

STEP VERİ DÖNÜŞÜM FORMATINDA 3 BOYUTLU PARÇA TEMSİLİ

Adem ÇİÇEK

Düzce Üniversitesi, Teknik Eğitim Fakültesi, Makine Eğitimi Bölümü,
Konuralp Yerleşkesi 81620, DÜZCE. Tel: 380 5413344 (1070), adecicek@yahoo.com

ÖZET

Grafik standartları farklı BDT ve BDİ sistemleri arasında veri dönüşümünü sağlayan yapılardır. STEP grafik standardı BDT/BDİ entegrasyonu için esas olan geometrik, topolojik ve teknolojik veriyi temsil etme yeteneğine sahiptir. Bu makale BDT ortamında tasarlanan 3 boyutlu parçaların STEP grafik standardında nasıl temsil edildiğini hiyerarşik bir şekilde açıklamaktadır. Aynı zamanda STEP formatının başlıca veri yapısı ve öğeleri incelenmektedir.

Anahtar Kelimeler - STEP grafik standardı, 3 Boyutlu model, BDT/BDİ

3D PART REPRESENTATION IN THE STEP DATA EXCHANGE FORMAT

ABSTRACT

Graphic standards are structures which provide data exchange between different CAD and CAM systems. STEP graphic standard is capable of representing geometric, topological and technological data which is essential for CAD/CAM integration. This paper hierarchically explains how to represent 3D parts designed in any CAD environment in the STEP graphic standard. The main data structure and entities of STEP format is investigated.

Key Words - STEP graphic standard, 3D model, CAD/CAM

1. GİRİŞ

BDT ve BDİ'nin entegrasyonu günümüzde önemli bir araştırma konusu olup bu problemin çözümüne yönelik çok sayıda çalışma yapılmaktadır. Her bir BDT sistemi bir parçanın geometrik modeli için kendi iç temsil tekniklerini ve veri tabanı yapısını kullanmaktadır. Bu da farklı BDT ve BDİ sistemleri arasında veri iletişiminin sağlanamaması anlamına gelmektedir. Bundan dolayı, BDT ve BDİ sistemleri arasında veri iletişimini sağlayacak bir ara yüz gereksinimi ortaya çıkmıştır. Bu gereksinimi karşılamak amacıyla özellikle 1980'li yıllarda tasarım ve imalatta standardizasyon çalışmaları başlatılmıştır. Bu çalışmalar sonucunda günümüze kadar bir çok grafik standardı geliştirilmiştir. Grafik standartları, parça çizim veya modeline ait geometrik ve/veya topolojik bilgileri belirli bir formatta kaydeden ve farklı BDT veya BDİ sistemleri arasında veri iletişimini sağlayan yapılar olarak tanımlanabilmektedir. Bundan dolayı, grafik standartları BDT/BDİ entegrasyonunun temel öğelerinden olan BDT/BDİ uygulamaları için büyük önem taşımaktadır. Bununla birlikte, grafik standartları çeşitli tasarım ve üretim çalışmaları için algoritma geliştirme çalışmalarında CAD

modellerinden veri elde edilmesi amacıyla da kullanılmaktadır. Standartların kullanımıyla, bir fabrika yada organizasyonda halihazırda kullanılmakta olan çeşitli yazılım ve donanımlar değişik uygulamalar için kullanılabilirliği gibi yeni yazılım ve donanımlar da rahatlıkla mevcut sisteme adapte edilebilmektedir. Günümüzde DXF, IGES, SET, VDA-FS, CAD*I, STEP gibi geliştirilen grafik standartları algoritma geliştirmede ve veri dönüşümünde halen yaygın olarak kullanılmaktadır. STEP standardını diğerlerinden ayıran en belirgin özellikler uluslar arası düzeyde desteklenen, 2 ve 3 boyutlu BDT modellerinin veri dönüşümünü sağlayan ve diğer grafik standartlarının tüm fonksiyonlarını içeren bir veri dönüşüm formatı olmasıdır [1-3]. Çiçek [4-5] parça tanıma yaklaşımı için yüz tabanlı ilişki matrisi olarak adlandırılan yeni bir parça tanımlama formatı geliştirmiştir. Geliştirilen matematiksel model, parçaları hem geometrik hem de topolojik veri açısından tanımlamaktadır. Yüz komşuluk ilişki matrisi olarak adlandırılan matematiksel modelde parçaya ait yüzey bağlanma ilişkilerinin yanı sıra parça üzerindeki tüm yüzeylere ait nitelikler de açıkça temsil edilmiştir. Matrisi oluşturmak için BDT ortamında tasarlanan parçanın STEP dosyası girdi olarak

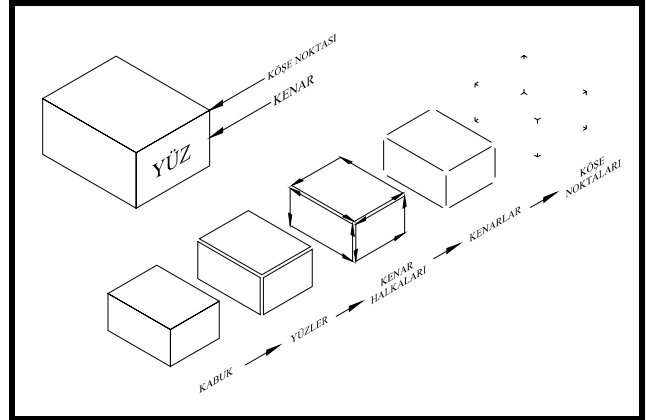
kullanılmıştır. Literatürde yaygın olarak kullanılan parça tanımlama şemalarının aksine, bu çalışmada geliştirilen yüz tabanlı ilişki matrisi ile parça temsil formatı hem basitleştirilmiş hem de bilgisayar formatına uygun hale getirilmiştir. Gülesin ve Jones [6-8] veritabanındaki bitmiş, orta düzey ve ham parçaları temsil etmek için yüz tabanlı komşu grafiği (FONG-Face Oriented Neighbouring Graph) olarak adlandırılan bir parça modeli temsil şeması geliştirmişlerdir. STEP dosyası kullanarak yüz ve unsurlar tayin edilmiş ve yüzler arasındaki açılar hesaplanmıştır. Aynı zamanda, yüzler arasındaki komşuluk, içbükeylik ve dışbükeylik ilişkileri saptanarak parça modeli kısa ve özlü bir şekilde temsil edilmiştir. Bu temsil formatı daha sonra bilgisayar destekli işlem planlama ve bağlama kalıbı tasarımı için kullanılmıştır. Dereli ve Filiz [9] 3 boyutlu parçalar üzerindeki unsurları tanımak için STEP tabanlı bir unsur tanıma sistemi geliştirmiştir. Sistemin önemli karakteristiklerinden iki tanesi, parçanın bitişiklik ilişkileri tabanlı olması ve girdi olarak modelin STEP bilgisini kullanmasıdır. Parçanın bitişik yüzleri arasındaki ilişkiler bulunmakta ve bu ilişkiler bir ilişki matrisi içinde saklanmaktadır. Bu matrisin elemanların izleyerek unsurlar önce çıkarılmakta ve sonra bu unsurlar veritabanında mevcut olan unsur tipleri ile karşılaştırılarak unsurlar tanımlanmaktadır. Bhandarkar ve Nagi [10] bir standart tabanlı biçim unsuru çıkarma sistemi geliştirerek STEP standardını desteklemişlerdir. Geliştirilen unsur çıkarma sistemi parçanın geometrisini ve topolojisini tanımlayan STEP AP 202 dosyasını girdi olarak kabul etmekte ve çıktı olarak biçim unsuru tabanlı işlem planlama için STEP AP 224 formatında biçim unsuru bilgisi ile bir STEP dosyası üretmektedir. Algoritma frezeleme operasyonları ile işlenen düzlemsel ve silindirik yüzeyler gibi birincil biçimleri içeren prizmatik katılar için geliştirilmiştir. Algoritma sırası ile önce boşluk unsurlarını sonra iç biçim unsurlarını sonra da dış biçim unsurlarını en son ise bağlantı düzenleyici unsurları çıkarır. Son olarak bu unsurların geometrisi ve topolojisi ayrı AP 224 öğelerine dönüştürülerek çıktı STEP dosyası üretilmiştir. El-Mehalawi ve Miller [11] veritabanındaki BDT modeli için bir temsil şeması geliştirmiştir. Parçalar, parçanın yüzlerini karşılayan düğümler ve parçanın kenarlarını karşılayan zincirleri içeren nitelikli grafikler kullanılarak temsil edilmiştir. Grafik, bir veri dönüşüm formatı olan parçanın STEP fiziksel dosyası yardımıyla hazırlanmaktadır. Bir parçanın grafiğini oluşturmak için işlemler iki aşamada gerçekleştirilmiştir. İlk aşamada, bir BDT sisteminde BDT modeli oluşturulmuş ve BDT modeli STEP part 21 formatına dönüştürülmüştür. İkinci aşamada ise STEP bilgisi temsil tabanlı nitelikli grafiklerine haritalanmıştır.

Literatür çalışmalarından anlaşılacağı gibi STEP standardı farklı BDT ve BDİ sistemleri arasında veri dönüşümünde kullanılmasının yanı sıra BDT tabanlı yazılım çalışmalarında da yaygın olarak kullanılmaktadır. Buda BDT ve BDİ'nin entegrasyonu için STEP standardının önemini artırmaktadır. Bu makalede, BDT ve BDİ sistemleri arasında veri dönüşümünde yaygın olarak kullanılan STEP grafik standardında 3 boyutlu parçaların hiyerarşik temsili açıklanmaktadır. STEP grafik standardında temsil edilen öğeler ve veri dönüşüm formatının veri yapısı

tanımlamak için 3 boyutlu bir örnek katı model ele alınarak en üst düzey STEP öğesi olan kapalı kabuk öğesinden en basit bir temsil olan nokta öğesine kadar tüm STEP öğeleri detaylı bir şekilde ele alınmıştır.

II. STEP VERİ DÖNÜŞÜM FORMATI

STEP (Standard for the Exchange of Product Model Data) grafik standardı, bir tasarımı mamul haline getirmek için gerekli bütün işlemlerin ve bağlı parametrelerin standardize edilmesini ve tanımlanmasını; hiçbir uygulama, tasarım ve üretim yazılımına bağlı kalmadan gerçekleştirebilmeyi sağlamak için tasarlanmıştır. Bütün diğer standartları bünyesinde toplayan STEP standardı, bu standartların aksine geometrik veri dönüşümünün yanı sıra tolerans ve yüzey kalitesi gibi teknolojik üretim bilgilerinin ve topolojik unsur ilişkilerinin tanımını da içermektedir. STEP yapısını diğerlerinden ayıran diğer bir özellik ise esnek ve dinamik bir yapıya sahip olmasıdır [1]. Bir STEP dosyası "ISO-10303-21;" satırı ile başlayıp "END-ISO-10303-21;" satırı ile sonlanmaktadır. Burada, ISO-10303 STEP grafik standardının uluslararası standart numarası, 21 ise STEP standardının bölüm numarasını temsil etmektedir. Bir STEP dosyası HEADER ve DATA bölümleri olmak üzere iki bölüme ayrılmaktadır. Tüm mamul verisi DATA bölümünde temsil edilmiştir. BDT sistemlerinin çoğu iç veri yapısı olarak B-Rep'in (Boundary Representation) bazı formlarını kullanmaktadır. Bu sistemlerin her birinin detaylı veri yapısı farklıdır. B-Rep'in genel veri yapısı tabanlı bir BDT ortamında temsili Şekil 1'de gösterilmiştir.



Şekil 1. B-Rep'te bir katının temsili için veri yapısı

Şekil 1'de gösterildiği gibi B-Rep'te 3 boyutlu parça geometrisi kabuk, yüzler, kenar halkaları, kenarlar ve köşe noktaları ile temsil edilmektedir. STEP formatında ise parça geometrisinin temsili manifold B-Rep üzerine dayandırılmıştır. Manifold katı B-Rep "topolojik temsil öğelerinin, referans eğrilerinin, yüzeylerin ve noktaların birkaç katmanı vasıtasıyla bir geometrik temsil" olarak tanımlanmıştır. Bu temsil ve daha önce gösterilen basit B-Rep arasındaki asıl fark geometrik öğelerin açıkça tanımlanmasıdır. Manifold B-Rep aynı zamanda topolojik bilgiyi geometrik bilgi ile birleştirmiştir [11].

STEP formatında her bir ögenin önünde rakam işareti (#) olan öge tanımlayıcısı bulunmaktadır. Yani, bu formatta her bir öge ayrı bir satırda temsil edilerek rakam işareti (#) ile başlamaktadır. Aynı zamanda, rakam işaretini takip eden her bir ögenin öge numarası vardır ve tüm ögeler açık bir şekilde tanımlanana kadar öge içerisinde farklı öğelere atıf yapılmaktadır.

```
#257=LINE(",#254,#256);
```

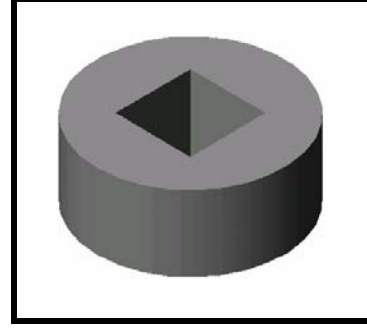
Örnek olarak yukarıdaki STEP ögesini tanımlamak gerekirse, rakam işaretini (#) takip eden "257" rakamı, "LINE" ögesinin öge numarasını temsil etmektedir. Ayrıca, parantez içerisindeki "254" ve "256" rakamları ise "LINE" ögesinin geometrisinin tanımlanması için gerekli olan öge numaralarını temsil etmektedir. "LINE" ögesi için ileri bir tanımlama yapmak için bu öge numarasına sahip olan STEP öğelerini tespit etmek gerekmektedir. STEP standardında parça tanımlama, en yüksek düzey kapalı kabuk tanımlamasından başlamakta ve en temel geometrik öğeler olan nokta ve yön tanımlamalarında son bulmaktadır. Dolayısıyla STEP standardı en yüksek düzeyden en düşük düzeye kadar ürün tanımlamasını belirli bir hiyerarşide temsil etmektedir. STEP standardında ürün tanımlaması EXPRESS dili ile yapılmaktadır. STEP standardı için tasarlanan EXPRESS dili güçlü bir veri modelleme dilidir. Yapı olarak PASCAL diline benzemektedir. STEP standardının en çok kullanıldığı alanlar otomotiv, uzay bilimleri ve özellikle BDT tabanlı yazılım geliştirme sektörü olarak söylenebilir. Bu bölümde STEP standardında 3 boyutlu bir parçayı tanımlamak için temsil edilen öğeler ve bu öğeler arasındaki ilişkiler tanımlanacaktır.

II.1. Kapalı Kabuk Temsili

STEP standardında parça tanımlamanın en yüksek düzeyi kabuktur. Bir kabuk kenarlar boyunca yüzeylerin katılmasıyla oluşturulan topolojik öğedir. Alanı birleşik, yönlü, sonlu ve yüzeyleri kendi aralarında kesişmezler. Alanı ile birlikte bir kabuk STEP fiziksel dosyasında tanımlanır. Kabuk kendisini oluşturan yüzlerin koleksiyonu ile temsil edilir. Örneğin Şekil 2'deki örnek parçanın kapalı kabuk tanımı aşağıdaki gibi olacaktır.

```
P=CLOSED_SHELL(",(Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8));
```

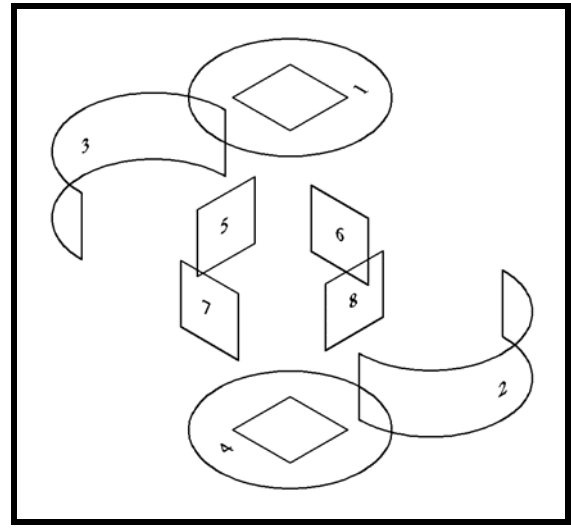
Burada P, kapalı kabuk (CLOSED_SHELL) ögesinin öge numarasını, parantez içindeki öge numaraları ise sırasıyla Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8 (Şekil 3) yüzlerinin öge numaralarını temsil etmektedir. Bu öge numaraları ile STEP dosyasındaki ilgili yüz temsillerine ulaşarak yüz temsilleri hakkında bilgi almak mümkündür.



Şekil 2. Örnek parça

II.2 Yüz Temsili

Tanımlamanın ikinci düzeyi yüzdür. Bir yüz, kenar halkaları ile sınırlanan yüzeyin bir parçası olan topolojik öğedir. Örneğin, Şekil 2'de örnek parçaya ait yüzeylerin STEP formatında temsili Şekil 3'te gösterilmiştir. STEP formatında düzlem yüzeylerin aksine silindirik, konik, küresel ve toroid yüzeyler iki özdeş yüzle temsil edilmiştir. Şekil 3'te gösterildiği gibi örnek parçanın 2 ve 3 numaralı yüzleri aynı silindirik yüzeyi biçimlendirirken silindirik yüzler için bir kenar halkası oluşturmak için iki bağımsız yüz olarak temsil edilmiştir. İki özdeş silindirik yüzün STEP standardında temsil edilen geometri ve topolojisi aynıdır.



Şekil 3. Örnek parçaya ait yüzeylerin STEP formatında temsili

STEP formatında bir yüz "ADVANCED_FACE" ögesi ile temsil edilmiştir. Şekil 3'teki parçanın 1 numaralı yüzey için tanımlama aşağıdaki gibidir.
Y1 = ADVANCED_FACE (" (YDS,YS),YT);

Burada YDS parametresi, Şekil 3'deki 1 numaralı yüzü sınırlayan yüz dış sınırına (dış kenar halkası) işaret eden bir değişkendir. Aynı zamanda, yüz dış sınırının kapsadığı unsurlar var ise bunlar yüz sınırı (YS) olarak ifade edilmiş olup 1 numaralı yüzün iç kenar halkasını tanımlamaktadır. Yüz dış sınırı her bir yüz için bir tane iken, yüz sınırları (iç kenar halkaları) yüzün içerdiği

unsurlara bağlı olarak bir veya birden fazla olabilmektedir. Bir yüz sınırı, yüzü sınırlamada kullanılmak için tasarlanan bir kenar halkasıdır. Yüzün dış sınırını tanımlamanın ilave anlamsallığını taşıyan yüz dış sınır çizgileri olarak adlandırılan ikincil bir tipi vardır. STEP standardında YDS ve YS değişkenleri aşağıdaki gibi tanımlanmaktadır.

```
YDS=FACE_OUTER_BOUND (" ,KH);  
YS=FACE_BOUND (" ,KH1);
```

Burada KH ve KH1 yüz dış sınırı ve yüz sınırını çevreleyen kenar halkası (EDGE_LOOP) ögesine atf yaparak öge numarasını vermektedir. YT ise 1 numaralı yüzeyin yüzey tipi tanımlanmaktadır. Yüzey tipi tanımlaması şu şekildedir.

```
YT=PLANE(" ,YEY);
```

“PLANE” ögesi ilgili yüzeyin düzlemsel yüzey olduğunu ifade ederek yüzey tipini tanımlamaktadır. YEY ise yüzeye ait STEP formatında tanımlanan lokal koordinat sisteminin orijinine, x eksen yönüne ve z eksen yönüne atf yapan “AXIS2_PLACEMENT_3D” ögesini temsil etmektedir.

Ayrıca farklı yüzey tipi temsilleri için tanımlamalar aşağıdaki gibidir.

```
YT=CYLINDRICAL_SURFACE(" ,YEY, R);  
YT =SPHERICAL_SURFACE(" ,YEY, R);
```

STEP formatında silindirik ve küresel yüzeyler için eksen yerleştirme ögesinin yanında R ile bu yüzeylerin yarıçapları verilmiştir.

```
YT =CONICAL_SURFACE(" ,YEY, R, KA);
```

Konik yüzeyler için yarıçap R ile, koniklik açısı ise KA ile temsil edilmiştir.

```
YT =TOROIDAL_SURFACE(" ,YEY, MAXR, MİNR);
```

Toroid yüzeyler ise silindirin döndüğü eksenin yarıçapı MAXR ile, silindir yarıçapı ise MİNR değişkeni ile tanımlanmıştır.

STEP standardında global koordinat sisteminin yanı sıra her bir yüze ait bir lokal koordinat sistemi tayin edilmiştir. Lokal koordinat sistemi YEY ögesi ile temsil edilmiştir.

```
YEY = AXIS2_PLACEMENT_3D ( “ ”, KSO, KZ, KX  
)
```

Burada KSO, yüze ait lokal koordinat sisteminin orijini karşılayan noktaya işaret eder. KZ, lokal koordinat sisteminin z ekseninin yönünü ve KX, lokal koordinat sisteminin x ekseninin yönünü belirtir.

```
KSO=CARTESIAN_POINT(" ,(NX,NY,NZ));
```

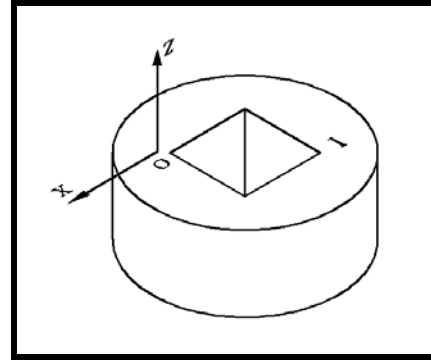
NX, NY, NZ lokal koordinat sisteminin orijininin sırasıyla X, Y ve Z koordinat değerlerini vermektedir.

```
KZ=DIRECTION(" ,( YX,YY,YZ));
```

YX, YY, YZ lokal koordinat sisteminin z ekseninin yönünü göstermektedir.

```
KX=DIRECTION(" ,( YX,YY,YZ));
```

YX, YY, YZ lokal koordinat sisteminin x ekseninin yönünü göstermektedir. Lokal koordinat sisteminin x ve y eksen yönleri parçanın global koordinat sistemine göre verilmiştir. Lokal z eksen, düzlemsel ve küresel yüzeylerin normalini, silindirik, toroid ve konik yüzeylerin eksen yönlerini göstermektedir. “CARTESIAN_POINT” ve “DIRECTION” ögeleri STEP standardının en temel ögeleri olup, bu ögelerde başka ögelere atf yapılmayıp doğrudan nokta ve yön tanımlamaları yapılmaktadır.



Şekil 4. STEP formatında yüzey temsili

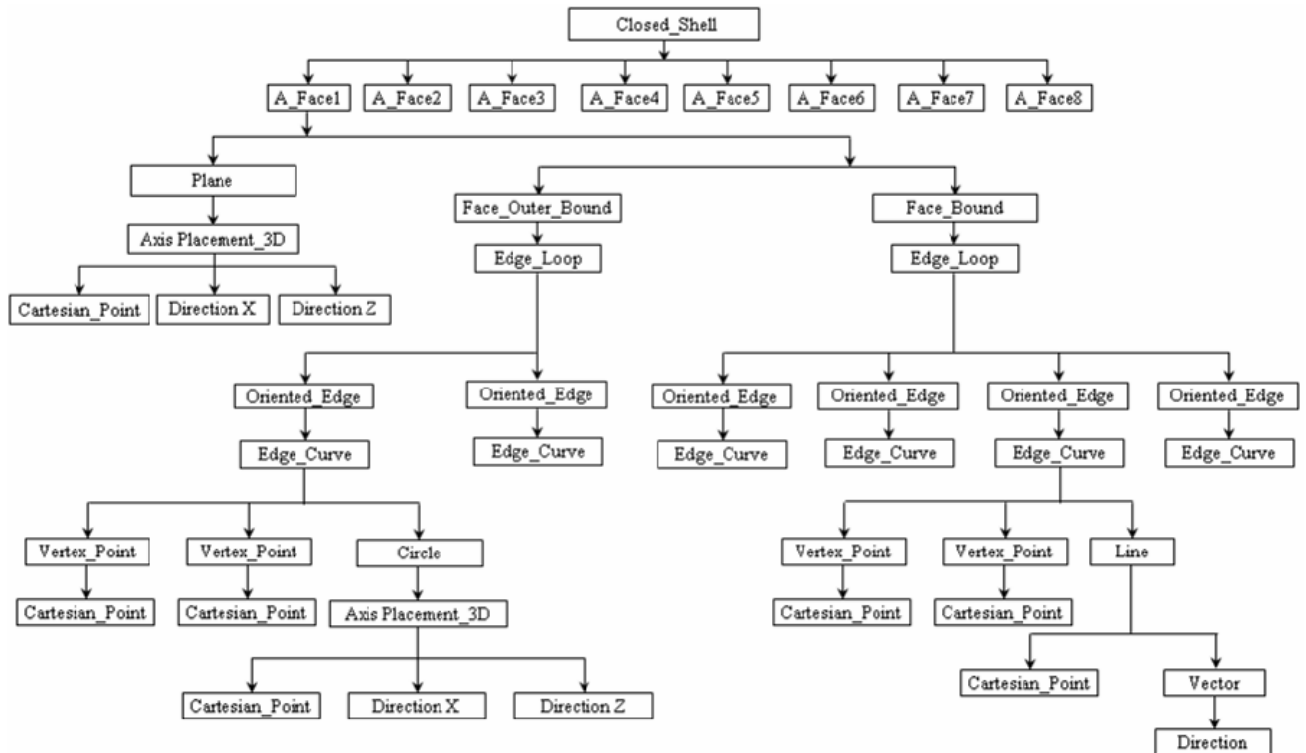
Şekil 4’te örnek parçanın 1 numaralı yüzeyi için lokal koordinat sistemi gösterilmiştir. Şekilden de anlaşılacağı gibi lokal koordinat sisteminin z eksen yönü düzlemsel yüzeyin yüzey normalinin yönünü karşılamaktadır. Bu şekilde, 3 boyutlu modelleri biçimlendiren yüz tiplerinin STEP standardında bir lokal koordinat sistemi vardır. Her bir yüze ait lokal koordinat sisteminin yardımıyla yüzlerini birbirine ve global koordinat sistemine göre konumları tespit edilebilmektedir. Ayrıca, düzlemsel yüzeyler arasındaki açılar hesaplanabilmekte ve yüzlerin birbirine göre içbükeylik ve dışbükeylik durumları tespit edilebilmektedir.

```
#130=CARTESIAN_POINT(",(8,0,4,5,0,0));
#131=VERTEX_POİNT("#130);
#140=CARTESIAN_POINT(",(3,0,4,5,0,0));
#141=VERTEX_POİNT("#140);
#142=CARTESIAN_POINT(",(5,5,4,5,0,0));
#143=DIRECTION(",(0,0,0,0,1,0));
#144=DIRECTION(",(1,0,0,0,0,0));
#145=AXIS2_PLACEMENT_3D(",#142,#143,#144);
#146=CIRCLE(",#145,2,5);
#147=EDGE_CURVE(",#131,#141,#146,T.);
#182=CARTESIAN_POINT(",(6,5,5,5,0,0));
#183=VERTEX_POİNT("#182);
#190=CARTESIAN_POINT(",(6,5,3,5,0,0));
#191=VERTEX_POİNT("#190);
#192=CARTESIAN_POINT(",(6,5,3,5,0,0));
#193=DIRECTION(",(0,0,1,0,0,0));
#194=VECTOR(",#193,2,0);
#195=LINE(",#192,#194);
#196=EDGE_CURVE(",#191,#183,#195,T.);
#221=CARTESIAN_POINT(",(4,5,3,5,0,0));
#222=VERTEX_POİNT("#221);
#223=CARTESIAN_POINT(",(4,5,3,5,0,0));
#224=DIRECTION(",(1,0,0,0,0,0));
#225=VECTOR(",#224,2,0);
#226=LINE(",#223,#225);
#227=EDGE_CURVE(",#222,#191,#226,T.);
#252=CARTESIAN_POINT(",(4,5,5,5,0,0));
#253=VERTEX_POİNT("#252);
#254=CARTESIAN_POINT(",(4,5,5,5,0,0));
#255=DIRECTION(",(0,0,-1,0,0,0));
#256=VECTOR(",#255,2,0);
#257=LINE(",#254,#256);
#258=EDGE_CURVE(",#253,#222,#257,T.);
#281=CARTESIAN_POINT(",(6,5,5,5,0,0));
#282=DIRECTION(",-1,0,0,0,0,0);
#283=VECTOR(",#282,2,0);
#284=LINE(",#281,#283);
#285=EDGE_CURVE(",#183,#253,#284,T.);
#305=CARTESIAN_POINT(",(5,5,4,5,0,0));
#306=DIRECTION(",(0,0,0,0,1,0));
#307=DIRECTION(",(1,0,0,0,0,0));
#308=AXIS2_PLACEMENT_3D(",#305,#306,#307);
#309=CIRCLE(",#308,2,5);
#310=EDGE_CURVE(",#141,#131,#309,T.);
#331=CARTESIAN_POINT(",(5,5,4,5,0,0));
#332=DIRECTION(",(0,0,0,0,1,0));
#333=DIRECTION(",(1,0,0,0,0,0));
#334=AXIS2_PLACEMENT_3D(",#331,#332,#333);
#335=PLANE(",#334);
#336=ORIENTED_EDGE(",*,#147,F.);
#337=ORIENTED_EDGE(",*,#310,F.);
#338=EDGE_LOOP(",#336,#337);
#339=FACE_OUTER_BOUND(",#338,T.);
#340=ORIENTED_EDGE(",*,#196,T.);
#341=ORIENTED_EDGE(",*,#285,T.);
#342=ORIENTED_EDGE(",*,#258,T.);
#343=ORIENTED_EDGE(",*,#227,T.);
#344=EDGE_LOOP(",#340,#341,#342,#343);
#345=FACE_BOUND(",#344,T.);
#346=ADVANCED_FACE(",#339,#345)#335,F.);
#347=CLOSED_SHELL(",(#166,#206,#237,#268,#290,#314,#330,#346));
#348=MANIFOLD_SOLID_BREP("90,#347);
```

Şekil: 5. 1 numaralı yüzeye ait öğelerin STEP formatında temsili

Şekil 5'te 1 numaralı yüzeye ait STEP dosyasında tanımlanan tüm öğeler verilmiştir. STEP dosyasında tüm öğeler hiyerarşik olarak tanımlanmıştır. Fakat Şekil 5'te de görüldüğü gibi bu öğeler gruplar halinde farklı öğe numaraları ile temsil edilmektedir. Buda STEP

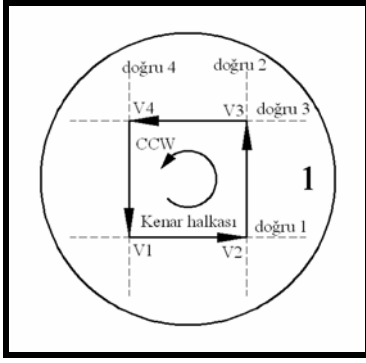
dosyasında öğelerin hiyerarşik takibini oldukça zorlaştırmaktadır. Bundan dolayı, STEP dosyasından 3 boyutlu CAD model verisini elde etmek için algoritma kurmak oldukça zordur. Şekil 6'da 1 numaralı yüzeyin hiyerarşik STEP veri yapısı ağacı gösterilmiştir.



Şekil: 6. Örnek parça ve 1 numaralı yüzey için hiyerarşik STEP veri yapısı ağacı

II.3. Kenar Halkası ve Kenar Temsili

Bir kenar halkası aynı köşe noktalarında başlayan ve biten, yönlü kenarlardan oluşan topolojik bir öğedir. Alanı kapalı bir eğridir ve STEP formatında her bir kenar halkası yüzey çevresinde saat ibresine ters yönde (CCW) yönlendirilmiştir. Örnek parçadaki 1 numaralı yüzey üzerindeki iç kenar halkasının yönü Şekil 7’de gösterilmiştir.



Şekil: 7. Örnek parçadaki 1 numaralı yüzey üzerindeki iç kenar halkası

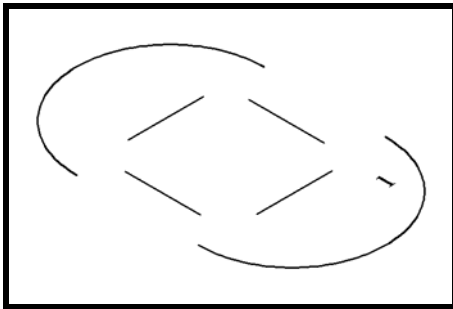
STEP standardında kenar halkası tanımlaması aşağıdaki gibidir.

$KH1 = \text{EDGE_LOOP} (“,(YK1, YK2, YK3, YK4);$

YK1, YK2, YK3, YK4, yönlü kenar tanımlamasına işaret etmektedir. Yönlü kenar tanımlaması ise STEP standardında aşağıdaki gibi temsil edilmiştir.

$YK1 = \text{ORIENTED_EDGE} (“, KE1);$
 $YK2 = \text{ORIENTED_EDGE} (“, KE2);$
 $YK3 = \text{ORIENTED_EDGE} (“, KE3);$
 $YK4 = \text{ORIENTED_EDGE} (“, KE4);$

Burada KE1, KE2, KE3, KE4 kenar eğrisi (EDGE_CURVE) tanımlamasını göstermektedir. Şekil 8’de örnek parçanın 1 numaralı yüzeyine ait kenarların STEP formatında temsili verilmiştir. Şekilden de anlaşılacağı gibi STEP formatında doğru öğesinin aksine çember öğesi iki özdeş yay olarak temsil edilmektedir.



Şekil: 8. Örnek parçanın 1 numaralı yüzeyine ait kenarların STEP formatında temsili

III.4. Kenar Eğrisi ve Köşe Noktası Temsili

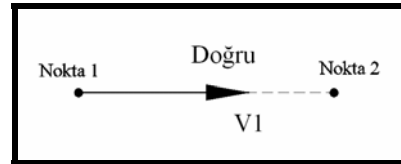
Bir kenar eğrisi koordinat düzleminde hareket eden bir noktanın takip ettiği yol olarak tanımlanabilir. STEP formatında kenar eğrilerinin bazıları doğru, yay, elips, b_spline eğri ve sınırlı eğri olarak sınıflandırılmıştır. Kenar iki köşe noktasını birleştiren topolojik bir öğedir. Kenarın sınırları iki köşe noktasıdır. Kenar ilk köşe noktasından ikincisine eğri tipine göre bir yön seçerek yönlendirilirler. Tüm kenarlar STEP fiziksel dosyasında birer yönlü kenar ile temsil edilmiştir. Kenar tanımlama geometrik öğelere dayanarak verilmiştir. Örneğin:

$KE1 = \text{EDGE_CURVE} (“, BAN1, BİN1, KET1);$

Burada BAN1, kenar eğrisini başlangıç noktası, BİN1 ise kenar eğrisinin bitiş noktasına tekabül eden “VERTEX_POINT” öğesini temsil etmektedir. KET1 ise KE1 kenar eğrisinin eğri tipi olup “LINE” öğesine atıf yapmaktadır. Kenar eğrisinin başlangıç ve bitiş noktaları aşağıdaki gibi tanımlanmıştır.

$BAP = \text{CARTESIAN_POINT} (“,(BAX,BAY,BAZ));$
 $BAN1 = \text{VERTEX_POINT} (“,BAP);$
 $BİP = \text{CARTESIAN_POINT} (“,(BİX,BİY,BİZ));$
 $BİN1 = \text{VERTEX_POINT} (“,BİP);$

Burada BİP ve BAP kenar eğrisinin başlangıç ve bitiş köşe noktalarını göstermektedir. BAX, BAY, BAZ ve BİX, BİY, BİZ sırasıyla kenar eğrisinin başlangıç ve bitiş noktalarının global koordinat sisteminde x, y ve koordinatlarını vermektedir. Şekil 9’da gösterildiği gibi doğru vektörünün başlangıç noktasını nokta 1 ve bitiş noktasını ise nokta 2 karşılamaktadır.



Şekil: 9. STEP formatında doğrunun temsili

Aynı zamanda, STEP standardında doğru vektörü orijini, büyüklüğü ve yönü ile temsil edilmiştir.

$KET1 = \text{LINE} (“,VOP, VEK);$

Burada “LINE” öğesi kenar eğrisi tipinin doğru olduğunu göstermektedir. VEK doğru vektörü tanımlamasını, VOP ise doğru vektörünün orijini temsil etmektedir. STEP standardında doğru vektörünün orijini, büyüklüğü ve yönü aşağıdaki gibi temsil edilmektedir.

$VOP = \text{CARTESIAN_POINT} (“,(VOPX,VOPY,VOPZ));$
 $VY = \text{DIRECTION} (“,(VOX,VOY,VOZ));$
 $VEK = \text{VECTOR} (“,VY,VB);$

Burada VB doğru vektörünün rakamsal büyüklüğünü temsil etmektedir. VY ise doğru vektörünün yönüne (“DIRECTION” öğesine) atıf yapmaktadır.

“DIRECTION” ögesinde temsil edilen VOX,VOY,VOZ değişkenler ise doğru vektörünün yönünü x, y ve z eksenlerine göre tanımlanmaktadır. VOPX,VOPY,VOPZ doğru vektörünün orijininin global koordinat sisteminde x, y ve z koordinatlarını vermektedir.

STEP formatında temsil edilen diğer bir kenar eğrisi tipi yaydır. Yay “CIRCLE” ögesi ile temsil edilmektedir.

KET2=CIRCLE(",KEY,R);

Burada KET2 ögesi tanımlanan kenar eğrisinin yay olduğunu ifade ederek kenar eğrisi tipini tanımlar. KEY ise yüzeye ait STEP formatında tanımlanan lokal koordinat sisteminin orijini, x eksen yönü ve z eksen yönünü vermektedir. R ise yay kenar eğrisinin yarıçapını tanımlamaktadır. Kenar eğrisi için eksen yerleştirme ögesi STEP standardında aşağıdaki gibi temsil edilmiştir.

KEY=AXIS2_PLACEMENT_3D ("KSO,KZ,KX);

Burada KSO, kenar eğrisine ait lokal koordinat sisteminin orijininin karşılaman noktaya işaret eder. KZ, lokal z ekseninin yönünü ve KX, lokal x ekseninin yönünü belirtir.

KSO=CARTESIAN_POINT(",(NX,NY,NZ));

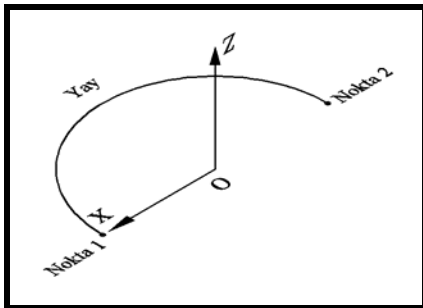
NX, NY, NZ lokal koordinat sisteminin orijininin sırasıyla X, Y ve Z koordinatını göstermektedir.

KZ=DIRECTION(",(YX,YY,YZ));

YX, YY, YZ lokal koordinat sisteminin z ekseninin yönünü göstermektedir.

KX=DIRECTION(",(YX,YY,YZ));

Burada YX, YY, YZ lokal koordinat sisteminin x ekseninin yönünü göstermektedir. x ve y eksenlerinin yönleri parçanın global koordinat sistemine göre verilmiştir. Lokal z eksen, yay ve elips kenar eğrilerinin eksen yönlerini göstermektedir.



Şekil: 10. STEP formatında yayın temsili

Şekil 10’da yay kenar eğrisi için lokal koordinat sistemi gösterilmiştir. Şekilden de anlaşılacağı gibi lokal koordinat sisteminin z eksen yönü yay eksenini karşılamaktadır. STEP formatında temsil edilen diğer bir kenar ögesi de eliptir. Elips STEP formatında aşağıdaki

gibi temsil edilmektedir.

KET3=ELLIPSE(",KEY,MAXR,MİNİR);

Burada kenar eğrisinin lokal koordinat sisteminin tanımlanmaktadır. Elipse ait lokal koordinat sistemi STEP formatında yaya ait lokal koordinat sistemi ile aynı şekilde tanımlanmaktadır. Fakat MAXR ve MİNİR parametreleri sırasıyla elipsi oluşturan küçük ve büyük yayın yarıçapını temsil etmektedir.

STEP ortamındaki öğelerin BDT ortamındaki öğeler ile aynı fakat tanımlama metodlarının farklı olduğu görülmektedir. Bununla birlikte, farklı BDT sistemleri kendine özgü temsil yaklaşımları tabanlı farklı metodlarla aynı geometrik öğeyi tanımlayabilir. STEP ortamında öğeler yüz, halka, kenar setleri, köşe noktası setleri olarak tanımlanmaktadır.

III. SONUÇ

BDT ve BDİ sistemlerinin entegrasyonunun sağlanması ve BDT/BDİ sistemlerinde parça verisinin etkin bir şekilde paylaşılabilmesi için nötr bir veri formatına halen ihtiyaç duyulmaktadır. Bu makalede, 3 boyutlu parçaların STEP dosyasında hiyerarşik temsiliyi açıklayarak BDT/BDİ uygulamalarında algoritma geliştirecek kullanıcılara STEP formatının kullanımını kolaylaştırmak için bir çalışma yapılmıştır. STEP formatı uluslararası alanda algoritma geliştirme için sık kullanılan bir veri dönüşüm formatı olduğundan dolayı, örnek bir katı model ele alınarak STEP temsili en üst düzey temsili olan kapalı kabuk ögesinden en basit temsili nokta ve yön temsiline kadar STEP öğelerinin açılımı yapılarak bundan sonra yapılacak algoritma geliştirme çalışmalarına katkı sağlanmıştır.

KAYNAKLAR

- [1] Dereli, T., Filiz, İ. H., Bilgisayar destekli tasarım ve üretim sistemlerinde grafik standartları, MATİK'97-Makine Tasarım Teorisi ve Modern İmalat Yöntemleri Konferansı, Ankara, 1997.
- [2] Yıldız, S., Aslan, E., STEP standardı içinde express tanımlama diliyle mekanik parçaların modellenmesi, MATİK'97-Makine Tasarım Teorisi ve Modern İmalat Yöntemleri Konferansı, Ankara, 1997.
- [3] Han, J. H., Kang, M., Choi, H., STEP-based feature recognition for manufacturing cost optimization, Computer Aided Design, 33, 671-686, 2001.
- [4] Çiçek, A., Parça tanıma yaklaşımı için geliştirilen bir parça temsil formatı, Politeknik Dergisi, Cilt. 9, Sayı. 3, Sayfa. 189-195, 2006.
- [5] Çiçek, A., Bilgisayar destekli parça tanıma sisteminin geliştirilmesi ve motor montajına uygulanması, Doktora Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 2005
- [6] Gülesin, M., An intelligent knowledge based process planning and fixturing system using the

- STEP standard, PhD Thesis, Coventry University, 1993.
- [7] Gülesin, M., 3-D part definition in the STEP standard for process planning, MATİK'97-Makine Tasarım Teorisi ve Modern İmalat Yöntemleri Konferansı, Ankara, 1997.
- [8] Gülesin, M., Jones, R. M., An expert process planning and fixturing system for prismatic parts using STEP Standard, 6th International Machine Design and Production Conference, Ankara, 1994.
- [9] Dereli, T., Filiz, İ. H., A note on the use of STEP for interfacing design to process planning, Computer Aided Design, 34, 1075-1085, 2002.
- [10] Bhandarkar, M. P., Nagi, R., STEP-based feature extraction from STEP geometry for agile manufacturing, Computers in Industry, 41, 3-24, 2000.
- [11] El-Mehalawi, M., Miller, R. A., A database system of mechanical components based on geometric and topological similarity. Part I: representation, Computer Aided Design, 35, 83-94, 2003.