.2.3 Birlikte İşlerlik Katmanı	26
2.7.2.4 Disiplinler Katmanı	26

2.7.3 ifcXML	28
2.8 XML	30
2.8.1 XML Belgelerinin İşlenişi ve XML Teknolojileri	32
2.8.1.1 DOM ve SAX	32
2.8.1.2 DTD	33
2.8.1.3 XML Schema	33
2.8.1.4 XSL	34
3. METODOLOJİ	36
3.1 Teknoloji	36
3.1.1 Standart (IFC/ifcXML)	36
3.1.2 BDT programı (ArchiCAD)	38
3.1.3 Programlama Dili (Java)	39
3.2 ESTARA	40
3.2.1 IFC-Java Kütüphanesinin oluşturulması	41
3.2.2 ifcXML-ESTARA Nesne Modelleri ve Eşleme	42
3.2.3 ESTARA Ayrıştırıcı Modül Bileşenleri	45
3.2.3.1 Ön İşlemci	45
3.2.3.2 Geometri İşlemcisi	46
3.2.3.3 Malzeme İşlemcisi	50
3.2.3.4 Sistem Veritabanı	51
3.2.4 Kısıtlar	52
4. SONUÇLAR ve ÖNERİLER	53
5. KAYNAKLAR	55

ŞEKİL LİSTESİ

<u>Şekil</u>		
<u>Numarası</u>	<u>Adı</u>	<u>Sayfa</u>
Şekil 2.1	Bilgisayar destekli sistemlerin gelişimi	6
Şekil 2.2	Veri değişim yöntemleri	8
Şekil 2.3	Veri modelleme yöntemleri	10
Şekil 2.4	STEP gelişimi	14
Şekil 2.5	EXPRESS Gösterimleri	17
Şekil 2.6	IFC Sürümleri	22
Şekil 2.7	IFC 2x3 platformu	23
Şekil 2.8	IFC model mimarisi	24
Şekil 2.9	IFC veri transfer mekanizmaları	27
Şekil 3.1	ESTARA bileşenlerinin işleyişi	42
Şekil 3.2	ArchiCAD-IFC-ESTARA modelleri	43
Şekil 3.3	IFC-ESTARA nesne eşleme diyagramı	44
Şekil 3.4	ESTARA Parser Modülü	45
Şekil 3.5	findGeometryBuildingElement’de Kullanılan IFC Sınıfları	48
Şekil 3.6	findGeometryFurnishingElement’de Kullanılan IFC Sınıfları	49
Şekil 3.7	findMaterialElement’de Kullanılan IFC Sınıfları	50
Şekil 3.8	ESTARA Veri Dosyası	51

KISALTMALAR

BDM (CAE)	:Bilgisayar Destekli Mühendislik – Computer Aided Engineering
BDT (CAD)	:Bilgisayar Destekli Tasarım – Computer Aided Design
BDÜ (CAM)	:Bilgisayar Destekli Üretim – Computer Aided Manufacturing
DOM	:Document Object Model – Belge Nesne Modeli
DTD	:Document Type Definition – Belge Tipi Tanımı
DXF	:AutoCAD Data Exchange Format – AutoCAD Veri Dönüşüm Formatı
DWG	:AutoCAD DraWing Format – AutoCAD Çizim Formatı
ESTARA	:Educational Simulation Tool for Architectural Room Acoustics – Mimari Oda Akustiği için Eğitimsel Benzetim Aracı
IAI	:International Alliance for Interoperability – Birlikte işlerlik için Uluslararası İşbirliği
IFC	:Industry Foundation Classes – Endüstri Altyapı Sınıfları
IGES	:Initial Graphics Exchange Specification
ISO	:International Organization for Standardization - Uluslararası Standardizasyon Organizasyonu
MSG	:Model Support Group – Model Destek Grubu
ODBC	:Open Database Connectivity - Açık Veritabanı Bağlanırlığı SAX : Simple API for XML – XML için Basit Programlama Arabirimi
SDAI	:Standart Data Access Interface – Standart Veri Erişim Arayüzü
SET	:Standard d'Exchange et de Transfer
SPF	:STEP Physical File - STEP Fiziksel Dosya
STEP	:Standards for the Exchange of Product model data – Ürün Modeli Verilerinin Değişimi için Standart
VDAFS	:Verband der Deytschen Automobilindustrie – Flachenschnittstelle
XML	:Extensible Markup Language – Genişletilebilir İşaretleme Dili
XSD	:XML Schema Definition – XML Şema Tanımı
XSL	:The Extensible Stylesheet Language - Genişletilebilir Biçimlendirme Dili

1. GİRİŞ

Teknolojinin gelişmesi ile birlikte yapım sektöründe bilgisayar destekli sistemlerin kullanımı artmakta; bu sistemler verimlilik ve ürün kalitesini artırmaya, tasarım ve üretim için harcanan zaman ve maliyeti düşürmeye yönelik olarak sürekli gelişmektedir. Fakat bu sistemlerin birbirinden bağımsız olarak geliştirilmesi ve her birinin kendi modelleme tekniklerini ve veri taban yapısını kullanması, bir bilgisayar destekli sistemin veritabanındaki veya çıktı dosyasındaki veriyi diğer bir bilgisayar destekli sistemin okuyamaması anlamına gelmektedir. Bu durum bilgisayar destekli sistemlerin kendilerinden beklenen performansı yerine getirememeleri sorununu ortaya çıkarmaktadır.

Bilgisayar destekli sistemler arasındaki farklılık ve bu farklılıktan doğan sorunlar, çok sayıda katılımcı ve disiplinin ortaklaşa çalıştığı yapı sektöründe kullanılan bilgisayar destekli sistemler arasında veri paylaşımı ve değişiminin sağlanmasının gerekliliğini ortaya çıkartmıştır. Bu gereksinimi karşılamak amacıyla 1980'li yıllarda bilgisayar destekli sistemler arasında birlikte işlerliği sağlamak için standardizasyon çalışmaları başlatılmıştır [1-5]. Bu çalışmalar sonucunda günümüze kadar bir çok veri paylaşım standardı (IGES, SET, VDA-FS, STEP, DXF, DWG, IFC) geliştirilmiştir. Veri paylaşım standartları, bina çizim veya modeline ait geometrik ve/veya topolojik bilgileri belirli bir formatta kaydeden ve farklı bilgisayar destekli sistemler arasında iletişimi sağlayan yapılar olarak tanımlanabilmektedir. Geliştirilen bu standartlardan STEP ve IFC geometri ile birlikte ürün bilgilerini (malzeme, maliyet, süre vb. yapı oluşum döngüsünü destekleyen proje bilgileri) de içeren nesne tabanlı standartlardır, diğer standartlar sadece geometri bilgisini içermektedir.

IFC (Industry Foundation Classes), yapım sektöründeki farklı bilgisayar destekli sistemler arasında veri paylaşım ve transferi için geliştirilen en geniş çaplı veri paylaşım standardıdır. Uluslar arası düzeyde geliştirilmeye devam edilen IFC, yapım sektöründe ki lider yazılım şirketlerinin (Autodesk, Graphisoft, Bentley vb.)

birçoğu tarafından desteklenmekte ve bu şirketlerin ürettiği bilgisayar destekli sistemlerde kullanılmaktadır [6]. Yapım sektöründe birlikte işlerliği amaçlayan dünya çapındaki bir çok çalışmada da IFC standartlarının kullanıldığı görülmektedir. Finlandiya Ulusal Teknoloji Kurumunun Stanford Üniversitesi'nden Fischer ve Kam ile beraber yürüttüğü Ürün Modeli ve Dördüncü Boyut (Product Model and Fourth Dimension – PM4D) araştırmasında IFC kullanılmıştır [7]. Bu çalışmada Helsinki Teknoloji Üniversitesi'nde bir oditoryumun tasarım ve yapımı PM4D yaklaşımıyla yürütülmüştür. Almanya'da Nytsch-Geusen yürütücülüğündeki ekip tarafından bir BDT sistemi ile Bilgisayar Destekli Akışkanlar Dinamiği (CFD) programı arasında veri transferi sağlayan araç geliştirilmiştir [8]. Singapur'da yapı projelerinin yönetmeliğe uygunluğunun otomatik kontrolünün ulusal uygulaması için doğrudan bağlantılı geliştirilen sistemde IFC uyumlu modeller kullanılmıştır [9]. Geliştirilen bu sistemde, BDT programından alınan IFC dosya biçimindeki proje bilgisi ile IFC uyumlu yapım yönetim modelleri arasında uyum kontrolü otomatik olarak yapılabilmektedir. İrlanda'da O'Sullivan ve Keane bina enerji simülasyonu için IFC tabanlı bir arabirim geliştirmişlerdir [10]. Bu arabirim, Bilgisayar Destekli Tasarım (BDT) ortamından alınan IFC dosya biçimindeki proje bilgisini enerji simülasyon programının kullandığı IDF dosya biçimine dönüştürmektedir.

Bu tez kapsamında IFC standardının, BDT ortamında oluşturulan tasarım verisinin farklı disiplinlerdeki bilgisayar programları ile paylaşımı ve dönüşümünde kullanılabilirliği üzerinde çalışılmıştır. Çalışmanın sonunda, bilgisayar destekli mimari tasarım aracı olan ArchiCAD programından hacim akustiği analiz aracı olarak geliştirilen ESTARA programına otomatik veri transferi gerçekleştirilmiştir. Java ortamında geliştirilen ESTARA programında veri transferi IFC standardı kullanılarak sağlanmıştır.

Yapı sektörüne yönelik nesne modeli verisinin temsil ve paylaşımının standardizasyonunu sağlamak amacıyla oluşturulan IFC, benzer paralellikte olan ve birçok sektörde başarı ile kullanılan STEP standardının mevcut alt yapısından yararlanmaktadır. Farklı bilgisayar destekli sistemler arasında veri transferi için STEP mekanizmalarını kullanan IFC, verinin dosya yoluyla transferinde bu mekanizmalara alternatif olarak ifcXML dosya biçimini sunmaktadır [11]. ifcXML

ile veriler, dünyada çok yaygın kabul gören ve standart bir dil olarak kabul edilen XML tabanlı dosyalarda tutulmaktadır. XML standardını destekleyici ve tamamlayıcı bir çok teknolojiye sahip olan nesne tabanlı programlama dili Java ortamında geliştirilen ESTARA programında da veri dönüşüm formatı olarak ifcXML kullanılmıştır.

1.1 Amaç ve Kapsam

Çok katılımcılı yapı sektöründe çeşitli bilgisayar programları kullanılmaktadır ve bu programların birbirleri ile veri alışverişinde bulunması beklenmektedir. Bu çalışma kapsamında, yapı sektöründe merkezi bir faaliyet olan mimari tasarım ve tasarım karar destek sistemlerinden biri olan akustik performans analizi programları arasındaki veri alışverişi dikkate alınmıştır.

Bu tez çalışmasının amacı BDT programından ESTARA programına veri transferi sağlayarak, ESTARA programının veri girişini otomatikleştirmek ve bu programı mimarlık öğrencileri tarafından kullanılabilir yapmaktır. Bu amacı gerçekleştirmek için yapılan çalışmada BDT ortamı olarak ArchiCAD, veri değişim formatı olarak ifcXML ve programlama dili olarak da JAVA kullanılmıştır.

Tez kapsamında, yapı sektöründe kullanılan bilgisayar destekli sistemler arasında veri paylaşımının temel sorunları ve çözüm arayışları incelenmiş, veri paylaşımında uluslar arası düzeyde kullanılan ve birlikte işlerliği gerçekleştirme çabalarında anahtar rol oynayan STEP ve IFC standartları ile XML hakkında detaylı araştırma yapılmıştır. Bu bilgiler ışığı altında kullanılacak yöntem ve araçlar belirlenerek geliştirilen yazılım ile ArchiCAD programından ESTARA programına otomatik veri transferi gerçekleştirilmiştir.

1.2 Çalışmanın Organizasyonu

Mimari tasarım ve modelleme aracı olarak kullanılan ArchiCAD programından hacim akustiği performansı analiz programı olarak geliştirilen ESTARA'ya otomatik veri transferi sağlamayı amaçlayan bu çalışma 4 ana bölümden oluşmaktadır. Bölüm 1, bu akademik çalışmanın gerçekleştirilmesinin arkasında yatan motivasyonel faktörleri, çalışmanın önemini, kapsamını ve amaçlarını içermektedir.

Bölüm 2'de birlikte işlerlik kavramından bahsedilerek bilgisayar destekli sistemler arasında veri paylaşımının önemi ve temel sorunları irdelenmiştir. Birlikte işlerliği sağlamanın çözüm arayışlarında anahtar rol oynayan, veri temsili ve paylaşımı için geliştirilmiş temel standartlar gözden geçirilmiştir.

Bölüm 3'de ise bu çalışma kapsamında geliştirilen ESTARA programının oluşturulmasında kullanılan teknolojilerden, bu teknolojilerin tercih edilme sebeplerinden bahsedilmiş ve programın yapısı hakkında bilgi verilmiştir.

Son olarak, dördüncü bölümde yapılan çalışmanın genel sonuçları özetlenmiş ve gelecek çalışmalar için öneriler sunulmuştur.

2. BİRLİKTE İŞLERLİK, VERİ PAYLAŞIMI VE STANDARTLAR

Yapı üretimi, giderek çok sayıda uzmanlık gerektiren alt üretim alanlarına (tasarım öncesi, tasarım, yapım, yönetim) bölünmüştür. Her bir alanda belirli konularda uzmanlaşmış çok sayıda katılımcı yer almaktadır. Bu parçalanmış yapı, her bir katılımcının kendi amaçlarına yoğunlaşırken, ana amaç olan binanın istenen süre, maliyet ve kalite koşullarında tamamlanması açısından olumsuzluklara neden olmaktadır. Fiziksel olarak bütünleştirilmesi mümkün olmayan bu parçalanmışlığı sanal olarak bütünleştirmek için birbiriyle uyumlu, birlikte çalışabilir, etkileşimli, izlenebilir ve denetlenebilir bilgi sistemlerine ihtiyaç duyulmaktadır. Bilginin katılımcılar arasında ve bilgi sistemlerinde kullanılabilme, paylaşılabilme ve transfer edilebilme yeteneği olarak açıklanabilen birlikte işlerliğin en geniş kapsamdaki tanımı, etkin bilgi paylaşımıdır.

Yazılım alanında birlikte işlerlik, farklı programların belirli iş süreçlerini kullanarak bilgi alışverişinde bulunabilmesi, aynı dosyaları okuyup yazabilmesi ve aynı protokolleri kullanabilmesidir. Yapı sektöründe birlikte işlerlik ise; yapı oluşum sürecinin bütün aşamalarının, bu aşamalarda rol alan disiplinlerin ve bu disiplinlerde kullanılan bilgisayar destekli sistemlerin, kendilerine düşen görevleri yerine getirebilmeleri için bilgiyi birbirleri arasında paylaşabilmeleridir.

Günümüzde yapı sektöründe kullanılan yazılımların birçoğu (Autodesk Arcitectoral Desktop, Graphisoft ArchiCAD, Autodesk Revit Building, Bentley Architecture, Nemetchek Allplan, vb.) yapı elemanlarını ve yapım eylemlerini temsil eden ürün modeli mantığında çalışmaktadır. Ancak bu yazılımların birbirinden bağımsız olarak geliştirilmesi ve her birinin ürün modeli tanımlamalarında kendi standartlarını kullanması sistemler arası veri paylaşımına engel oluşturmaktadır. Birlikte işlerliğin sağlanabilmesi için bu ürün modeli tanımlamalarının standart bir çatı altında toplanması gerekmektedir.

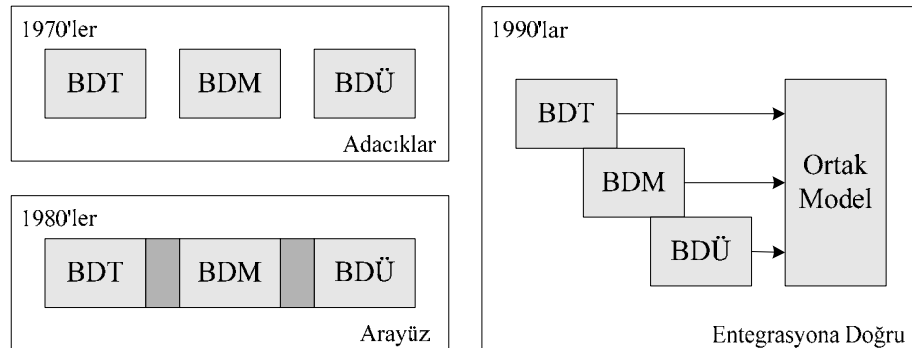
Bu bölümde, yapım sektöründe birlikte işlerliği sağlamaya yönelik olarak bilgisayar destekli faaliyetlerin entegrasyonunun tarihsel gelişimi, veri paylaşımının

önemi, temel sorunları, geometrik modelleme, nesne(ürün) modelleme, ve ürün model verilerinin değişimi için var olan standartlar hakkında bilgi verilecektir.

2.1 Bilgisayar Destekli Faaliyetlerin Entegrasyonu

Günümüzde, yapım sektöründe verimliliğin, ürünün kalitesinin artırılmasına ve tasarım ve üretim için harcanan zamanın ve maliyetin azaltılmasına yönelik çok sayıda bilgisayar destekli tasarım (BDT), üretim (BDÜ) ve mühendislik (BDM) programları bulunmaktadır. Bu bilgisayar programları arasında iletişimin sağlanamadığı ve veri paylaşımı gerçekleştirilemediği sürece bu programların potansiyellerinden tam olarak yararlanmak mümkün değildir.

1970’li yıllarda BDT,BDM ve BDÜ gibi sistemler birbirlerinden bağımsız olarak geliştirildiler. Bu sistemlerin birbirlerinden kopukluğu “otomasyon adaları” olarak anılan bilginin adacıklar halinde öbeklenmesi denilen bir problemi oluşturmuştur. Gerek donanım ve yazılım platformu, gerekse uygulama açısından bakıldığında birbirinden kopuk olarak çalışan her sistem birer "ada" diye nitelendirilirdi [12,13]. 1980’li yıllarda ise çalışmalar bu sistemler arasında arayüzler oluşturma doğrultusunda olmuştur [14,15]. 1990’lı yıllarda ise uygulamaların, dolayısıyla bilgisayarların birbirine entegrasyonunun önem kazanmaya başlamasıyla artık amaç bu sistemleri bütünleştirmek olmuştur ve bu amaç ortak model yaklaşımını doğurmuştur [16-22].



Şekil 2.1 Bilgisayar destekli sistemlerin gelişimi

Bilgisayar destekli sistemlerin gelişimindeki bu paralellikle birlikte elektronik veri değişimi kavramı giderek önem kazanmaktadır. Yapı tasarım ve üretiminde hızlı, kaliteli ve ekonomik çözümler üretilebilmesi için yapım sektöründeki tüm aşamalarda tüm katılımcılar arasında veri değişiminin tutarlı, hızlı ve etkin olması gerekmektedir.

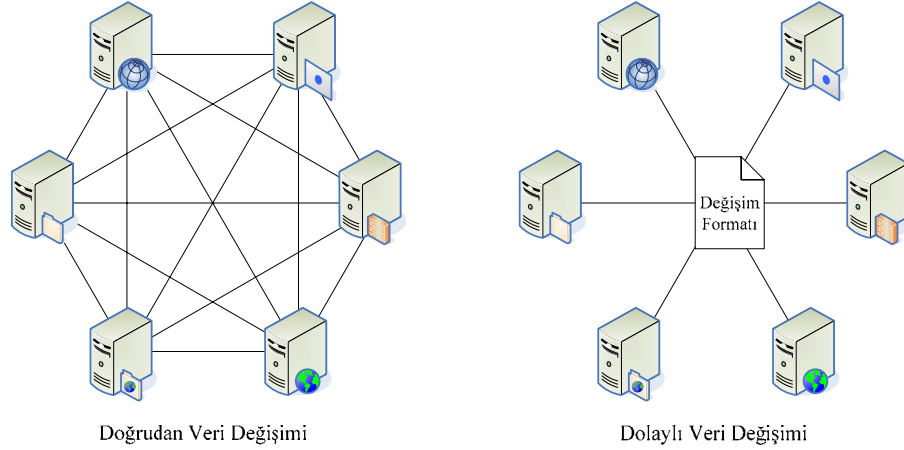
2.2 Veri Değişimi ve Sorunları

Veri değişimi, bir bilgisayar destekli sistemden diğer bir bilgisayar destekli sisteme tekrar veri girişi yapılmaksızın transfer edilebilen, yapısal, bilgisayar tarafından işlenebilir formattaki bilgilerin elektronik olarak transferidir.

Genel olarak, veri değişimi iki şekilde gerçekleştirilebilir. Bunlar “Doğrudan” ve “Dolaylı” yöntemlerdir. Dolaylı yöntemde iki format arasındaki çeviri bir “ara format” üzerinden gerçekleştirilir. Sunucu verisi önce ara formata dönüştürülür, istemci ara formatta veriyi transfer eder ve kendi formatına dönüştürür. Buradaki ara format çoğunlukla “Değişim Formatı” olarak anılır. Doğrudan yöntemde ise, iki format arasında doğrudan bir çeviri uygulanır. Doğrudan yöntemde çeviri yalnızca iki formata yönelik olduğundan, bilgi kaybı dolaylı yöntemde göre daha azdır [23].

Doğrudan yöntem, çok sayıda sistemin veri değişiminde bulunabilmesi için gerekli çevirici sayısı ve çeviricilerin yenilenmesi yada yeni çeviricilerin eklenmesi bakımından sorunludur. Şekil 2.2’de mevcut sistemlere, yeni bir sistem eklenmesi durumunda yeniden geliştirilmesi gerekli çevirici sayısı dolaylı yöntemde yalnızca bir iken, doğrudan yöntemde mevcut sistem sayısı kadardır. Çevirici geliştirme, her iki formatın da çok iyi bilinmesini gerektirdiğinden, oldukça zaman alıcı bir işlemdir. O nedenle, doğrudan çevirici geliştirme çoğu zaman eldeki proje süresi açısından geçerli bir çözüm olmayacaktır. Dolayısıyla, veri değişiminde geleneksel olarak daha yaygın kullanılan yöntem, dolaylı yöntem olmuştur [23]. Ancak yapı sektöründe veri kullanıcılarının çeşitliliği, bütün kullanıcı gruplarının gereksinimlerini karşılayacak bir değişim formatı tasarımını çok güç, belki de olanaksız yapmıştır. Dahası, tasarımcılarına göre “çok iyi” tasarlanmış olması,

değişim formatının standartlaşmasını garanti etmemiştir. Dolayısıyla, bugüne kadar SET, IGES, DXF, DWG, STEP, IFC vb. pek çok değişim formatı geliştirilmiş olmasına rağmen bugün veri değişimi için tek bir standart mevcut değildir.



Şekil 2.2 Veri değişim yöntemleri

Veri değişiminde karşılaşılan temel sorunlar verinin çeşitli bilgisayar programlarında farklı içerik ve formatla temsil edilmesidir [24]. İçerik farklılıklarından biri “duvar” tanımında yer alan “duvarın malzeme bilgisinin”, diğer bir tanımda bulunmaması gibi proje modelinin tanımlandığı bilgi kapsamıdır. Proje modelinin yalnızca geometrik ya da geometrik ve topolojik olarak tanımlanması da içerik farklılıkları olarak görülmektedir. Format farklılığına sebep ise verinin modelleme yöntemidir. Birbirinden bağımsız geliştirilen her bilgisayar programı kendi modelleme yöntemlerini kullanmaktadırlar. Çağdaş BDT sistemleri “Nesne Tabanlı” modelleme yönteminde geliştirilmişlerdir. Aşağıdaki bölümlerde Nesne Tabanlı Modelleme yöntemi ile bu tür bir modellemenin gerekliliğine götüren ve nesne tabanlı veri modellemesinin bir önceki aşaması olan Geometrik Modelleme yöntemi anlatılmıştır.

2.3 Geometrik Modelleme

Geometrik modelleme terimi, bir nesnenin geometrik özelliklerinin tam bir matematiksel tanımlamasını yapabilecek metotların bir araya getirilmesi olarak açıklanmaktadır. Geometrik modelleme sistemleri tarihsel gelişimine göre (1) tel kafes modelleme sistemleri, (2) yüzey modelleme sistemleri ve (3) katı modelleme sistemleri olarak sınıflandırılır [25].

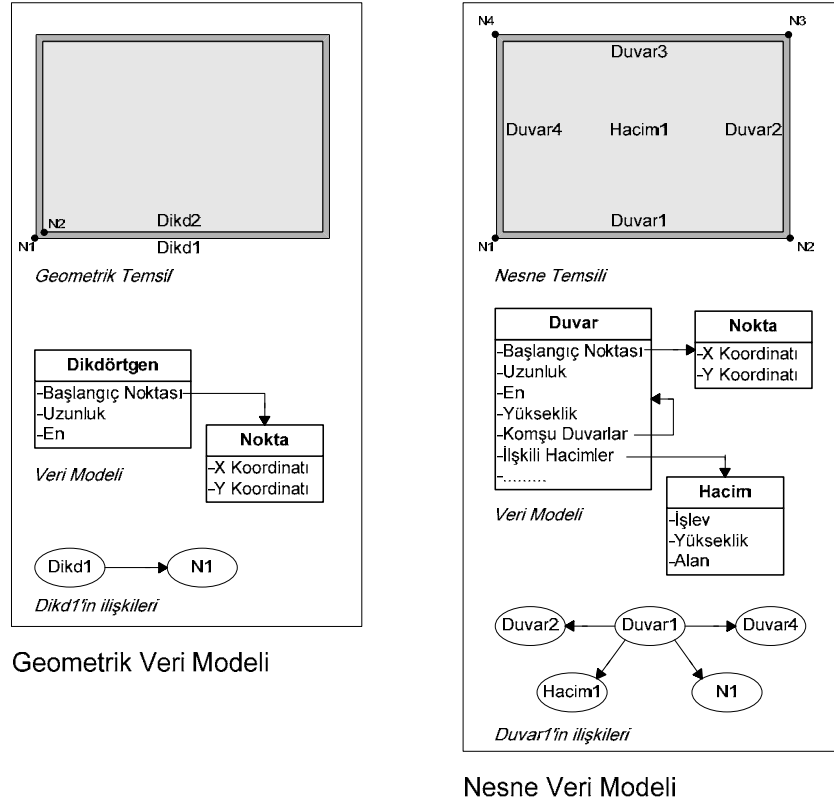
Tel kafes modelleme sistemleri bir modeli onun çizgileri ve son noktaları vasıtasıyla gösterir. Geleneksel teknik resim çizimlerinin bilgisayarlaştırılmış hali olarak görülebilir. Tel kafes modelleme en eski 2 ve 3 boyutlu geometrik modelleme yöntemidir.

Yüzey modelleme sistemleri yüzey, kenar ve köşelerin tanımını içerir. Tel Kafes model içinde yer alan bilgileri sağladığı gibi yüzeylerin bağlantısı hakkında da bilgi sağlar.

Katı (solid) modelleme sistemleri bir modeli kapalı bir hacim olarak modellemek için kullanılır. Tel kafes örgü ve yüzey modelleme sistemlerinden farklı olarak eğer kapalı bir hacim olarak biçim verilemezse, basit yüzeylerin veya karakteristik çizgilerin oluşmasına izin verilmez. Yüzey modelleme sistemlerinin sağladığı bilgiye ek olarak katı modelleme sistemiyle oluşturulmuş bir şeklin matematiksel tanımı herhangi bir yerin kapalı hacmin içinde, dışında veya üzerinde bir yerlerde olup olmadığını tanımlayan bilgiyi de (topolojik bilgi) içerir. Bu yüzden katı modelin hacmiyle ilgili herhangi bir bilgi türetilebilir, ve böylece uygulama programları yüzey düzeyinde işlem yapmak yerine hacim düzeyinde işlemler yapmak için yazılabilir [26].

Geometrik modelleme de yalnızca geometri ve topolojiyi içeren bilgi modellenmektedir. Oysa parçalı ve karmaşık bir yapıya sahip olan yapım sektörünün farklı aşamalarında kullanılan çeşitli bilgisayar destekli sistemler (BDT, BDÜ, BDM) boyutlar, üretim işlemleri, dayanım gücü, ısıl özellik, malzeme bilgisi ve yüzey özellikleri gibi bilgi ile ilgilenirler. Bu sistemler arasında bütünleşmenin

sağlanması ve yapı üretimi ile ilgili uygulamaların otomatikleştirilmesi ihtiyacını karşılamak için BDT programları nesne tabanlı modelleme tekniğine yönelmişlerdir. Şekil 2.3'de dikdörtgen biçimindeki bir odanın geometrik ve nesne veri modeli gösterilmiştir.



Şekil 2.3 Veri modelleme yöntemleri

2.4 Nesne Tabanlı Modelleme (Ürün Modeli)

Nesne tabanlı modelleme, nesnelere şeklinde ayrıştırılan bir sistemde bilginin modellenmesini sağlayan bir tekniktir. Bu modelleme tekniğinde, bağımsız olarak oluşturulan nesnelere bir kimlik, bir durum ve bir davranış belirtirler. Nesne tabanlı modelleme sistemlerinde, sistemin parçaları, kendi "bilgilerini (veri)" ve "davranışlarını (metotlar)" içeren, nesnelere dönüştürülürler [27].

Her nesne kendine ait bilgileri içerir. Her nesnenin kendi bilgisini diğer nesnelere bir çerçeve ile ayırması "kapsülleme" olarak adlandırılmaktadır.

Kapsülleme sayesinde sistemin her nesnesine diğer nesnelere bağımsız olarak müdahale edilebilir. Bu durum nesnelere güncellenmesine olanak sağlar. Sistemin parçaları olan nesnelere arasındaki iletişim ve etkileşim ise "mesajlar" ile yapılır. Sistemin nesnelere alt sınıf/üst sınıf hiyerarşisi içinde kalıtılarak, bir sınıf şeklinde oluşturulurlar [28].

Örneğin; bir binayı ele alırsak; bu binanın duvarları bir nesnedir. Döşemeleri, girişleri, kapıları, pencereleri ayrı birer nesnelere. Duvarın yüksekliği, rengi, malzemesi vs, duvarın özellikleri olarak verisini; üretim zamanı, dayanım gücü gibi hareketleri onun davranışlarını karakterize ederler. Bu şekilde bir duvar kapsülendirilerek duvar nesnesi oluşturulur ve kendi bilgisine sahip olur. Ayrıca duvar nesnesi diğer nesnelere iletişim ve etkileşim için mesajlara da sahiptir.

Nesne tabanlı modelleme tekniğinde amaç geometrik ve geometrik olmayan özellikler arasındaki parametrik ilişkileri bütünleştirmektir [29]. Geometrik veri nesnelere 3 boyutta katı veya yüzey olarak tanımlarını içerirken, geometrik olmayan veri ise nesnelere malzemesini, ağırlığını, dayanım gücünü, ısısal özelliği gibi tanımları kapsamaktadır.

Çağdaş BDT programları ürün modeli mantığı üzerine kurulmuşlardır. Nesne tabanlı modelleme ile geliştirilen BDT programları kullanıcılara sadece geometrik ve topolojik bilgi sağlamak yerine bir ürün modeli sunar. Ürün modelleri üretilecek elemana (giriş,kolon,duvar) ait boyut, dayanım, malzeme, maddi değer, üretim sıra ve zamanı ile ilgili bilgileri taşıyan sayısal modellerdir. Proje modeli ise yapıya ait tüm ürün modellerinin birleşmesi ile ortaya çıkan projeye ait yukarıda sayılan tüm değerleri içeren sayısal yapı modelidir. Proje modelleri nesne-yönelimli yaklaşımla oluşturulmaktadır [30].

Bütünleştirilmiş bir yapı tasarım ve üretim ortamının oluşturulmasında bilgisayar destekli sistemler arasında veri paylaşımı için, ürün modellemenin yapılması kaçınılmazdır [31]. Nesne yönelimli ürün modelleme üzerinde araştırmalar sürmekte ve bütünleştirilmiş bir yapı tasarım ve üretim sistemlerinin oluşturulması için çözüm olacak bir yaklaşım olarak düşünülmektedir. Ancak

günümüzde kullanılan modeller analiz, tasarım vb. aşamalar için ayrı ayrı üretilmekte ve bu modellerin birbirleri ile bütünleştirilmesi güçleşmektedir. Bu duruma çözüm olarak tüm dünyada veri tanımına ve değişimine ait standartlar belirlenmektedir.

2.5 Veri Tanımlanması ve Paylaşımı için Standartlar

Verilerin elektronik olarak çizim sistemleri arasında transfer edilmesinin gerekliliği 1970’li yılların başlarında ortaya konmuştur. Bu konudaki ilk girişim veri değişimine baz olarak alınabilecek bir dosya formatı standardının geliştirilmesi olmuştur. 1970’li yılların sonlarına doğru ise, veri transferinin sadece 2 boyutlu çizim sistemleri arasında değil 3 boyutlu modelleme sistemleri arasında da olması gerektiği ortaya konmuş ve katı modelleme verileri için bir dosya şartnamesi hazırlanmıştır [32].

Bu bölümde, veri değişimi için geliştirilen ve en çok kullanılan dosya formatları ile standartlar hakkında bilgi verilmektedir.

2.5.1. IGES

IGES(Initial Graphics Exchange Specification) İlk veri değişim standardıdır. Bu standart bilgisayarlı çizim sistemlerinin veri tabanları arasında veri değişimini sağlamaya yönelik geliştirilmiştir. Geliştirildiği ilk yıllarda IGES’in kapsam alanı, ayrıntılı bir çizimin gerek duyduğu tipik geometrik ve grafiksel gereksinimlerle sınırlıydı. Bugün ise, IGES katı modelleme sistemleri arasındaki veri değişimlerini de kapsamaktadır [33]

2.5.2 SET

SET(Standard d'Exchange et de Transfer) standardı, Fransız Aerospatiale firması tarafından, IGES dosyalarının çok büyük olmasından ve kendi BDT sistemlerine ait verilerin tamamının IGES aracılığıyla transferinin yapılamamasından dolayı geliştirilmiştir. Bu standart her ne kadar IGES'i baz almışsa da farklı bir dosya formatına sahiptir ve IGES ile kıyaslandığında dosya büyüklüğü oldukça küçüktür [3].

2.5.3 VDA-FS

VDA-FS (Verband der Deytschen Automobilindustrie - Flachenschnittstelle) standardı, IGES'in serbest formdaki yüzey datalarının değişimi konusundaki yetersizliği sebebiyle, otomobil üreten Alman firmalarının ulusal, uluslararası, ve yan sanayi ile olan veri değişim işlemlerini sağlamak amacıyla geliştirilmiştir. Fakat VDA-FS, BDT işlemlerinin sadece çok dar bir alanına hitap etmektedir [2].

2.5.4 PDES

PDES (Product Data Exchange Standard) ürün veri değişim standardı, 1983 yılında, IGES organizasyonu tarafından geliştirilmiştir. Bu standart IGES'in dosya büyüklüğünün ve bilgisayar işlem zamanının çok fazla olması ve iş idaresi bilgisi, üretim süreç bilgisi gibi bilgilerin transferinden çok sadece veri transferi üzerinde yoğunlaşmış olması gibi bazı olumsuzlukları gidermek amacıyla hazırlanmıştır [4].

2.5.5 DXF ve DWG

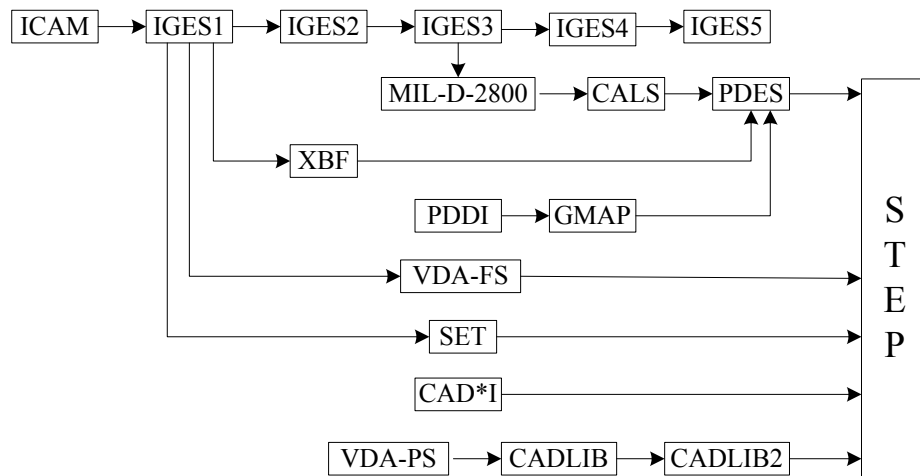
DXF(Drawing eXchange Format) ve DWG(DraWinG), Autodesk firması tarafından geliştirilen Bilgisayar destekli tasarım programı AutoCAD'in dosya formatıdır. 2 ve 3 boyutlu basit çizimleri taşımak için kullanılan en yaygın dosya

formatıdır. Birçok rakip firma da AutoCAD'in piyasa hakimiyetinden dolayı DXF ve DWG dosya formatı dönüşümüne izin vermektedir.

2.5.6 STEP

STEP(Standards for the Exchange of Product model data – ürün modeli verilerinin değişimi için standart), sayısal ürün verisinin gösterimi ve değişiminin nasıl olması gerektiğini nesne tabanlı tanımlayan bir standarttır. STEP, bilgisayar destekli sistemlerin entegrasyonunun sağlanması için yapılan çalışmalar sonucunda ilk olarak 1984 senesinde önem kazanmış ve ilk standartlar 1994'de çıkarılmıştır [34].

Uluslararası katılımı ile gerçekleşen STEP standardı, endüstriyel otomasyon sistemleri ve ürün verilerinin sunuluşu ve değişimin bütünleştirilmesine yönelik standartlar serisi şeklinde halen geliştirilmektedir. Bu standart, daha önceden var olan bir çok farklı ulusal standardın (IGES, SET, VDA-FS, PDES, DXF, vb.) tanımlarını da içererek, bilgisayar destekli sistemler arasında veri paylaşımını uluslararası düzeyde tek bir standart çatı altında toplamayı hedeflemiştir [34]. Ancak bugün bilgisayar destekli sistemler DXF, DWG ve IGES başta olmak üzere diğer veri standartlarını da kullanmaktadır. Şekil 2.4'de STEP gelişimine etki eden standartlar gösterilmektedir.



Şekil 2.4 STEP gelişimi

2.5.7 IFC

IFC (Industry Foundation Classes), yapı sektöründeki farklı bilgisayar destekli sistemler arasında veri paylaşımı ve transferi için, uluslararası birlikte işlerlik kuruluşu IAI (International Alliance for Interoperability) tarafından geliştirilmiş uluslararası bir standarttır [47]. Bu standart, nesne modeli bilgisinin bilgisayar destekli sistemler arasında paylaşılabilir olması için içeriğinin ve yapısının nasıl olması gerektiğini tanımlamaktadır.

Yapı sektörüne yönelik nesne modeli verisinin temsil ve paylaşımının standardizasyonunu sağlamak amacıyla oluşturulan IFC, benzer paralellikte olan ve birçok sektörde başarı ile kullanılan STEP standardının mevcut alt yapısı üzerine geliştirilmiştir. IFC'nin asıl amacı yapı sektöründe ortak bir veri değişim "platformu" oluşturmaktır. Yapı projelerinin tüm profesyonel alanlarında yer alan süreçler arasında karşılıklı çalışabilirliği sağlaması düşüncesiyle oluşturulan bu platform, donanım ve yazılımdan bağımsızdır. Bu platformun bina modelini oluşturan nesne modellerini ve bu nesne modellerinin de bilgisayar destekli sistemlerde ki tüm ortak özelliklere ait bilgileri barındırması amaçlanmaktadır [48].

IFC'i bina modeline ait bütün bilgi sisteminin oluşturulması, depolanması ve aralarındaki iletişimin sağlanması amacıyla hazırlanan bir standart çalışması olarak tanımlamak mümkündür.

2.6 STEP (ISO 10303) Metodolojisi ve Amacı

STEP(Standards for the Exchange of Product model data – ürün modeli verilerinin değişimi için standart), bilgisayar destekli sistemler arasındaki veri değişimi için geliştirilmiş uluslararası bir standarttır. Bu standart, sayısal ürün verisinin gösterimi ve değişiminin nasıl olması gerektiğini tanımlayan kapsamlı bir ISO (International Organization for Standardization - Uluslararası Standardizasyon Organizasyonu) standardıdır (ISO 10303) [35].

STEP, ürünü tasarlandığı aşamadan yapımının tamamlandığı aşamaya kadar tanımlamak için gerekli olan bilgileri temsil etmenin standart bir yöntemi ve bu bilgileri ifade eden verilerin elektronik yolla değişimi için standart metotlar bütünüdür. Bu standardın amacı, çok sayıda farklı ulusal standartlara sahip olmak yerine, tüm endüstrilerde ürün oluşum döngüsünün tüm yönlerini kapsayacak tek bir standart sağlamaktır [34].

STEP standardının hedefi, farklı bilgisayar programları arasındaki iletişimi desteklemek ve tasarım ve analiz gibi çeşitli ürün oluşum döngüsü fonksiyonlarını yerine getiren sistemler arasında bütünleşmeyi sağlamaktır [36].

STEP, amaca göre sınıflara ayrılan ve bölüm diye adlandırılan bir dizi dokümandan oluşturulmuştur. Esas olarak, STEP dokümantasyonu beş ana kategoriye ayrılır: (1) Tanımlama Metotları, (2) Erişim Metotları, (3) Uyumluluk testi metodolojisi, (4) Bütünleşik Çekirdek Kaynaklar, (5) Uygulama Protokolleri [37].

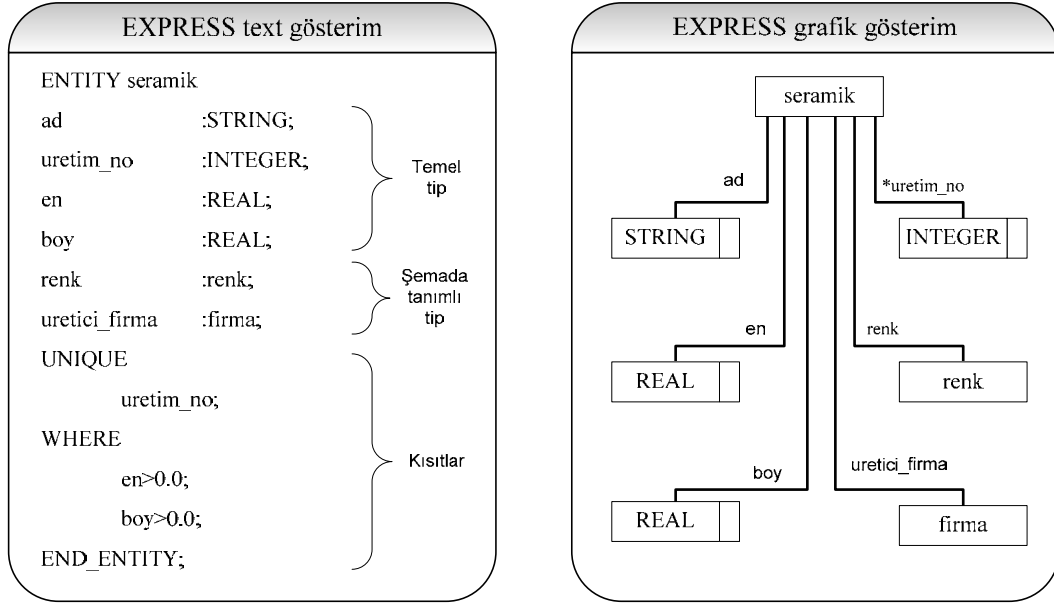
2.6.1 Tanımlama Metotları (Description Methods)

Bu grupta STEP alt yapısını oluşturan açıklama metotları yer almaktadır. Grup STEP evrensel tanımlarının yer aldığı genel giriş bölümü ve ürün tanımlaması için gerekli olan bilgileri belirtmek için kullanılan EXPRESS bilgi modelleme dilinin belirteçlerini içeren bölümlerden oluşmaktadır. Tanımlama metotları grubu bölümleri 1'den 19'a kadar olan sayılar ile numaralandırılmıştır.

EXPRESS, ürün verisinin temsil edilmesinin ve dönüşümün resmi tanımlarını sağlamak amacıyla bir ISO standardı olarak STEP için geliştirilmiştir (ISO 10303-11) [38]. EXPRESS, bilgi semantiğinin ifade edilmesini sağlayan bilgi tanımlama ve modelleme dilidir. Express dili ile oluşturulmuş şema tanımları, farklı programlama ve veri tabanı dilleri tarafından kullanılabilir şekilde dönüştürülebilmektedir.

EXPRESS grafik ve metin tabanlı olmak üzere iki farklı gösterime sahiptir (Şekil 2.5). Metin tabanlı gösterimler (EXPRESS-I, EXPRESS-M ve EXPRESS-c)

bir gramer tarafından tanımlı sözcüksel notasyon ve söz dizimi kullanan biçimsel dillerdir. EXPRESS dilinin grafik gösterimi ise EXPRESS-G olarak adlandırılmıştır. EXPRESS-G’de elemanlar ve öznitelikler kutularda gösterilirler. Elemanlar ve elemanlar ile öznitelikler arasındaki ilişkiler ise çizgiler olarak gösterilirler. Kısıtlamalar bu gösterimde yer almaz [39].



Şekil 2.5 EXPRESS Gösterimleri

2.6.2 Erişim Metotları (Implementation Methods)

STEP ürün verilerinin, veri değişim ortamında fiziksel olarak nasıl ifade edileceğine dair belirtileri sağlar. Bu bölüm için verilen bölüm numaraları ise 21’den 29’a kadardır.

- Bölüm 21 (ISO 10303-21): Bilgi modeli için oluşturulmuş herhangi bir EXPRESS şemasına uyan veri, bölüm numarası ile aynı adı taşıyan “part21” veya “SPF” olarak da bilinen dosya biçimi ile ifade edilir. Part21 dosyası ASCII şifreleme formundadır [40].
- Bölüm 22 (ISO 10303-22): Veritabanı yada bilgisayar programının içinde bulunan veri modellerine erişmek için SDAI(Standart Data

Access Interface) kullanılmaktadır. Bu arayüz ile ilgili tanımlar ISO 10303 standartlar serisinin 22. bölümünde belirtilmiştir [41].

- Bölüm23 (ISO 10303-23): C++ programlama dili için standart veri erişim arayüzüne bağlayıcı tanımlamaktadır [42].
- Bölüm24 (ISO 10303-24): C programlama dili için standart veri erişim arayüzüne bağlayıcı tanımlamaktadır [43].
- Bölüm25 (ISO 10303-25): EXPRESS'den XMI'a (XML Metadata Interchange) dönüştürücü tanımlamaktadır [44].
- Bölüm26 (ISO 10303-26): IDL programlama dili için standart veri erişim arayüzüne bağlayıcı tanımlamaktadır. Bu bölüm yürürlükten kaldırılmıştır.
- Bölüm27 (ISO 10303-27): JAVA programlama dili için standart veri erişim arayüzüne bağlayıcı tanımlamaktadır [45].
- Bölüm28 (ISO 10303-28): EXPRESS - XML dil bağlaması tanımlanmaktadır. EXPRESS modelleme dili ile tanımlanmış şema ve bu şemalara uyan veri modelleri, bu bölümde belirtilen spesifikasyonlar ile eşdeğer bir XML şema ve veri dosyalarına dönüştürülürler [46].
- Bölüm29 (ISO 10303-29): JAVA programlama dilinin sadeleştirilmiş şekli için standart veri erişim arayüzüne bağlayıcı tanımlamaktadır. Bu bölüm yürürlükten kaldırılmıştır.

2.6.3 Uyumluluk Testi Metodolojisi (Conformance Testing Methodology)

Bu bölüm STEP standardının uygunluk testi için belirtileri içermektedir. Uyumluluk testi, belirli bir test durumu için standardın farklı biçimde uygulanmasını önlemek amacıyla düşünülmüştür. Bölüm numarası olarak ise 31, 32 diye devam eden seri verilmektedir.

2.6.4 Bütünleşik Çekirdek Kaynaklar (Integrated Information Resource)

Bu grup kendi içinde genel kaynaklar ve uygulama kaynakları olarak iki alt gruba ayrılır. Genel Kaynaklar (kaynak bilgi modeli) şekil betimlemeleri, malzemeler, form özellikleri, toleranslar vb. gibi farklı türdeki uygulamaları destekleyen bilgi modellerinin spesifikasyonlarını içermektedir. Bölüm numarası olarak ise 41 ve daha yukarıları verilmektedir. Uygulama Kaynakları ise belli bir uygulama alanı(çizim, analiz vb.) için yapılmış olan bilgi modelleri ile ilgilidir. Bu gruptaki bölümler 101 – 110 arası numaralandırılır.

2.6.5 Uygulama Protokolleri (Application Protocols - AP)

Uygulama Protokolü spesifik bir uygulama için bilgi gereksinimleri ve konu olan alana uygun entegre edilmiş kaynakların kullanımını tarif eder. Uygulama protokolleri, içeriği, kullanımı ve bir ürünün yaşam sürecinde yer alan analiz gibi özel bir amaca yönelik bir çeşit ürün verisini tanımlayan standartlardır. Uygulama protokolleri STEP standardının ana bölümlerini teşkil etmektedir. Uygulama protokolleri özellikle spesifik uygulama sistemleri arasında veri değişimi için gerekli olan STEP bilgi modelinin alt kümelerini belirtmektedir.

Sonuç olarak, STEP standardı , bir tasarımı ürün haline getirmek için gerekli bütün işlemlerin ve bağlı parametrelerin standardize edilmesini ve tanımlanmasını; hiçbir yazılıma bağlı kalmadan gerçekleştirebilmeyi sağlamak için tasarlanmıştır. Fakat STEP birçok sektöre hitap eden genel bir standart olduğu için yapı sektörüne

ait ürün modeli tanımları sınırlıdır. Bu sebeple STEP, yapı sektörünün birlikte işlerlik problemlerinin çözümünde yetersiz kalmaktadır.

2.7 IAI - IFC

IFC (Industry Foundation Classes), yapı sektörüne yönelik bilgisayar destekli sistemler arasında birlikte işlerliği sağlamak amacıyla, uluslararası birlikte işlerlik kuruluşu IAI (International Alliance for Interoperability) tarafından geliştirilmiş uluslararası bir standarttır. IAI, yapım sektöründe birlikte işlerliği sağlamak amacıyla Ağustos 1994'de Amerika'da 12 şirketin bir araya gelmesi ile oluşturulmuştur. Şu an IAI, 24 ülkeden 400'ün üzerinde BDT şirketi ve diğer yapım sektörü ortaklarının (mimarlık ve mühendislik firmaları, malzeme üreticileri, meslek birlikleri ve kamu kurumları) katılımının yer aldığı geniş bir kuruluştur. IAI'ın, dünyanın çeşitli yerlerine yayılmış uluslararası konsey düzeyinde 11 şubesi bulunmaktadır [47].

IAI, yapım sektöründeki en geniş standartlaşma çabasıdır. Bu çabanın amacı mimarlık, mühendislik, inşaat, tesis yönetimi ve benzeri diğer faaliyetler ile ilgili olarak bilgisayar destekli sistemler (BDT, maliyet analizi, yapı ruhsatları, inşaat programları, işletme - bakım, vb.) arasında dinamik veri transferi ve paylaşımı için uluslararası standartları oluşturmaktır. IAI bu amaç doğrultusunda, veri paylaşımı için uluslararası bir standart olmasını hedefledikleri IFC standartlarının ilk sürümünü 1996 yılında ortaya çıkartmıştır.

2.7.1 IFC Sürümleri

IFC standartlarının ilk sürümü IFC 1.0 adı altında yayınlanmış ve 1996 yılının Aralık ayında IFC test dosyalarının ilk değişimi prototip uygulamalar arasında gerçekleştirilmiştir. IFC 1.0 Mimari tasarım, HVAC, bina yönetimi ve maliyet hesaplama disiplinlerinin birkaç sürecine ait tanımları içermekteydi [47].

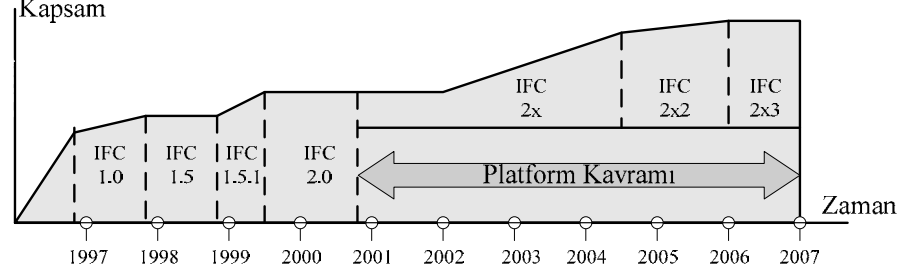
IFC 1.5.1, Ekim 1998'de yayınlanmıştır. Bu ikinci sürümde, yer alan disiplinlerde herhangi bir değişiklik olmamış ancak IFC 1.0 sürümünün deneyimleri doğrultusunda standardın yapısında değişiklik yapılmıştır. IFC nesne modeli çekirdeği genişletilmiş ve ticari yazılım geliştiricileri için güvenilir bir platform sağlamak amacı ile oturtulmuştur. IFC ticari olarak kullanılmaya ilk bu sürümle başlamıştır ve 2000-2001 yılları arasında IAI tarafından 6 yazılıma IFC 1.5.1 uyumluluk sertifikası verilmiştir (Autodesk - Architectural Desktop, Graphisoft – ArchiCAD, Nemetschek – ALLPLAN, Olof Granlund – Bsp, Data Design Systems – HousePartner, HAN Dataport – EliteNT) [47].

IFC 2.0 Nisan 1999'da yayınlanmıştır. Bu üçüncü sürümde standardın kapsadığı disiplinler genişletilmiştir. Standarda yeni eklenen disiplinlerdeki süreçleri destekleyen özellikler ve modeller eklenmiştir. 2001 yılına kadar 13 yazılım IFC 2.0 ile uyumlu sertifikası almıştır.

Ekim 2000'de yayınlanan IFC 2x; IFC 1.5.1 ve IFC 2.0 versiyonlarından gelen geribildirimler doğrultusunda geliştirilmiştir. Bu dördüncü sürümde IFC modellerin geliştirilmesine yönelik iki önemli değişim olmuştur. 1) Modellerin geliştirilmesi için iskelet yaratılmıştır. Bu iskelet sayesinde yeni model oluşturulması ve mevcut modellerin modüler bir şekilde genişletilebilmesi sağlanmıştır. 2) Önceki IFC sürümlerinin aksine IFC 2x kararlılık(stabilite) garantisi ile gelmiştir. 4 yıl boyunca değişmeyecek sabit çekirdek garantisi vermektedir. Bu versiyon aynı zamanda disiplinlerin uzantı modelleri için platform sağlamaktadır. Platform kullanımı ile projeler model geliştirebilmekte ve bu modeller tamamlanmış mevcut modellerden bağımsız olarak platforma eklenebilmektedir. Şu ana kadar IFC 2x ile uyumluluk sertifikası almış yazılımlar: Autodesk - Architectural Desktop, Bentley Systems – Microstation Triforma, Graphisoft – ArchiCAD, Nemetschek – ALLPLAN, Olof Granlund – BSpro, Vizelia – Vizelia FM, Data Design Systems – HousePartner [49].

IFC2x'den gelen geri bildirimler doğrultusunda Temmuz 2004'de IFC2x2 geliştirilmiştir. Şubat 2006'da ise günümüzdeki son IFC versiyonu olan IFC2x3

çıkartılmıştır[6]. Şekil 2.6’da IFC versiyonlarının tarihsel gelişimi grafik olarak gösterilmektedir.

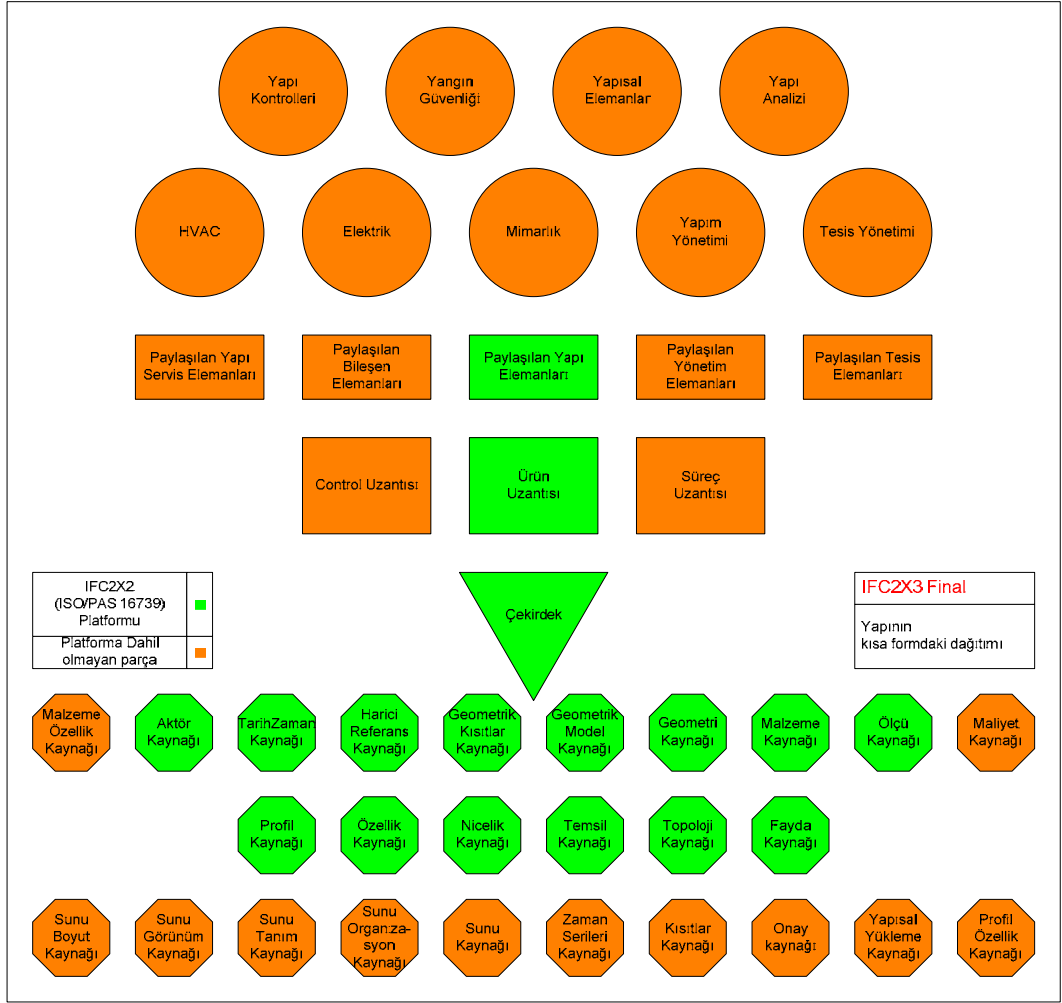


Şekil 2.6 IFC Sürümleri

IFC modelinin geliştirilmesi Thomas Liebich başkanlığındaki model destek grubu (MSG-Model Support Group) tarafından sürdürülmektedir. Şekil 2.7’de son sürüm olan IFC2x3 platformunda yer alan sınıflar gösterilmektedir. Platformun yeşil renkte gösterilmiş olan parçaları ISO onaylı sınıfları içermektedir ve bu sınıfların gelişimi değiştirilmemek üzere tamamlanmıştır. Şuan ki çalışmalar, tasarım, yapım ve yönetim alanlarındaki disiplinlerin daha çok kavramlarını temsil etmek için modeli genişletmek ve mevcut sınıflardaki diğer eleman tanımlarını sabit tutmak üzerinedir [6].

IFC ilk günden beri yapı oluşum döngüsü boyunca bütün bina topluluğuna hizmet etmeyi hedeflemektedir. Bu hedef doğrultusunda IFC geliştirme çalışmaları devam etmektedir. Aynı zamanda bu çabalar ticari geliştiricilerin de çalışabildiği sabit standartlarının daha fazlasını oluşturmak amacıyla modelin bir çok parçası için ISO belgelendirmesi alma yolundadır.

10 yılı aşkın bir süredir geliştirilmekte ve yeni versiyonları çıkmakta olan IFC, yapım sektöründeki en geniş çaplı standartlaşma çalışmasıdır. Bu standartta şuan 650’nin üzerinde model mevcuttur [6]. Yapım sektörünün tüm ihtiyacını karşılamak üzere model sayısının artması ve model tanımlarının genişlemesi beklenmektedir. IFC model mimarisi gerekli görülen uzantıların ve eklemelerin yapılmasına olanak sağlayacak şekilde katmanlı bir yapıya sahiptir.



Şekil 2.7 IFC 2x3 platformu

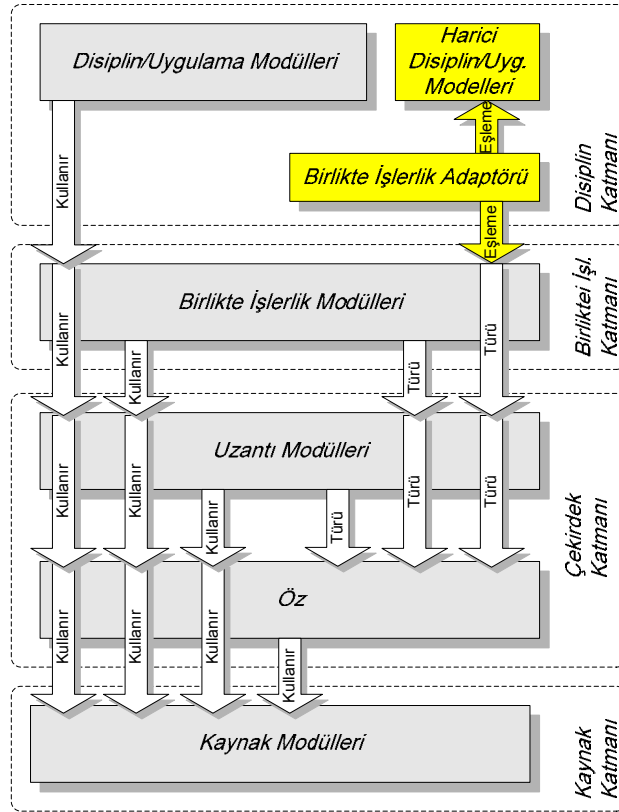
2.7.2 IFC Model Mimarisi

IFC model mimarisi bir takım ilkeler doğrultusunda geliştirilmiştir. Bu ilkeler:

- Model için modüler bir yapı sağlamak,
- Yapım sektöründe çeşitli disiplinler arasında bilgi paylaşımı için iskelet(frame-work) sağlamak,
- Model gelişiminin ve bakımının devamlılığını kolaylaştırmak,

- Model geliştiricileri için model bileşenlerinin yeniden kullanımını sağlamak,
- Model sürümleri arasındaki uyumluluk düzeyini artırmayı sağlamak.

IFC model, modüler olarak düzenlenmiş 4 katmanlı bir yapıya sahiptir (Şekil 2.8). Bu katmanlar; (1) Kaynak Katmanı (Resource Layer), (2) Çekirdek Katmanı (Core Layer), (3) Birlikte işlerlik Katmanı (Interoperability Layer) ve (4) Disiplin Katmanı (Domain Layer) [48].



Şekil 2.8 IFC model mimarisi

2.7.2.1 Kaynak Katmanı (Resource Layer)

IFC model yapısında temeli oluşturan en alt katmandır. Bu katman üst katmanlarda ki sınıflar tarafından kullanılan kaynak sınıflarını (ölçü, geometrik temsil, vb.) sağlar. Bu sınıflar uygulama ve disiplinlerden bağımsız en temel nesne

veya kavramları temsil etmek için temel veri yapılarını sağlamaktadır. Örneğin temel geometrik eleman tanımları IfcGeometryResource sınıfında yapılmaktadır. Bu sınıftaki geometri tanımlarını kullanacak olan nesne modeli üst katmanlardaki sınıflarda tanımlanmış olmalıdır. Benzer biçimde maliyet kavramı ile ilgili tüm tanımlar IfcCost entity de yer almaktadır. IfcCost, maliyet özelliğini gerektiren anlamsal modellerin herhangi biri yoluyla kullanılabilen veri yapısını sağlar. Bu katmandaki sınıflar tek başlarına bir anlam ifade etmezler. Üst katmandaki sınıflar ile birlikte kullanılırlar.

2.7.2.2 Çekirdek Katmanı (Core Layer)

Çekirdek proje modelini sağlayan bu katmanda öz ve çekirdek uzantılarının tanımları yer almaktadır. Çekirdek katmanı , IFC nesne modelinin temel yapısını sağlar. Üst katmanlardaki sınıflarda detaylandırılan kavram tanımlarının genel tanımları bu katmanda yapılmaktadır. Çekirdek katmanı kendi içinde (a) Öz (Kernel) ve (b) Çekirdek Uzantıları (Core Extensions) olmak üzere iki ayrı katmana bölünmüştür.

a) Öz (Kernel) katmanı , mevcut IFC sürümü kapsamında yer alan IFC modelleri için gerekli olan bütün temel kavramların yer aldığı sınıfları (IfcProduct, IfcProcess, IfcModelingAid, IfcDocument, vb.) sağlar. Öz katmanında kavramlar(modeller) en genel halleri ile tanımlanmaktadır. Bu katman, modellerin temel veri tiplerini veya ilişkilerini temsil etmek için genel yapıları(Nesne ilişkileri, özellikler, gruplar, vb.) sağlamaktadır. Öz, geliştirilen model içindeki bütün diğer sınıfların formunu tanımlayan bir şablon modeli gibi görülebilir. Bu katmandaki sınıfların yapıları çok geneldir, disipline veya uygulamaya özel değildir. Öz, Çekirdek Modelin altyapısıdır.

b) Çekirdek Uzantıları (Core Extensions) katmanı, Öz katmanında tanımlanan kavramların yapı sektörü spesifik tanımlarını ve uzantılarını içermektedir. Bunun anlamı Öz katmanındaki kavramlar, bu katmanda yapı sektöründe kullanımlarına göre genişletilir ve özelleştirilir. Her bir Çekirdek Uzantısı, Öz de yer alan sınıfların ihtisasıdır. Ayrıca başlıca ilişkiler ve roller de, Çekirdek Uzantıları içinde

tanımlıdır. Çekirdek Uzantısı katmanı , IFC model yapısında yapı sektörü projeleri için ilk anlamsal katmandır. Ürünler, işlemler, yapı kaynakları gibi temel kavramları tanımlar.

2.7.2.3 Birlikte İşlerlik Katmanı (Interoperability Layer)

Bu katmanda yapı sektöründeki disiplinler arası paylaşılan yapı elemanlarının (Shared Building Elements) ve yapı servis elemanlarının (Shared Building Service Elements) tanımları bulunmaktadır. Birlikte işlerlik katmanının tasarımındaki esas hedef, iki veya daha çok disipline ait modelleri tanımlayan şemaları oluşturmaktır. Bu şema, farklı disiplin modelleri arasında birlikte işlerliği etkinleştirir. Plug-in model yaklaşımı bu katmanda ortaya çıkmaktadır. Plug-in yaklaşımı , disiplin modellerinin dış kaynaklar tarafından da geliştirilmesine imkan tanımaktadır. IFC uyumlu yazılımlar, veri paylaşımı ara birimlerini(interface) oluşturmak için bu katmandaki modelleri kullanacaklardır.

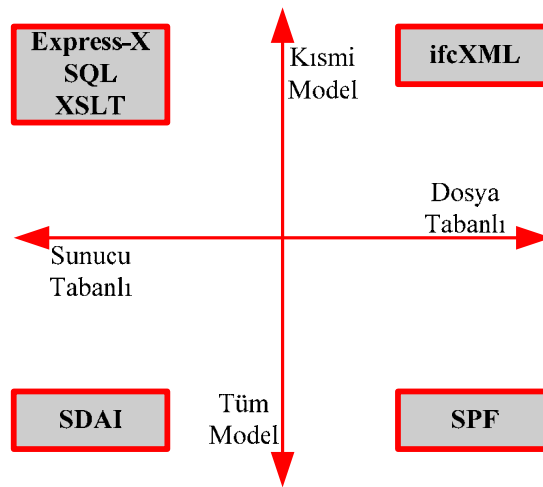
2.7.2.4 Disiplinler Katmanı (Domain Layer)

Bu katmanda mimarlık, tesis yönetimi, yapı yönetimi gibi yapı sektöründeki disiplinlere özel detaylandırılmış tanımlar yer almaktadır. Bu katmandaki modellerin her biri Çekirdek katmanın da veya Kaynak katmanın da tanımlanan herhangi bir sınıfı kullanabilen veya bu sınıflara başvurabilen modeldir. Ayrıca, disiplinlere özel diğer endüstri çabaları tarafından geliştirilmiş dış modellerin IFC ile uyumunu sağlamak için geliştirilen bağdaştırıcılar(adaptör) bu katmanda yer alır.

Modüler yapıdaki IFC katmanları arasında birtakım referans ilkeleri kullanılmaktadır. Herhangi bir katmandaki bir sınıf, aynı katmandaki veya daha düşük katmandaki başka bir sınıfa başvurabilir ve bu sınıf tanımlarını kullanabilir. Ancak daha üstteki katmandaki bir sınıftan yararlanamaz. Aynı katman içinde başvurular, model tasarımında ki modülerliği bozmamak için çok dikkatli

tasarlanmalıdır. Kaynak katmanındaki sınıflar sadece bu katmandaki başka bir sınıftan referans alabilirler.

IFC sınıf hiyerarşisi , yapıyı oluşturan elemanları, elemanların geometri ve malzeme özellikleri, maliyeti, iş planı, organizasyonu v.b. çekirdek proje bilgisini kapsar. IFC, bu proje bilgisinin bilgisayar destekli sistemler arasında paylaşılabilmesi ve transferi için standart veri modeli ve tarafsız dosya biçimi sağlar. IFC’de kullanılan dört farklı veri transferi mekanizması Şekil 2.9’da gösterilmiştir. Veri tabanına kaydedilen veri modeline ulaşmak için “STEP SDAI” arabirimi yada “Express-X” gibi veritabanı sorgulama dilleri kullanılır. IFC, verinin dosya yoluyla transferi için “SPF” ve “ifcXML” olmak üzere iki çeşit dosya biçimi sağlar. EXPRESS dili ile oluşturulmuş IFC şemasına uyan proje bilgisi, part21 olarak da bilinen SPF dosya biçimi ile ifade edilir. SPF dosya biçimi STEP’in veri transferi için kullanılan en yaygın mekanizmasıdır(ISO 10303-21). Bu dosyalar , Web üzerinden taşınabilmektedir. Fakat STEP/EXPRESS anlayışındaki, dosyaların yalnızca yazılımlar tarafından yorumlanması mantığı dosyaların anlaşılabilirliği konusunda problem oluşturmaktadır. SPF dosya biçiminde tutulan proje bilgisini, STEP yardımcı araçları olmadan işlemek ve analiz etmek karmaşık ve zordur. SPF dosya biçimine bir alternatif, XML dili ile oluşturulmuş ifcXML şemasına göre biçimlendirilmiş ifcXML dosya biçimidir [50].



Şekil 2.9 IFC veri transfer mekanizmaları

2.7.3 ifcXML

ifcXML, EXPRESS modelleme dili ile tanımlanan IFC modelinin XML dili ile temsilidir. IAI, yapı sektörüne yönelik nesne modeli verisinin temsil ve paylaşımının standardizasyonunu sağlamak için oluşturduğu IFC standardını, benzer paralellikte olan ISO/STEP standartlar serisinin mevcut altyapısından yararlanmak ve genel nesne modellemeye ilişkin standart tanımlarını kullanabilmek için EXPRESS modelleme dili ile tanımlamıştır. Ancak EXPRESS modelleme dilinin yardımcı araçlar olmadan anlaşılması ve kullanılması oldukça güçtür.

XML dilinin kolay okunabilir, anlaşılabilir ve işlenebilir olmak gibi sahip olduğu bir takım avantajlar, bir çok yazılım ve internet şirketinin bu teknolojiyi kullanmasına ve geliştirmesine etken olmuştur. XML topluluğunun esas odak noktası web tabanlı servisler olmasına rağmen, XML'in birlikte işlerlik amacına yönelik kullanımı giderek artmaktadır ve birçok sektörde olduğu gibi yapım sektöründe de birçok çalışma ontoloji standardı olarak XML şeması önerileri için yatırım yapmaktadır.

XML dilinin bu yönde kullanımındaki ve bu dile yönelik geliştirilen teknolojilerdeki hızlı artış sonucunda IAI, EXPRESS dili ile tanımlı IFC tanımlamasını, aynı zamanda XML dili ile de tanımlama girişimlerini başlatmıştır. ifcXML olarak adlandırılan ve 2001 yılında başlayan bu girişim, Thomas Liebich başkanlığındaki IAI model destek grubu (Model Support Group) tarafından geliştirilmiştir [51]. ifcXML'in hedefi, XML topluluğunca IFC tanımlamasının içerik ve yapısının uluslararası düzeyde anlaşılabilir olmasını sağlamaktır. Bu hedef doğrultusunda EXPRESS dili ile oluşturulan IFC model verisinin ve bu verinin yapısını tarif etmek için tanımlanan mevcut IFC şemasının, XML dili ile oluşturulması çalışmalarına başlanmıştır.

ifcXML, EXPRESS dilindeki IFC model tanımlarının otomatik dönüşümü sağlanarak oluşturulmaktadır. IFC EXPRESS'ten ifcXML'in otomatik üretimi, EXPRESS şema ile XML şema(XSD) arasında, EXPRESS şema semantiğine uyan

bir dönüşümü sağlayan EXPRESS-XML dil bağlaması (ISO10303-28) kullanılarak yapılmaktadır [46].

ifcXML'in kullanılmasının hedeflendiği iki alan vardır. Bunlar:

- IFC Model verisi transferinde, alternatif olarak XML durum belgeleri şeklindeki IFC Model verisi dosyalarının kullanılması.
- Veri paylaşımı ve transferi konularında başlatılan XML tabanlı girişimlerde, IFC içerik ve yapısının kullanılması.

Bu iki alanda da ifcXML kullanımını sağlayabilmek için İki aşamalı dönüşüm yöntemi uygulanmaktadır.

- 1. Aşama: IFC modelin transferinde kullanılan yöntem olan "EXPRESS tabanlı STEP Fiziksel Dosya (SPF) biçimindeki IFC Model verisi"ne alternatif olarak "XML dosyası biçimindeki IFC Model verisi"nin kullanılmasını sağlamak için EXPRESS dilindeki IFC modelinin otomatik olarak XML şema notasyonuna dönüştürülmesi aşamasıdır. Bu otomatik çeviri yoluyla elde edilen XML şema dosyası, tipik XML yapısından çok farklı olup, tamamen IFC EXPRESS modelinin sözdizimi kurallarını korumaktadır. Bu sebeple, 1. aşama sonrası oluşan XML şema dosyası tipik XML kullanımından çok daha büyük ve karmaşıktır.
- 2. Aşama: XML tabanlı standart girişimlerinin kullandığı XML veri havuzu olarak hizmet etmek için IFC EXPRESS modelinin optimize edilen çevirisidir. Bu aşamada en uygun XML veri tanımları sağlamak için, EXPRESS dilindeki IFC şema dosyasından otomatik yolla üretilmiş XML dilindeki IFC şema dosyası (1. aşama sonucu), XML dilinin sözdizimi kurallarına uygun olarak dönüştürülür. Değişiklik sadece sözdizimi kurallarında olmaktadır. Optimizasyon işlemi sonucu oluşacak dosya, IFC EXPRESS modelinin semantiğini sürdürür ama sözdizimi kuralları XML yapısına çok daha uygundur.

EXPRESS tabanlı dosyadan otomatik çeviri ile elde edilen XML tabanlı dosyanın, XML söz dizimi kurallarına uygun hale dönüştürülmesi(ifcXML) işlemi ile XML diline en uygun sonuçları elde etmek için birtakım ayarların yapılması gerekmektedir. Bu ayarlar konfigürasyon dosyasında tanımlanır. Bütün IFC versiyonlarının şema dosyalarının XML diline otomatik dönüştürülerek ifcXML şema dosyalarının üretilmesi için IAI Model Support Group bu konfigürasyon dosyasını standart hale getirmiş ve IAI resmi web sitesinde yayınlamıştır [52].

XML işaretleme dili ile EXPRESS modelleme dili yapıları ve içerikleri birebir örtüşmez. EXPRESS dilinde var olan bazı yapılar XML dilinde yoktur. Bu yüzden EXPRESS dili ile oluşturulmuş IFC Model tanımlarından otomatik türetilen ifcXML şema dosyasında bazı bilgi kayıpları yaşanmaktadır. Şuan hiçbir ifcXML temsili orijinal modelin tüm bilgisini koruyamamaktadır.

2.8 XML

XML (Extensible Markup Language – Genişletilebilir İşaretleme Dili), W3C (World Wide Web Consortium – WWW Birliği) tarafından, herhangi bir belgenin yapısını ve görünümünü tanımlamak için geliştirilmiş uluslararası bir standarttır.

İlk olarak 1996 yılının Ağustos ayında XML dilinin basit dizayn yapısının çalışmalarına başlanmıştır. SGML’96 konferansında, XML’ in ilk taslak versiyonu (draft) yayınlanmıştır. Diğer ayrıntıların tamamlanması bir yıl daha sürmüştü ve Şubat 1998’de XML 1.0 bir standart olarak W3C tarafından yayınlanmıştır [53].

XML verinin tanımlanması ve tarif edilmesi için kullanılan bir işaretleme dilidir. XML işaretleme etiketlerini kullanır ve kullanılan bu etiketler önceden tanımlı değildir. Yani bir XML dokümanının yapısı tamamıyla kullanıcı tarafından oluşturulur. XML yapıda bilgi içeren bir dosya aslında, verilerin okunurluğunu ve kontrolünü kolaylaştırmak için belirli etiketler (tag) kullanarak formatlanmış metin (text) dosyasından başka bir şey değildir. XML özellikle verilerin tanımlanması ve farklı ortamlar arasında transferini sağlamak için geliştirilmiştir.

Önceden tanımlanmış bir forma sahip olmayan XML , diğer dillerin tanımlanabildiği bir meta-dildir. Meta-dil, dil yaratmaya yarayan dil anlamına gelmektedir. Diğer bir deyişle XML başka dillerin yapısını tanımlamakta kullanılan bir dildir. XML ile herhangi bir uygulama için bir XML belgesinin içinde bulunacak verinin içeriği ve içerdiği veri tiplerini tanımlayacak uygulamaya özel bir işaretleme dili tanımlanabilmektedir [54].

XML' in diğer işaretleme dilleri ile arasındaki en belirgin fark XML' in verinin kendisiyle ilgilenmesidir. XML diğer işaretleme dilleri gibi veriyi görüntülemek için değil de ana amaç olarak veriyi tanımlamak için geliştirilmiştir. Diğer işaretleme dilleri sadece veriye ilişkin şekillendirme bilgilerini içerirken, XML dokümanları verilerin tanım bilgilerini içermektedir. XML' de veri kendi kendini tanımlayabilir bir yapıdadır. XML verisi belirli tanımlama tabloları, farklı veri tanımlama araçları gibi unsurlara gerek duymaz çünkü verinin tanımı kendi içerisinde bulunmaktadır. XML, verinin sadece görüntülenmesinin yanında tamamı ile kullanılabilir kılınmasını sağlamaktadır. Bu özelliği ile XML, özellikle verinin işlenmesine yönelik uygulamalar için önemli bir avantaj sunmaktadır [54].

Veri transferinin kolaylaşmasını hedefleyen XML de verilerin içerik, yapı ve sunum kısımları ayrı modüller halinde farklı XML dokümanlarında tutulmaktadır [55]. XML ile verilerin yapı bakımından modülerlik kazanması, veri transferini kolaylaştırmaktadır. Ayrıca XML, veriyi düz metin (plain text) formatında saklamasından dolayı veriyi paylaşırma konusunda hem yazılım hem de donanımdan bağımsız hareket edebilme imkanını sunmaktadır.

XML, belgenin yapısının içeriğin oluşturulmasından bağımsız olarak belirlenebilmesine; içeriğin biçimden tamamen ayrılabilmesine; belli bir yapıdaki verinin farklı uygulamalarda ve ortamlarda anlaşılabilmesine; aynı verinin birden çok biçimle sunulabilmesine ve aynı biçimin birden çok veri öbeğine uygulanabilmesine olanak tanımaktadır.

2.8.1 XML Belgelerinin İşlenişi ve XML Teknolojileri

Genel olarak bir XML dokümanını üç dosyayla tanımlayabiliriz. Dosyalardan biri XML formatlı veri içerik dosyası, biri şema dosyası(XSD veya DTD), biri de stylesheet (XSL) dosyasıdır. Bu üç dosya tipi de XML tabanlı olduğundan XML destekli tüm sunucu ve veri tabanlarında çalıştırılabilmektedir.

XML standardına göre hazırlanmış bir dokümandaki verilerin işlenmesinde ayrıştırıcı (DOM, SAX) kullanılır. Ayrıştırıcı, XML dokümanındaki veriyi şema dosyasındaki tanımlara göre XML'e özgü hiyerarşik yapıda işler, uygulama ve sorgulamalara hazır hale getirir. Dokümanın gösterimi için de stylesheet işlemcisi, doküman ve bu dokümana ait stylesheet dosyasını birleştirerek dokümanın görüntülenmesini sağlar.

2.8.1.1 DOM (Document Object Model) ve SAX (Simple API for XML)

Ayrıştırıcılar (Parser) XML veriyi işlemek için kullanılan programlar veya arabirimlerdir. XML dokümanlarının işlenmesi ve analizinde günümüzde DOM(Belge Nesne Modeli) ve SAX (XML için Basit Programlama Arabirimi) ayrıştırıcıları kullanılmaktadır.

DOM, XML belgesindeki verileri, belgenin kendisindeki yapıyı taklit eden hiyerarşideki bir ağaç benzeri yapı ile bellekte depolar. DOM ile XML belgesinde sunulan belgenin kök elemandan başlayarak bir ağaç yapısına dönüştürülmesi ve XML ile temsil edilen verinin uygulama tarafından kullanılması esnasında , bu ağaç yapısını kullanılması sağlanır. Bu arabirim, bir XML belgesinin bileşenlerini temsil edebilen ve değiştirebilen nesnelere, özelliklere, yöntemlere ve olaylardan oluşan bir nesne modeli biçimindedir [56].

SAX, kullanıcının belge boyunca belirli yerlerde yerine getirilecek belirli davranışları tanımladığı olaya dayalı bir arabirimdir. Bu ayrıştırıcı XML dokümanının tüm yapısını inceleyip bellekte tutmaz. XML dokümanını ayrıştırırken

herhangi bir elemana rastladığında programcının belirleyeceği bir fonksiyonu çağırır ve okuduğu XML elemanı ile ilgili bilgileri bu fonksiyona aktarır. Bu arabirim, XML belgesinin bir örneğini bellekte yaratmadığı için ana belleğe bağlı bir sınırlamaya tabi olmadan performansını sürdürebilmektedir [56].

Bu iki ayrıştırıcının birbirlerine göre üstünlükleri bulunmaktadır.

- SAX hızlıdır, az bellek harcar, DOM yavaştır ve çok kaynak tüketir.
- DOM veri üzerinde değişikliğe izin verir, SAX bunu yapamaz sadece veriyi görüntüleme amaçlıdır.
- DOM bir kez yüklenir ve veriye istenildiği gibi erişme hakkı verir, SAX veriye erişim için kısıtlara sahiptir.

2.8.1.2 DTD (Document Type Definition - Belge Tipi Tanımı)

XML belgelerini oluşturan elemanları tanımlamak için kullanılan şema dosyalarıdır. XML veri dosyalarının yapısı hakkında bilgileri içermektedirler. DTD hangi isimli elemanın doküman içerisinde yer alabileceğini, hiyerarşik yapının ne olacağını, olması mecburi olan ve olmayan elemanların hangileri olduğunu gösterir. Bazı XML dosyalarının belirli bir standartta olması gerekir. DTD ile standardın tanımı ve XML dosyalarının bu standartlara uyup uymadığı kontrol edilebilir. DTD, XML dosyasının içerisinde ya da bulunduğu yerin adresinin belirtilmesi yolu ile harici olarak XML belgesi ile ilişkilendirilebilir [57].

2.8.1.3 XML Schema (XSD - XML Şema Tanımı)

XML Schema, DTD'nin alternatifi olan bir XML dokümanı tanımlama standardıdır. XML Schema tıpkı DTD lerde olduğu gibi doküman içindeki elementlerin ve içerik modellerinin tanımlanması işlemini gören şema dosyalarıdır. Ancak XML Schemalar elementlerin bu tanım işleminde DTD ye göre daha zengin

bir veri tipi kaynağı ve özellik(attribute) desteği sağlamaktadır. DTD içerisinde, element içeriği string ve diğer birkaç veri tipi ile sınırlıdır. XML Schema çok geniş bir veri tipi desteği sağlamaktadır (integer, floating, point sayılar, date, time, gibi). XML schema'nın kendisi de bir XML uygulamasıdır. XML schema kullanarak şema dosyalarının oluşturulması tıpkı diğer XML dokümanlarının oluşturulmasına benzer. Bu DTD lere göre bir üstünlüktür [57].

2.8.1.4 XSL(Genişletilebilir Stil Sayfa Dili)

XML ile belirtilen verinin nasıl sunulacağını belirten, kendisi de XML ile tanımlanan bir dildir. XSL, XML kodlarını değişik formatlara çevirmek için kurallar tanımlayan bir stil uygulamasıdır. Bir sayfada verinin nasıl konumlandırılacağını ve gösterileceğini belirten dil XSL-FO' dur. Bir XML' i başka bir XML' e veya diğer bir formata dönüştüren dil de XSLT' dir. XSL, çok miktarda verinin aynı görünümde sunulabilmesine olanak tanıdığı gibi, aynı verinin farklı biçimde sunulabilmesine de olanak tanır. Kullanıcılar, aslında aynı olan veriyi kendi tercihlerine veya ihtiyaçlarına uyacak biçimde görebilirler. Aynı verinin farklı çözünürlük veya renk özelliklerine sahip bilgisayar ekranlarında, hatta bilgisayar dışında cep telefonu, ATM cihazları veya çeşitli makinelerin görüntü cihazlarına uygun olarak gösterilmesi sağlanabilir [57].

XML' in en önemli özelliklerinden biri W3C tarafından bir standart olarak kabul edilmesidir. İş dünyasında, birbirleri ile işlem yapacak dolayısıyla veri değişimi gerçekleştirecek işletmelerin veya kurumların birbirleri ile aynı dili konuşmaları standart bir dilin oluşması ile mümkün olabilecektir. XML, genel yapısı ve dünyada çok yaygın kabul görmesinin yanı sıra standart olarak kabul edilmesi ile birlikte işletmelere verimli etkileşim imkanı sağlamaktadır. Microsoft, IBM, Sun ve Oracle gibi birçok teknoloji lideri firma XML' i elektronik veri değişiminin yeni standardı olarak kabul etmekte, XML ve XML uygulama standartlarını desteklemekte ve XML tabanlı yeni ürünlerini bilgi teknolojisi uygulamalarının kullanımına sunmaktadır [58]. XML' in popülerliğinden dolayı yapım sektöründe birçok çalışma ontoloji standardı olarak XML şeması önerileri için yatırım

yapmaktadır(yönetimsel uygulamalar için vb.). XML topluluğunun esas odak noktası web tabanlı servisler olmasına rağmen, XML birlikte işlerlik perspektifi açısından da oldukça önemli bir noktaya ulaşmıştır.

3. METODOLOJİ

Bu bölümde tez kapsamında geliştirilen ESTARA programı hakkında bilgi verilmektedir. ESTARA, mimarlık eğitiminde hacim akustiği ile ilgili temel konuların öğretilmesine yardımcı olmak amacıyla planlanan ve henüz gelişim aşamasında olan yeni bir bilgisayar programıdır. Bu program, hacim akustiği analizinde kullanılacak verilerin BDT ortamında oluşturulan proje bilgisinden otomatik olarak sağlanmasına olanak tanımaktadır. Bu tür bir yaklaşımla dijital ortamda bulunan mimari proje verilerinin akustik analiz programına tekrar el ile girilmesinin yaratacağı zaman kaybı ve hatalı veri girişi riski engellenmiş olacaktır. Otomatikleştirilen ve kolaylaşan veri girişi sayesinde yapının akustik performans analizi zaman kaybı olmadan yapı oluşum döngüsünün çeşitli aşamalarında projesindeki değişiklikler doğrultusunda tekrarlanabilecektir. Bu program analizcilere, mimarlık eğitimcilerine ve öğrencilerine akustik performans analizin yapılmasına yönelik veri girişine destek olmak amacıyla tasarlanmıştır.

Yapılan çalışma, kullanılan teknoloji ve sistemin yapısı olmak üzere iki bölümde detaylandırılacaktır.

3.1 Teknoloji

Bu bölümde, ESTARA'nın geliştirilmesinde kullanılan teknolojiler hakkında bilgi verilecektir. Mevcut teknolojiler arasından, veri temsil ve transferinde standart olarak IFC/ifcXML'in, BDT programı olarak ArchiCAD'in ve programlama ortamı olarak JAVA'nın tercih edilmesinin sebeplerinden bahsedilecektir.

3.1.1 Standart (IFC/ifcXML)

Yapım sektöründe kullanılan BDT sistemleri birbirleriyle veri alışverişinde bulunabilmek için IGES, DWG, DXF, STEP ve IFC gibi veri temsil ve transferine

yönelik geliştirilen standartları kullanmaktadırlar. Bölüm 2’de her biri hakkında bilgi verilen bu standartların belli sınırlamaları bulunmaktadır. DXF ve DWG üç boyutlu katı model bilgilerini, IGES ise geometrik veriler dışında ki bilgileri içermemektedirler. STEP ürünün üretim süreci bilgilerini de geometrik bilgileri gibi barındırmakta, fakat birçok sektöre hitap eden genel bir standart olduğu için yapı sektörüne ait nesne modeli tanımları sınırlıdır. IFC ise yapı sektörüne özel geliştirilen bir standarttır ve geometri ve ürün verisinin yanı sıra tasarım, analiz, uygulama ve bakım gibi yapı oluşum döngüsünü destekleyen proje bilgisini içermektedir.

Bu çalışmada; BDT programında oluşturulan proje bilgisinin, IFC standardının veri paylaşım mekanizmalarından biri olan ifcXML dosya çıktısı olarak alınması tercih edilmiştir. Bunun sebeplerinden biri ifcXML’in tasarım ve analiz programlarında kullanılan terim ve ilişkilerin bir çoğunun temsilini sağlamasıdır. Ayrıca JAVA ortamında geliştirilen ESTARA projesinde, Java’nın XML bağlayıcılarından (binding) yararlanan ifcXML ile çalışmak bu projede bize kolaylık sağlamıştır. Bir diğer sebep ise ifcXML dosyalarının XML tabanlı olmasıdır. ifcXML dosyalarının içerik ve yapısının tanımladığı XML şema dosyaları(configuration.xsd, ex.xsd ve IFC2X2_FINAL.xsd) IAI resmi internet sitesinde herkesin ulaşımına açıktır [52]. XML için geliştirilen ve herkes tarafından edinilmesi mümkün olan birçok araç yardımı ile ifcXML şema dosyaları programlama dillerine uygun olarak dönüştürülebilmektedir. Proje bilgisinin ifcXML formatında temsil edilmesinin bu projede sağladığı bir diğer avantaj, XML dokümanlarının sorgulama ve analizinin yapılabilmesi kolaylığıdır.

ifcXML şeması, EXPRESS dilindeki IFC model tanımlarının otomatik dönüşümü sağlanarak oluşturulmuştur. Bu dönüşüm sırasında XML işaretleme dili ile EXPRESS modelleme dilinin birebir örtüşmemesinden dolayı bazı bilgi kayıpları olmaktadır. Şuan hiçbir ifcXML temsili orijinal modelin tüm bilgisini koruyamamaktadır. Fakat hacim akustiği analizi programı, nesnelerin sadece geometrik ve malzeme tanımları ile ilgilendiği için bu bilgi kaybından etkilenmez.

Bu projede ifcXML dosyaları ile çalışmanın yarattığı bir diğer sorun ise; EXPRESS den otomatik olarak türetilen ifcXML şema dosyalarının karmaşık bir yapıda olması ve boyutunun çok büyük olmasıdır.

3.1.2 BDT Programı (ArchiCAD)

Analiz programlarında kullanıcıların en çok vakit kaybettikleri kısım, mekana ait 3 boyutlu geometrik ve geometrik olmayan bilgiyi barındıran başlangıç girdisini oluşturmaktır. Başlangıç girdisini BDT ortamındaki mevcut tasarım bilgisinden alınmasını otomatikleştirmek bu zaman kaybı ve hatalı veri girişi sorunlarına çözüm olacaktır.

Günümüzde kullanılan modern BDT programları ile hacim akustik analiz programları için gerekli verinin tümünü barındıran modelleri oluşturmak mümkün olmaktadır. Bu programlar aynı zamanda uzantı ve eklerin(add-ons) geliştirilmesi için kendi arayüzlerini sunmaktadırlar. Ancak eğitim amaçlı geliştirilen ESTARA'nın, ticari BDT programlarından bağımsız olarak tutulması özellikle tercih edilmektedir. Bu projede “BDT – Akustik Analiz” arasında veri transferi dosya tabanlı veri değişim mekanizması ile sağlanmıştır. Bu tür dosya tabanlı veri değişimi mekanizmasına olanak tanıyan IFC, önde gelen BDT firmaları tarafından desteklenmektedir.

IFC'nin XML dilinde temsil edilmiş şekli olan ifcXML dosyaları ile çalışan ESTARA, bu özelliği sayesinde IFC uyumlu herhangi bir BDT programı kullanılarak akustik analiz için model oluşturulmasına imkan tanımaktadır. Bu tez kapsamında BDT sistemi olarak Graphisoft firmasının ürettiği ArchiCAD programı kullanılmıştır. ArchiCAD, IAI kuruluşunun IFC uyumluluk sertifikasına sahip programlardan biridir. Bu programa Graphisoft firmasının resmi internet sitesinden ulaşılabilen IFC2x2 eklentisinin ilave edilmesiyle IFC dosyalarının (.ifc ve .ifcXML), okuyabilmesi ve yazabilmesi sağlanmaktadır [59].

3.1.3 Programlama Dili(Java)

Bütünleştirilmiş bir yapı tasarım ve üretim ortamının oluşturulmasında nesne yönelimli ürün modelleme yaklaşımının öneminden, birlikte işlerlik için geliştirilen standartların ve sektörde kullanılan yazılımların birçoğunun nesne tabanlı olduğundan bahsedilmiştir. Bu bağlamda ESTARA programı da nesne yönelimli programlama dillerinden biri olarak Sun Microsystems tarafından geliştirilen JAVA programlama dili ile yazılmıştır. JAVA dilinin nesne yönelimli olmasının yanı sıra yazılım ve donanımdan bağımsız, çeşitli platformlarda çalışabilen bir uygulama geliştirme ortamı sunması, ESTARA'nın bu programlama dili ile yazılmasının bir diğer sebebidir. Java'da yazılan bir program; Unix, Macintosh, Windows 95 veya Windows NT veya herhangi bir 32 bit makinede hiç değiştirilmeden kullanılabilir [60].

Sun Microsystems şirketinin XML standardını destekleyici ve tamamlayıcı nitelikte geliştirdiği ve hizmete sunduğu bir çok teknoloji bulunmaktadır [61]. Bu teknolojiler arasında XML belgeleri yaratmak ve işlemek için tasarlanmış olan ve yaygın olarak kullanılan iki arabirim bulunmaktadır:

- JAXP (XML İşlemleri için Java API'si - Java API for XML Processing),
- JAXB (XML İlişkilendirilmesi için Java Mimarisi - Java Architecture for XML Binding).

JAXB, XML belgeleri ile Java nesneleri arasında dönüşüm sağlamaya yarayan bir kütüphanedir. Bu kütüphane XML şemasındaki tanımlardan Java sınıflarının yaratılmasını sağlamaktadır [62].

JAXP, XML belgelerinin işlenmesi ve analizinde en genel kullanılan iki çeşit ayrıştırıcıyı kapsamaktadır: DOM (Document Object Model - Belge Nesne Modeli) ve SAX (simple API for XML - XML için Basit Programlama Arabirimi). DOM, XML belgesindeki verileri, belgenin kendisindeki yapıyı taklit eden hiyerarşideki bir ağaç benzeri yapı ile bellekte depolar. Bu arabirim, bir XML belgesinin bileşenlerini

temsil edebilen ve deęiřtiren nesnelere, özellikler, yöntemler ve olaylardan oluşan bir nesne modeli biçimindedir. SAX, kullanıcının belge boyunca belirli yerlerde yerine getirilecek belirli davranışları tanımladığı olaya dayalı bir arabirimdir. Bu arabirim, XML belgesinin bir örneğini bellekte yaratmadığı için ana belleğe baęlı bir sınırlamaya tabi olmadan performansını sürdürebilmektedir. JAXP, SAX ve DOM'u ortak kullanarak çalışmaktadır [63].

Java programlama dili ile yazılan ESTARA programında ifcXML dosyalarının okunabilmesi ve analiz edilebilmesi için ifcXML şemasının ayrıştırılıp derlenerek IFC-JAVA kütüphanesinin oluşturulması gerekmektedir. IfcXML şemasının ayrıştırılmasında, DOM arabirimini desteklemesinden dolayı Altova XMLSpy'ın JAXP paketi kullanılmıştır [64].

3.2 ESTARA

Hacim akustięi benzetim(simülasyon) programları, mekan bileşenlerinin biçim, boyut ve malzemelerinin mekan akustięini nasıl etkilediğini analiz edebilen, değerlendirme ve karar verme araçlarıdır. ESTARA(Educational Simulation Tool for Architectural Room Acoustics) mimarlık eğitiminde hacim akustięi ile ilgili temel konuların öğretilmesine yardımcı olmak amacıyla planlanan ve henüz gelişim aşamasında olan yeni bir programdır.

ESTARA, öğrencilerin çeşitli tasarım kararlarının hacim akustięi üzerindeki etkilerini benzetim ortamında değerlendirmelerine imkan veren bir bilgisayar programı olarak tasarlanmıştır. Bu program;

- Örnek kapalı mekanın BDT ortamında modellenerek, akustik performansının değerlendirilmesi ve analiz sonuçlarının karşılaştırmalı değerlendirme tabanı olarak kullanılmasına,
- Parametrelere dayalı alternatif modellerin oluşturulmasına, bu modellerin akustik açıdan kendi aralarında ve taban analiz değerleri ile karşılaştırılmasına,

- Pasif ve aktif parametreleri akustik uygunluğu sağlayacak şekilde entegre eden, en etkin modelin belirlenmesine yönelik bir hedef doğrultusunda geliştirilmiştir.

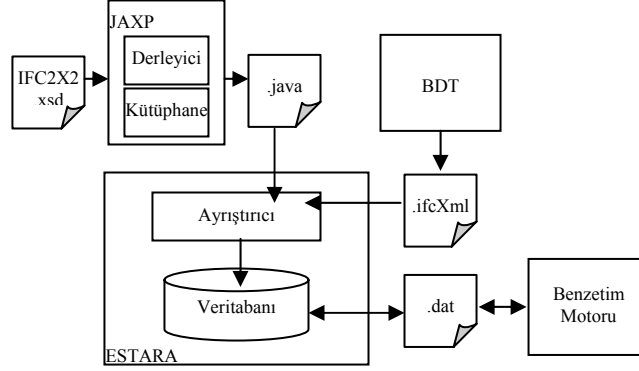
ESTARA programı planlanırken üzerinde durulan en önemli konu başlangıç girdisinin otomatikleştirilmesi olmuştur. Çünkü benzetim programlarında kullanıcıların en çok vakit kayb ettikleri kısım, mekana ait 3 boyutlu geometrik ve geometrik olmayan bilgiyi barındıran başlangıç girdisini oluşturmaktır. ESTARA programında kullanıcıların veri girişini kolaylaştırmak ve gereksiz zaman kaybını engellemek için yeni bir kullanıcı arayüzü geliştirmek yerine, başlangıç girdisinin BDT ortamında oluşturulan proje bilgisinden transferi sağlanmıştır. ifcXML dosya çıktısına sahip BDT programları ile birlikte işlerliğin sağlandığı bu programın yapısı ve bileşenleri aşağıdaki bölümlerde anlatılmıştır.

3.2.1 IFC-Java Kütüphanesinin oluşturulması

JAVA ortamında geliştirilen ESTARA ile oldukça büyük ve karmaşık veri yapısına sahip ifcXML dosyalarını işleyebilmek ve bu dosyalar üzerinde çalışabilmek için ifcXML şemasından türetilen IFC-JAVA kütüphanesi oluşturulmuştur. Bunun için öncelikle IAI tarafından yayınlanan IFC 2x2'e ait ifcXML şemaları (IFC2X2_FINAL.xsd, ex.xsd, configuration.xsd) bir JAXP derleyici yardımı ile derlenmektedir. Derleme işlemi sırasında, şema içinde tanımlanan her bir XML elemanı için uygun Java sınıfları oluşturulmaktadır. Bu işlem sonucunda 2000 den fazla sınıf elde edilmektedir. Daha sonra bu sınıflar JAXP paketinin kütüphanesinde yer alan ve XML dosyasını okuma, analiz, değiştirme gibi işlemleri yapmaya yardımcı olan standart JAVA sınıfları ile birleştirilmektedir. Bu işlemler sonunda ifcXML dosyalarının okunmasına, verilerin analizine ve sorgulanmasına alt yapı oluşturan bir IFC-JAVA kütüphanesi elde edilmektedir. ESTARA programının tüm bileşenleri bu kütüphanedeki sınıflardan yararlanarak işlevlerini yerine getirebilmektedirler.

ESTARA Ayırıştırıcı modülü, proje bilgisini içeren ifcXML dosyasının içinden ESTARA ile ilgili bilgiyi ayıklamaktadır. Bu modül daha sonra ESTARA

nesnelerini oluşturmakta ve bu nesnelerin sahip olduğu bilgiyi parametrik çalışmalar için sistem veritabanına kaydetmektedir. ESTARA bileşenlerinin işleyişi şekil 3.1’de gösterilmiştir.



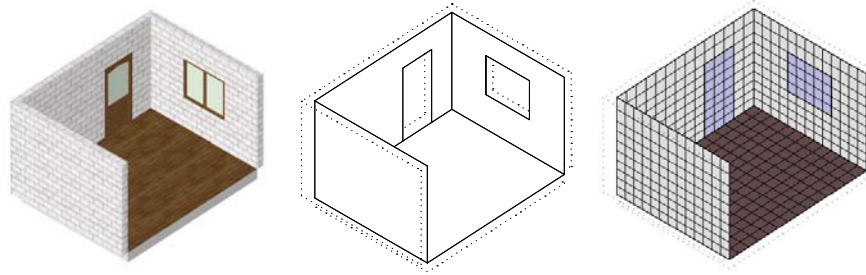
Şekil 3.1 ESTARA bileşenlerinin işleyişi

3.2.2 IfcXML-ESTARA Nesne Modelleri ve Eşleme

Bir hacim akustiği benzetimini gerçekleştirmek için gerekli olan bilgi çok karmaşık değildir. ESTARA'nın hacim akustiği benzetim motorunun veri girişi için ihtiyaç duyulan bilgiler şunlardır:

- 3 boyutlu geometri bilgisi (Tek bir genel başlangıç noktası ile bağıntılı koordinat bilgileri)
 - Sınır elemanları (duvarlar, döşemeler)
 - Açıklıklar (kapılar, pencereler)
 - Sınır elemanları üzerindeki eklentiler (perdeler, panolar)
 - Engeller (alçak bölücü duvarlar, tefriş elemanları)
- Hacmi tanımlayan bileşenlerin malzemeleri
- Hacim içindeki ses kaynaklarının tipleri, yerleri, ve yönleri
- Hacim içindeki alıcı noktaların yerleri.

Bu çalışma da BDT ortamında oluşturulan proje bilgisi, ifcXML dosyası olarak kaydedilmekte ve daha sonra ESTARA Ayrıştırıcı modülü ile bu dosyadan akustik benzetim için gerekli olan bilgi ayıklanmaktadır. Bu projede BDT sistemi olarak kullanılan ArchiCAD'in nesne modellerinin sahip olduğu birçok özelliği IFC modelin içine aktarılamamaktadır. ESTARA Ayrıştırıcı modülü, ifcXML dosyasından ESTARA benzetim motoru için gerekli bilgiyi ayıkladığında modelde bir sadeleştirme daha olmaktadır. Modelde ki bu dönüşüm şekil 3.2'de gösterilmektedir.



Şekil 3.2 ArchiCAD-IFC-ESTARA modelleri

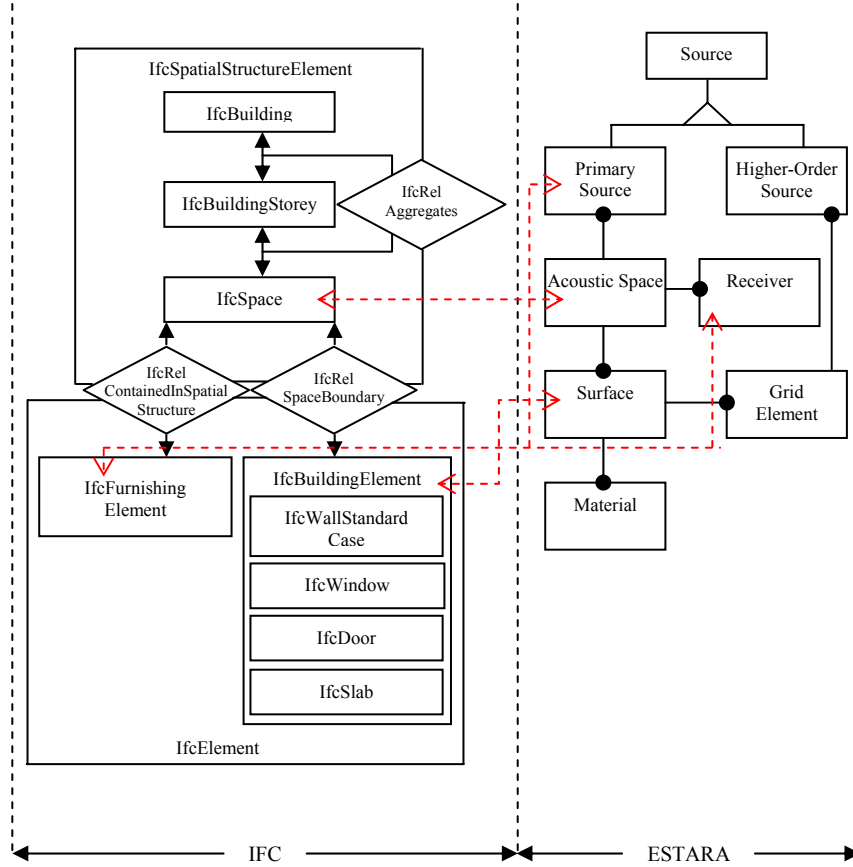
ESTARA ve IFC'nin kısmi nesne modelleri ve gerekli eşleme şekil 3.3'de gösterilmiştir. Bu eşlemeler:

IfcSpace - acousticSpace

IfcBuildingElement- Surface

IfcFurnishingElement (“Speaker”) - PrimarySource

IfcFurnishingElement (“Listener”) - Receiver

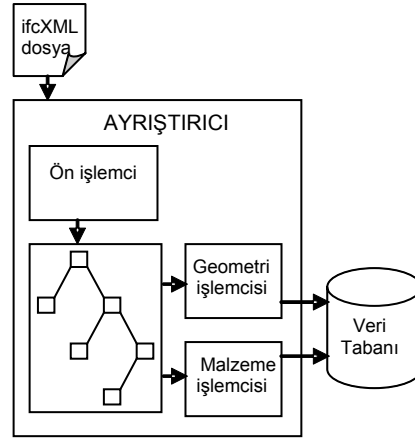


Şekil 3.3 IFC-ESTARA nesne eşleme diyagramı

ESTARA benzetim motoru, hacmin sınırlarını oluşturan duvarların sadece iç yüzeyleri ile ilgilenmektedir. Bundan dolayı Ayrıştırıcı modül tarafından hacmi oluşturan sınırlar olarak sadece yüzeyler yaratılmaktadır. Benzetim motoru yüzeylerin geometrik özelliklerinin yanı sıra yüzey malzemelerinin yansıtıcı ve tutma özelliklerini de kullanmaktadır. Malzemeler ile ilgili bu spesifik özellikler, ESTARA'nın veri tabanında bulunmaktadır. Kullanıcılar nesnelere için malzeme tanımlarını ArchiCAD programında yapmakta ve ifcXML dosyasına taşınan bu bilgi Ayrıştırıcı modül tarafından alınmaktadır. Eğer projede ArchiCAD'in içindeki özel bir malzeme kullanıldıysa ve bu malzeme ESTARA veritabanında bulunmuyorsa, kullanıcı müdahalesi gerekmektedir.

3.2.3 ESTARA Ayırıştırıcı Modül Bileşenleri

ESTARA Ayırıştırıcı modülü geliştirebilir yapıya sahip üç ana bileşenden oluşmaktadır. Hacmi oluşturan nesnelere ifcXML dosyasından ayıklanması ve ESTARA nesnelere ilişkin geometri ve malzeme bilgisinin çıkarılması işlemi ayrı bileşenler tarafından yapılmaktadır. IFC standardı ve IFC uyumlu BDT sistemleri geliştikçe ifcXML şemasında düzeltmeler ve değişiklikler olabilecektir. ESTARA Ayırıştırıcı modülünün bileşenlere ayrılmış yapısı sayesinde bu gibi durumlar da sadece ilgili bileşenin yenilenmesi ihtiyacı karşılayacaktır. ESTARA Ayırıştırıcı modülünün yapısı şekil 3.4’de gösterilmiştir.



Şekil 3.4 ESTARA Ayırıştırıcı Modülü

3.2.3.1 Ön İşlemci

Ön işlemci bileşeni, ifcXML dosyasının okunması ve bu dosyadan ESTARA benzetim motoru için gereken bilgilerin ayıklanması işlemlerini yerine getirmektedir. Hacim ile ilgilenen ESTARA benzetim motorunun veri girişi dosyası hacmi oluşturan yüzeylere, ses kaynaklarına ve alıcılara ait geometri ve malzeme bilgisinden oluşmaktadır. Bu ihtiyaçlar doğrultusunda veri ayıklama sistematığı şu adımlardan oluşmaktadır. (1) Projede bulunan hacimleri (IfcSpace) ayıklamak, (2) Hacimlerin sınırlarını belirleyen yapı nesnelere (IfcBuildingElement) belirlemek ve ayıklamak, (3) Ses kaynaklarını (IfcFurnishingElement) ayıklamak.

Ön işlemci bileşeni bu bilgilerin alınacağı IfcSpace, IFCBuildingElement ve IfcFurnishingElements gibi ilgili IFC nesnelerini ifcXML dosyasından ayıklayarak DOM ağaç yapısı içine yerleştirmektedir. Bu bileşen daha sonra ayıklanan bu nesneleri, ESTARA nesnelere karşılık gelen tanımlamalara dönüştürmektedir. Son olarak ESTARA nesnelerini ifcXML nesnelere gelen “Name” ve “ID” gibi temel özellikler ile eşlemektedir. Bu bileşen;

- ifcXML dosyasını okuyan “read”,
- Hacmin adını bulan “findSpaceName”,
- Toplam yüzey sayısını bulan “findTotalSurfaceNumber”,
- Hacim yüzeylerini ayıklayan “findSpaceBoundary”,
- Ses kaynakları ve alıcıları bulan “findSpaceContained” metotlarına sahiptir.

3.2.3.2 Geometri İşlemcisi

Geometri işlemcisi, ifcXML dosyasının analizini yapan ön işlemci bileşeninden gelen ESTARA nesnelere (yüzeyler, ses kaynakları, alıcılar) ait geometrik bilgiye ulaşmakta ve ESTARA gereksinimlerine göre koordinat bilgisini çevirmektedir. Bu bileşen;

- Hacmi oluşturan yüzeylerin geometri bilgilerini ayıklayan “findGeometryBuildingElement”,
- Ses kaynağı ve alıcı noktalarının geometri bilgilerini ayıklayan “findGeometryFurnishingElement”,
- Koordinatları ESTARA formatına dönüştüren “coordinateConverter” metotlarından oluşmaktadır.

Hacmi oluşturan yüzeylerin geometri bilgilerine ifcxml dosyasından iki farklı yol ile ulaşılabilmektedir. Birinci yol, hacmi oluşturan yapı elemanlarının sınıflarındaki referanslarla geometri bilgilerini çıkartmaktır(IfcStandartWall, IfcSlab, IfcDoor vb). İkinci yol ise hacmin sınırlarını tanımlayan “IfcRelSpaceBoundary” sınıfındaki “ConnectionGeometry” özelliğinden geometri ile ilgili bilgilere ulaşmaktır. İkinci yol ile aynı bilgiye daha az adımda ve daha hızlı ulaşılabildiği için bu projede ikinci yol tercih edilmiştir. findGeometryBuildingElement metodu ile ifcXML dosyasından hacim ve hacmi oluşturan yüzeylerin geometri bilgilerine ulaşmak için izlenen IFC yapısında IfcRelSpaceBoudary sınıfından başlayan ve birbirini takip eden referans sınıflar şekil 3.5’de gösterilmiştir.

Bu metot ile hacmi oluşturan her bir yüzeye ait konum(location), eksen(axis), yön(refDirection) ve koordinat(coordinate) bilgileri bulunmaktadır. Daha sonra bu bilgiler coordinateConverter metodu ile ESTARA formatına uygun koordinatlara dönüştürülmektedir.

Ses kaynağı ve alıcı noktalarının geometri bilgilerine “IfcFurnishingElement” sınıfındaki “ObjectPlacement” ve “Representation” tanımlarından ulaşılmaktadır. findGeometryFurnishingElement Metodu ile ArchiCAD programında ses kaynağı olarak tanımladığımız “Speaker” ve alıcı olarak tanımladığımız “Listener” elemanlarının geometri bilgilerinin yer aldığı ifcXML dosyasındaki IFC sınıfları şekil 3.6’da gösterilmiştir.

Geometri işlemcisi ile ifcXML dosyasından çıkartılan geometri bilgileri yine bu bileşen tarafından ESTARA benzetim motorunun formatına dönüştürülerek ODBC bağlantı yoluyla ESTARA veri tabanına kaydedilmektedir.

IfcRelSpaceBoundary

- Id:
- GlobalId:
- OwnerHistory:
- Name:
- RelatingSpace: *IfcSpace*
- RelatedBuildingElement: *IfcWallStandardCase*
- ConnectionGeometry
 - ↳ IfcConnectionSurfaceGeometry
 - ↳ SurfaceOnRelatingElement
 - ↳ IfcCurveBoundedPlane
 - ↳ Basis Surface
 - ↳ IfcPlane
 - ↳ Position
 - ↳ IfcAxis2Placement3D
 - ↳ Location
 - ↳ IfcCartesianPoint
 - ↳ Coordinates
 - ↳ IfcLengthMeasure:
 - ↳ Axis
 - ↳ IfcDirection
 - ↳ DirectionRatios
 - ↳ ex:double-wrapper:
 - ↳ RefDirection
 - ↳ IfcDirection
 - ↳ DirectionRatios
 - ↳ ex:double-wrapper:
 - ↳ OuterBoundary
 - ↳ IfcCompositeCurve
 - ↳ Segments
 - ↳ IfcCompositeCurveSegments
 - ↳ ParentCurve
 - ↳ IfcPolyLine
 - ↳ Points
 - ↳ IfcCartesianPoint
 - ↳ Coordinates
 - ↳ IfcLengthMeasure:
 - ↳ IfcCartesianPoint
 - ↳ Coordinates
 - ↳ IfcLengthMeasure:
 - ↳ IfcCartesianPoint
 - ↳ Coordinates
 - ↳ IfcLengthMeasure:
 - ↳ IfcCartesianPoint
 - ↳ Coordinates
 - ↳ IfcLengthMeasure:

Şekil 3.5 findGeometryBuildingElement’de Kullanılan IFC Sınıfları

IfcFurnishingElement

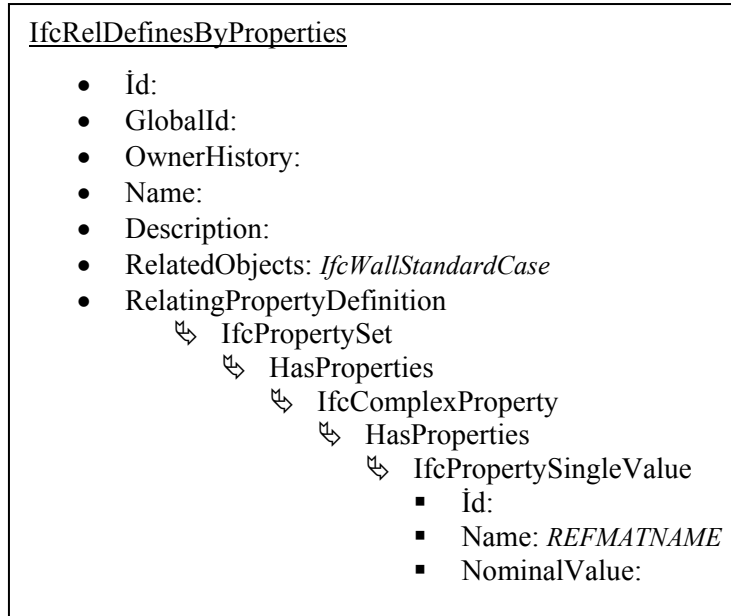
- Id:
- GlobalId:
- OwnerHistory:
- Name: *Speaker*
- RelatingSpace: *IfcSpace*
- ObjectPlacement
 - ↪ IfcLocalPlacement
 - ↪ PlacementRelTo
 - ↪ RelativePlacement
 - ↪ IfcAxis2Placement
 - ↪ Location
 - ↪ IfcCatesianPoint
 - ↪ Coordinates
 - ↪ IfcLengthMeasure:
 - ↪ Axis
 - ↪ IfcDirection
 - ↪ DirectionRatios
 - ↪ ex:double-wrapper:
 - ↪ RefDirection
 - ↪ IfcDirection
 - ↪ DirectionRatios
 - ↪ ex:double-wrapper:
- Representation
 - ↪ IfcProductDefinitionShape
 - ↪ Representations
 - ↪ IfcShapeRepresentation
 - ↪ ContextofItems
 - ↪ RepresentationIdentifier: *Body*
 - ↪ RepresentationType: *Brep*
 - ↪ Items
 - ↪ IfcFacetedBrep
 - ↪ Outer
 - ↪ IfcClosedShell
 - ↪ CfsFaces
 - ↪ IfcFace
 - ↪ Bounds
 - ↪ IfcFaceOuterBound
 - ↪ Bound
 - ↪ IfcPolyLop
 - ↪ Polygon
 - ↪ IfcCartesianPoint

Şekil 3.6 findGeometryFurnishingElement’de Kullanılan IFC Sınıfları

3.2.3.3 Malzeme İşlemcisi

Malzeme işlemcisi ön işlemci bileşeni ile ayıklanan IFC nesnelерinin malzeme bilgilerine ulaşmakta ve bu malzemelerin karşılıklarını ESTARA malzeme veri tabanından alarak gerekli eşlemeleri yapmaktadır. Bu işlemden sonra Malzeme işlemcisi ilgili ESTARA nesnelерine malzeme atamasını yapmakta ve nesnelер ESTARA veri tabanına kaydedilmektedir.

Bu bileşen yapı elemanlarına ait malzeme tanımlarına “findMaterialElement” metodunu kullanarak ulaşmaktadır. Malzeme tanımları İfcXML dosyasında “IfcRelDefinesByProperties” nesnesinde yer almaktadır. Bu sınıfta yer alan “RelatedObjects” ve “RelatingPropertyDefinition” tanımları sayesinde “hangi yapı elemanı hangi malzeme ile ilişkilendirilmiştir” bilgisine ulaşılabilir. Bu nesne yardımı ile yapı elemanlarının ön, arka ve yan yüzeylerine ait malzeme bilgileri ayrı elde edilmektedir. “findMaterialElement” metodunda yapı elemanlarına ait malzeme tanımlarına ulaşmak için takip edilen IFC sınıfları şekil 3.7’de gösterilmiştir.



Şekil 3.7 findMaterialElement’de Kullanılan IFC Sınıfları

3.2.3.4 Sistem Veri tabanı

Ön işlemci bileşeni ile ayıklanan IFC nesnelere, ESTARA nesnelere dönüştürülüp, Geometri işlemcisi ve Malzeme işlemcisi tarafından gerekli bilgilerin atanması yapıldıktan sonra bu nesnelere sistem veritabanına gönderilmektedirler. Bu programda veritabanı olarak Microsoft Access kullanılmıştır. Veritabanı ile program arasındaki iletişim ODBC bağlantı yolu tanımlanarak sağlanmıştır.

Son olarak veritabanında ki veriler akustik benzetim motorunun girdi dosyası olan “.dat” formatına dönüştürülmektedir ESTARA programının girdi dosyasında yer alması gereken bilgiler şekil 3.8’de gösterilmiştir:

| |
|-------------------------------|
| <u>Hacim Bilgisi:</u> |
| ’mekanın adı’ |
| ’mekanın hacmi’ |
| ’toplam yüzey sayısı’ |
| yüzey bilgisi: |
| ’kenar sayısı’ |
| geometri bilgisi: |
| ’x y z’ |
| yönü: |
| ’r s t’ |
| ’malzemesi’ |
| ’sınır elemanı mı?’ |
| ’yüzey alanı’ |
| ’boşluk yada eklenti var mı?’ |
| ses kaynağı bilgisi: |
| ’kaynak sayısı’ |
| konumu: |
| ’r s t’ |
| yönü: |
| ’alfa beta’ |
| alıcı bilgisi: |
| ’noktasal mı? düzlem mi?’ |
| yüzey ise: |
| ’düzlem yüzey bilgisi’ |

Şekil 3.8 ESTARA veri dosyası

3.2.4 Kısıtlar

Oluřturulan modelin ESTARA girdi gereksinimleri ile uyumlu bir model olup olmadığının kontrolü ArchiCAD programı tarafından yapılmaz. Bu sebepten dolayı kullanıcıların ArchiCAD programında aőağıda belirtilen sınırlar içinde çalışması beklenmektedir.

- Hacmin temsili için bir “Zone” tanımlanması gerekmektedir.
- Hacim sınırları mutlaka kapanmalıdır.
- Kütüphaneye “Speaker” adıyla tefriő elemanı eklenmeli ve ses kaynakları için bu eleman kullanılmalıdır.
- Kütüphaneye “Listener” adıyla tefriő elemanı eklenmeli ve alıcılar için bu eleman kullanılmalıdır.

4. SONUÇLAR VE ÖNERİLER

Bu çalışmada Balıkesir Üniversitesi'nde geliştirilmekte olan hacim akustiği analiz programı ESTARA için gereken veriyi; BDT ortamlarındaki tasarım bilgisini barındıran ifcXML dosyalarından ayıklayarak hazırlayan bir yazılım geliştirilmiştir. Bu sayede öğrenciler tasarımlarını alışık oldukları BDT ortamında hazırladıktan sonra ifcXML dosyası olarak ESTARA'ya aktarabilmektedirler. Öğrenmeleri gereken ESTARA arayüzü ise üç boyutlu tasarım araçlarından arındırılmış olarak sadece parametrik çalışmalara yönelik olarak basitleşmiş olacaktır. Öğrenciler yeni bir tasarım aracı öğrenmekle vakit kaybetmeden hacim akustiği konuları üzerine yoğunlaşabileceklerdir. Bu yaklaşım aynı zamanda ESTARA için karmaşık, üç boyutlu veri girişini destekleyen bir arayüz gereksinimini ortadan kaldırmaktadır.

Bu çalışmada, birbirinden farklı biçimdeki veriler ile çalışan BDT programı ile ESTARA programı arasındaki veri transferi IFC platformu üzerinden sağlanmıştır. IFC'nin veri transfer mekanizmalarından biri olan ifcXML dosya biçimi sayesinde XML tabanlı olan tasarım bilgisinin sorgulanması ve ayrıştırılması işlemleri başka araçlara gerek kalmaksızın Java-XML teknolojileri ile gerçekleştirilmiştir. Java ortamında geliştirilen yazılım, BDT sistemi olarak seçilen ArchiCAD programında modellenen ve ifcXML dosya formatında kaydedilen proje bilgisinden gerekli verileri ayıklamakta, ESTARA formatına dönüştürmekte ve veritabanına kaydetmektedir.

ifcXML IFC'nin diğer mekanizmaları ile karşılaştırıldığında hem STEP/EXPRESS araçlarını gerektirmemesi hem de dosyaların anlaşılabilirliği açısından bu projenin geliştirilmesinde önemli bir kolaylık sağlamıştır. Buna karşılık ifcXML'in iki dezavantajı büyük dosya boyutları ve EXPRESS'ten otomatik dönüşüm sırasında oluşan kayıplardır. ESTARA'nın büyük binaların ayrıntılı tasarımlarına değil basit hacimlere yönelik olması ve IFC modelindeki temel nesnelerin akustik analizlerin veri gereksinimlerini karşılayabilmesinden dolayı bu dezavantajlar bu çalışmada sorun olmamıştır.

IFC ile çalışmanın teknik sorunlarına ifcXML iyi bir alternatif sunuyor olsa bile, dosyalarda bulunan veri temelde IFC yapısında olduğundan IFC'nin yakından tanınması gerekmektedir. Bunun için de belli bir EXPRESS bilgisi kaçınılmaz olmaktadır oysa günümüzde Java ve XML ile çalışan çoğu ekibin veri modelleme için tercihi UML olmaktadır. Gelecekte EXPRESS – UML dönüşümünün ele alınması yararlı olacaktır.

Bir birlikte işlerlik platformu olarak IFC'nin gelişiminde halen ele alınmamış bir diğer konuda analiz programlarının sonuçlarının tasarım ortamına geri besleme yapamamasıdır. Bu açıdan IFC halen tek yönlü bir veri aktarımı öngörmektedir ve bu çalışmada da veri transferi ArchiCAD'den ESTARA'ya doğru olmak üzere sadece tek yönlü yapılmaktadır. Bu durumda hacim akustiği analiz sonuçları BDT ortamında görselleştirilememektedir. Gelecekteki çalışmaların akustik analiz sonuçlarının tasarım verisi ile bütünleşmesi ve BDT ortamında görselleştirilmesine yönelik olması gerçek anlamda bir birlikte işlerliğe doğru ilk adımları oluşturacaktır.

KAYNAKLAR

- [1] Smith, B., Roger, N. and Wellington, J., “IGES-Initial Graphics Exchange Specification”, Autofact III: Conference Proceedings, Detroit, Michigan, (1981).
- [2] ISO TC 184/SC4, “Contribution of West Germany, VDAFS”, N3, International Standards Organization, Geneva, Switzerland, (1984).
- [3] ISO TC184/SC4, “Contribution of France, SET: Standard for Exchange and Transfer”, N10, International Standards Organization, Geneva, Switzerland, (1984).
- [4] D. Appleton Company, Inc., “Development Plan for a Product Data Exchange Specification, PDES Version 1.0” D-669-87-03, South Carolina Research Authority, South Carolina, (1987).
- [5] Smith, B. and Rinaudot, G., “Product Data Exchange Specification: First Working Draft”, NISTIR 88-4004, National Institute of Standards and Technology, Gaithersburg, Maryland, (1988).
- [6] IAI, “Industry Foundation Classes IFC2x Edition 3 Online Documentation”, URL: http://www.iai-international.org/Model/R2x3_final/index.htm, (2006).
- [7] Fischer, M., Kam, C., “PM4D Final Report” CIFE Technical Report #143, CIFE - Stanford University, Stanford, (2002).
- [8] Nytsch-Geusen, C., Klempin, C., Nunez von Voigt, J. and Radler, J., “Integration of CAAD, Thermal Building Simulation and CFD by Using The IFC Data Exchange Format”, Eighth International IBPSA Conference, Eindhoven, Netherlands, (2003).
- [9] Yang, Q.Z. and Xu, X., “Design Knowledge Modeling and Software Implementation for Building Code Compliance Checking”, *Building and Environment*, 39, (2004), 689-698.
- [10] O’Sullivan, B. and Keane, M., “Specification of an IFC Based Intelligent Graphical User Interface to Support Building Energy Simulation”, Ninth International IBPSA Conference, Montreal, Canada, (2005), 875-881
- [11] Liebich, T., Nispert, N., “ifcXML Implementation Guide”, version 1.0, International Alliance for Interoperability, Washington, D.C., (2005), 48.
- [12] Hoskins, E. M., “Computer Aids in System Building”, *Computer-Aided Design*, North-Holland Press, (1973), 127-140.

- [13] Eastman C.M. and Henrion M. "GLIDE: A Language for Design Information Systems", *ACM SIGGRAPH Computer Graphics*, 11/2, (1977), 24-33.
- [14] Borkin H., McIntosh J. and McIntosh P., "ARCH:Model - A Geometric Modeling Relational Database System, ARCH:Model User's Manual", Architectural Research Laboratory, University of Michigan, Ann Arbor, Michigan, (1981).
- [15] Fenves S. J., Flemming U., Hendrickson C., Maher M.L. and Schmitt G., "An Integrated Software Environment for Building Design and Construction", Proceedings of the Fifth Conference on Computing in Civil Engineering, American Society of Civil Engineers, Alexandria, Virginia, (1988).
- [16] Luiten, G., Froese, T., Björk, B.C., Cooper, G., Junge, R., Karstila, K. and Oxman, R., "An information reference model for architecture, engineering, and construction", Proceedings of the First International Conference on the Management of Information Technology for Construction, Singapore, (1993), 391– 406.
- [17] Junge, R. and Liebich, T., "Product Modeling for Applications", Models for Next Generation CAAD in Proceedings of sixth International Conference on Computing in Civil Engineering, Rotterdam, (1995).
- [18] Björk, B.C., "Requirements and information structures for building product data models", VTT Publications No 245, Espoo Technical Research Centre of Finland (VTT), Finland, (1995).
- [19] Alshawi M. and Underwood J., "Improving the constructability of a design solution through an integrated system", *the International Journal of Engineering, Construction and Architectural Management*, 3(1&2), (1996), 47-67.
- [20] Amor R.W., Clift M., Scherer, R., Katranuschkov, P., Turk, Z. and Hannus, M., "A Framework for Concurrent Engineering – ToCEE", European Conference on Product Data Technology, CICA, Sophia Antipolis, France, (1997), 15-22.
- [21] Aouad G., "Integration: From a modelling dream into an implementation reality", Proceedings of CIB W55, International support for building economics, Lake District, UK, (1997) , 7-29.
- [22] Eastman C.M., "Building Product Models: Computer Environments Supporting", CRC Press, Boca Raton, Florida (1999), 424.
- [23] Goldstein, B., Kemmerer, S., and Parks, C., "A Brief History of Early Product Data Exchange Standards", NISTIR 6221, National Institute of Standards and Technology, Gaithersburg, (1998).
- [24] Sun, M. and S. Lockley., "Data exchange system for an integrated building design system", *Automation in Construction*, 6, (1997), 147-155

- [25] Mortenson, M.E., "Geometric Modelling", John Wiley & Sons, New York, (1985).
- [26] Farin, G., Hahmann, S., Brunnett, G., "Geometric Modelling", *Computing* : 72. 2004, 1/2, Dagstuhl 2002, N. - Wien : Springer, (2004), 246.
- [27] Hvam, L., Riis, J. and Hansen, B.J., "CRC Cards for Products Modelling", *Computer in Industry*, 50, (2003), 57-70.
- [28] Shlaer, S., Mellor, S., "Object-Oriented Systems Analysis: Modelling the World in Data", Yourdon Press, Englewood Cliffs, New Jersey, (1988).
- [29] Booch, G., "Object Oriented Design with Applications", The Benjamin/Cummings Publishing Company, Inc., California, (1991), 608.
- [30] Türk, Z., "Object Oriented Modelling Techniques and Integrated CAD" CIB W78 & CBS meeting in Montreal, Quebec, Canada, (1992).
- [31] Björk, B.C., "A Unified Approach for Modeling Construction Information", *Building and Environment*, 27/ 2, (1992), 173-194.
- [32] Wilson, P.R., "A Short History of CAD Data Transfer Standard", *IEEE Computer Graphics and Applications*, 7/6, (1987), 64-67.
- [33] Mayer, Ralph J., "IGES.", *Byte* 12/6, (1987), 209-214.
- [34] Owen, J., "STEP: An Introduction", Information Geometers Ltd., UK, (1993).
- [35] ISO TC184/SC4, "Industrial automation systems and integration-Product data representation and exchange-Part 1: Overview and fundamental principles", ISO 10303-1, International Standards Organization, Geneva, Switzerland, (1994).
- [36] Fowler, J. "STEP for data management exchange and sharing", Technology Appraisals Ltd., Great Britain, (1995).
- [37] Wellington, J., and Bradford S., "ISO TC184/SC4 Reference Manual", NISTIR 5665, National Institute of Standards and Technology, June (1995).
- [38] ISO TC184/SC4, "Industrial automation system and integration-Product data representation and exchange-Part 11: Description methods: The EXPRESS language reference manual EXPRESS language reference manual", ISO 10303-11, International Standardisation Organisation, Geneva, Switzerland, (1997)
- [39] Schenck, D.A., Wilson, P.R., "Information Modelling: The EXPRESS Way", Oxford University Press, Oxford, (1994).

- [40] ISO TC184/SC4, “Industrial automation systems and integration-Product data representation and exchange-Part 21: Implementation methods: Clear text encoding of the exchange structure”, ISO 10303-21, International Standards Organization, Geneva, Switzerland, (2002).
- [41] ISO TC184/SC4, “Industrial automation systems and integration-Product data representation and Exchange-Part 22: Implementation Methods: Standard Data Access Interface”, ISO 10303-22, International Standards Organization, Geneva, Switzerland, (1996).
- [42] ISO TC184/SC4, “Industrial automation systems and integration-Product data representation and exchange-Part 23: Implementation Methods: C++ Programming Language Binding to the Standard Data Access Interface Specification”, ISO 10303-23, International Standards Organization, Geneva, Switzerland, (1997).
- [43] ISO TC184/SC4, “Industrial automation systems and integration-Product data representation and exchange-Part 24: Implementation Methods: C Programming Language Binding to the Standard Data Access Interface Specification”, ISO 10303-24, International Standards Organization, Geneva, Switzerland, (1996).
- [44] ISO TC184/SC4, “Industrial automation systems and integration-Product data representation and Exchange-Part 25: Implementation methods: EXPRESS to XMI binding”, ISO 10303-25, International Standards Organization, Geneva, Switzerland, (2003).
- [45] ISO TC184/SC4, “Industrial automation systems and integration-Product data representation and Exchange-Part 27: Implementation methods: Java programming language binding to the standard data access interface with Internet/Intranet extensions”, ISO 10303-27, International Standards Organization, Geneva, Switzerland, (1999).
- [46] ISO TC184/SC4, “Industrial automation systems and integration-Product data representation and Exchange-Part 28: Implementation methods: XML representations of EXPRESS schemas and data ”, ISO 10303-28, International Standards Organization, Geneva, Switzerland, (2004).
- [47] IAI, “A Short History of the IAI and the IFC Information Model”, URL: <http://www.iai-international.org/About/History.html>
- [48] Liebich,T., Wix, J., “IFC Technical Guide”, International Alliance for Interoperability, Washington, D.C., (2000), URL: http://www.iai-international.org/Model/documentation/IFC_2x_Technical_Guide.pdf
- [49] Liebich, T., “IFC 2x Edition 2 Model Implementation Guide” v.1.7, International Alliance for Interoperability, Washington, D.C., (2004), URL:

http://www.iai-international.org/Model/files/20040318_Ifc2x_ModelImplGuide_V1-7.pdf

[50] Nisbet N., Liebich T., “ifcXML Implementation Guide”, International Alliance for Interoperability, Washington, D.C., (2005), URL: http://www.iai-international.org/ifcXML2/RC2/IFC2X2_FINAL/ifcXML_Implementation_Guide_v1-0.pdf

[51] Liebich, T., “XML schema language binding of EXPRESS for ifcXML”, MSG-01-001(Rev 4), International Alliance for Interoperability, Washington, D.C., (2001).

[52] IAI, “IfcXML2 Release Candidate 2”, URL: <http://www.iai-international.org/Model/IfcXML2.htm>

[53] <http://www.w3c.org>

[54] Harold, E.R., “XML Bible”, IDG Books Worldwide Inc., United States of America, (1999).

[55] DuCharme, B., “XML: The Annotated Specification”, Prentice Hall, Upper Saddle River, New Jersey (2000), 368.

[56] Matthias, N. and Jasmi, J., “XML Parsing: A Threat to Database Performance” In Proceedings of CIKM'03, New Orleans, Louisiana, (2003), 175-178.

[57] Goldfarb, C.F., “XML Handbook”, Prentice Hall PTR, New Jersey, (2001).

[58] <http://www.XML.org>

[59] <http://www.graphisoft.com/support/ifc/>

[60] <http://java.sun.com/>

[61] Morgenthal, J.P., “Portable Data/Portable Code: XML & Java Technologies”, Sun Microsystems Inc., URL: <http://java.sun.com/xml/ncfocus.html>

[62] <http://java.sun.com/webservices/jaxb/index.jsp>

[63] <http://java.sun.com/webservices/jaxp/index.jsp>

[64] http://www.altova.com/products/xmlspy/xml_editor.html