



## REPRESENT OF MECHANICAL PARTS USING STEP STANDARDS AT COMPUTER ENVIRONMENT

İ. ÇELİK\* & S. YALDIZ\*\* & A. ÜNÜVAR\*\*

\*Dumlupınar Üniversitesi, Mühendislik Fakültesi, Makine Müh. Bölümü, Kütahya, Türkiye  
[icelik@dumlupinar.edu.tr](mailto:icelik@dumlupinar.edu.tr)

\*\*Selçuk Üniversitesi, Teknik Bilimler MYO., Makine Bölümü, Konya, Türkiye  
[syaldiz@selcuk.edu.tr](mailto:syaldiz@selcuk.edu.tr)

\*\*Selçuk Üniversitesi, Mühendislik Fakültesi, Makine Müh. Böl., Konya, Türkiye  
[aunuvar@selcuk.edu.tr](mailto:aunuvar@selcuk.edu.tr)

### ABSTRACT

Currently, the development of Computer Aided Manufacturing systems is still a major field where extensive researches are being made. These researches suggest that there are certain problems with the full integration of CAD and CAM systems. These problems result from CAD system, fully automated Computer Aided Process Planning system and Computer Aided Manufacturing Interface. In recent years, the feature-based design approach in CAD systems has made an important progress and has supported to the integration of CAD/CAM systems. In this paper, the representation of a mechanical part having a hole and two steps with the feature-based design concept, using STEP standard format, has been handled in computer environment.

**Keywords:** Computer Aided Manufacturing, Computer Aided Design, Feature Based Modelling, STEP Standards.

## STEP STANDARTI KULLANARAK MEKANİK PARÇALARIN BİLGİSAYAR ORTAMINDA TEMSİL EDİLMESİ

### ÖZET

Günümüzde Bilgisayarla Tümlşik İmalat Sistemlerinin geliştirilmesi hala üzerinde yoğun araştırmaların sürdürüldüğü önemli bir konudur. Araştırmalar Bilgisayar Destekli Tasarım (BDT) ile Bilgisayar Destekli Üretim (BDÜ) sistemlerinin tam olarak bütünleştirilmesinde birtakım eksikliklerin olduğunu göstermektedir. Bunlar; BDT sistemi, tam otomatik Bilgisayar Destekli İşlem Planlama (BDİP) sistemi ve BDÜ ara yüzü olarak sıralanabilir. Son yıllarda BDT sistemlerindeki özellik tabanlı tasarım yaklaşımı büyük bir gelişme göstererek BDT/BDÜ sistemlerinin bütünleşmesinde önemli bir fayda sağlamıştır. Bu çalışmada özellik tabanlı tasarım kavramıyla üzerinde delik ve kademe özellikleri bulunan mekanik bir parçanın STEP standardı kullanılarak bilgisayar ortamında temsil edilmesi ele alınmıştır.

**Anahtar Kelimeler:** Bilgisayar Destekli İmalat, Bilgisayar Destekli Tasarım, Özellik Tabanlı Modelleme, STEP Standartları

## 1.GİRİŞ

Günümüzde makine parçalarının tasarım işlemlerini kolaylaştıran gelişmiş BDT sistemleriyle, işlem planlamacıların görevlerini kolaylaştıran bir çok BDÜ sistemi vardır. Fakat bu sistemler birbirlerinden bağımsızdırlar [1]. Mevcut tüm BDT sistemleri, kendi veri tabanlarında, kendilerine özgü matematiksel temsil tekniklerine sahip olmaları nedeniyle, bu sistemler birbirleriyle doğrudan iletişim kurmakta sorunlara sahiptir [2]. BDT sistemleri arasındaki iletişim sorununu ortadan kaldırmak üzere, Başlangıç Grafikleri Değiştirme Özelliği (IGES), SET, VDA-FS gibi bazı tabii standart formatlar geliştirilmiştir [3]. Bu standartlar içerisinde en iyi bilinen IGES, BDT sistemleri arasında veri transferi yapmak üzere yaygın bir şekilde kullanılmaktadır [2]. Bununla beraber bu tip standartlar, BDT ve BDİP/BDÜ sistemleri arasındaki iletişim problemlerini tam olarak çözmemiştir. Grafik standartları, BDT ve BDÜ sistemleri arasında veri transfer etmekten ziyade, orijinal olarak BDT sistemleri arasında geometrik verileri transfer etmek üzere geliştirilmiştir. Grafik standartların en büyük eksikliği, sadece şekil özelliği tanımlamalarıdır. Halbuki BDİP sistemleri- şekil özellikleri yanında- toleranslar, yüzey pürüzlülüğü, boyutlar ve malzeme bilgisi gibi diğer teknolojik bilgilere de ihtiyaç duyarlar. Bundan dolayı grafik standartlar, komple bir ürünü tanımlamada yetersizdir. Mevcut standartların ihtiyaçlara cevap vermedeki yetersizliği yüzünden, IGES/PDES organizasyonu tarafından yeni bir standart olan Ürün Verileri Değiştirme Özelliği (PDES) tasarlanmıştır. Bu yeni standart halen Ürün Model Verileri Değiştirme Standardı (STEP) adıyla geliştirilmektedir. STEP'in 1.0 versiyonu 1988' de yayımlanmış olup, ISO tarafından uluslararası grafik verileri standardının ilk taslak teklifi olarak kabul edilmiştir [4]. STEP komple bir ürünün tanımlanması için gerekli olan geometri, topoloji, şekil özelliği, tolerans, yüzey pürüzlülüğü ve malzeme gibi geometrik ve teknolojik bilgileri birlikte içermektedir. Günümüz BDİP sistemlerinde kullanılan Özellik Tanıma yaklaşımı, teknolojik bilgi desteği sağlamaz, çünkü BDT modelleri, bu tür bilgileri içermez. Özellik Tabanlı Tasarım, Özellik Tanıma prosedürünü ortadan kaldıran yeni bir yaklaşımdır [5]. Bu metotla ürünler, ortak erişilebilir ve kolayca yorumlanabilen veri yapıları olarak hazırlanabilir ve otomatize edilmiş BDİP sistemi ile, tam olarak Bilgisayarla Tümüleşik Üretim (BTÜ) başarılabılır [6]. Özellik tabanlı tasarım sistemlerinde tasarım özellikleri imalat varlıkları olarak adlandırılırlar ve işlem planlama sistemlerinde kullanılırlar [7]. Parçalar özellik tabanlı tasarım yaklaşımını kullanarak tanımlandıktan sonra, STEP gibi belirli bir formatta ortak veri tabanında depolanır. Böylece BDT, BDİP ve BDÜ sistemleri ürün verilerine doğrudan erişebilir. Standart STEP dosya yapısındaki bilgi, doğrudan işlem planlama, modelleme, montaj ve imalat uygulamalarında kullanılabilir [8]. Özellik tabanlı tasarım önemli derecede gelişmiştir. Bu yüzden gelecekte parça tanımlamada, ilk önce parçanın katı modeli oluşturulacak ve sonra imalat özellikleri tanımlanacak veya oluşturulan katı modelden imalat özellikleri çıkartılacaktır [9]. Amaç ürünün özellikler olarak veri değişiminin sağlanması ve özellikler hakkında mantıksal ilişkiyi yakalamak, taşımak ve veri değişimi için fiziksel bir mekanizma sağlamaktır. 1980 li yıllardan itibaren bu çaba uluslar arası bir katılıma dönüştü [10].

## 2. STEP STANDARTLARI İLE MEKANİK PARÇALARIN TANIMLANMASI

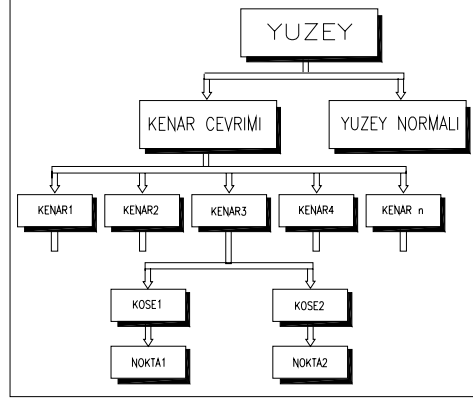
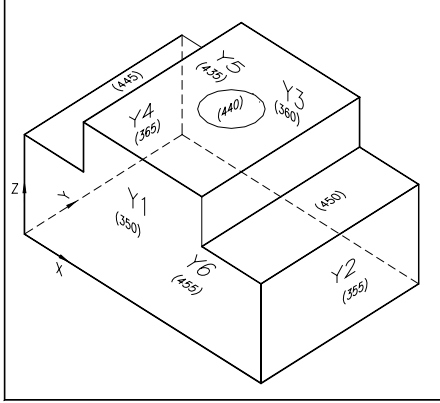
Mekanik parça bilgisi, EXPRESS dili kullanılarak doğrudan fiziksel dosyada temsil edilir. EXPRESS dili ürün bilgisinin tanımlanmasının ve değişiminin resmi tarifnamesini sağlayan bir ISO standardı olarak STEP çalışmaları içinde geliştirildi [11]. Bu dil ürünün yaşam döngüsü içindeki her aşamayı özel bilgi modelleri olarak temsil edebilecek şekilde tasarlandı. EXPRESS, karmaşık bilgiyi düzgün ve tam olarak temsil etmek için gerekli sözdizimini ve anlamsallığı sağlar [12]. EXPRESS dili işlem modellemesini destekleyen zengin bir yapı setine sahiptir. Bir SUPERTYPE-SUPTYPE hiyerarşisi şeklinde seviyeler ve şekillendirilebilen özellikler ifade edilebilir [13]. Bir STEP dosyası, STEP ile başlar ve ENDSTEP ile sona erer. STEP dosyası, HEADER ve DATA kısmı olmak üzere iki bölümden meydana gelir. Ürün verileri DATA bölümünde temsil edilir. Bu bölümde Şekil 1’de verilen örnek parçanın, STEP’te temsil edilmesi açıklanmıştır. Örnek parçanın, Sınır Temsili (ST) modellemesi kullanan bir BDT sisteminde tasarlanmış olduğu kabul edilmiştir. Örnek olarak alınan parça, altı düzlem yüzeye ve yüzeylerden biri üzerine yerleştirilen bir deliğe ve iki kademe özelliğine sahiptir. Bir yüzey; kenarlar, köşeler (noktalar) ve bir yüzey normali ile ST’de temsil edilir. Şekil 1 de verilen parça için, her köşe Şekil 2’de gösterildiği gibi bir köşe noktasına karşılık gelir.

### 2.1. Temel Varlıkları Tanımlama

#### 2.1.1. Nokta Tanımlama

STEP’te varlık tanımlaması "#" işareti ile başlayıp bunu varlığın özel numarası, ismi (varlık belirleyici) ve ilgili parametreleri takip eder. Nokta varlığı, üç boyutlu uzayda ise, üç koordinat değerine sahiptir ve EXPRESS dilinde aşağıdaki gibi temsil edilir.

Bu açıklamalara göre Şekil 1’de verilen parçaya kademe özellikleri yerleşmeden önce, tüm yüzeylerinin köşelerine karşılık gelen noktalar Şekil 3’te gösterilmiştir. Böylece Y1 yüzeyinin noktaları, fiziksel dosyaya, aşağıdaki şekilde yazılır. Burada noktaların sıralaması, daha ileride açıklanan yüzeylerin kenar çevrimi dikkate alınarak yapılmıştır.



Şekil 1. STEP formatı hazırlanan parça

Şekil 2. Bir yüzeyin STde topolojik temsili

```
(*  
ENTITY point  
  SUPERTYPE OF (ONEOF  
    (cartesian_point,  
    point_on_curve,  
    point_on_surface))  
  SUBTYPE OF (geometry);  
END_ENTITY;  
(*
```

```
(*  
ENTITY cartesian_point;  
  SUBTYPE OF (point);  
  x coordinate: lenght_measure;  
  y coordinate: lenght_measure;  
  z coordinate: OPTIONAL lenght_measure;  
END_ENTITY;  
(*
```

(*x\_coordinate* yerleştirilen noktanın *X*, *y\_coordinate* *Y*, *z\_coordinate* *Z* koordinatıdır.)

Şekil 1’de verilen parçayı oluşturan tüm varlık tanımlamalarına ait STEP formattaki fiziksel dosya yapısı çalışmanın sonunda verilmiştir.

```
#5=POINT(0,0,0);#10=POINT(120,0,0); #15=POINT(120,0,60);#20 =POINT(0,0,60);
```

### 2.1.2 Köşe Tanımlama

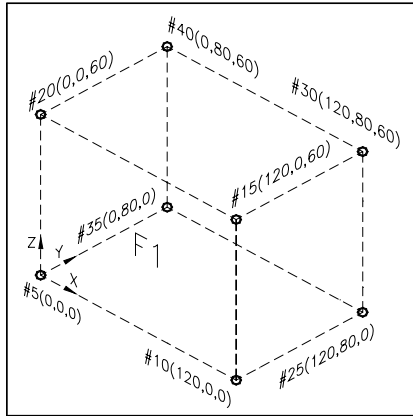
Noktalar belirlendikten sonra köşeler tanımlanabilir (Şekil 4). Noktalar önce belirlendiği için, köşeler, noktaların özel varlık numaralarını, referans olarak kolayca tanımlanır. EXPRESS’te bir köşe şu şekilde tanımlanır:

```
(*  
ENTITY vertex  
  SUBTYPE OF (topology);  
  vertex_point: OPTIONAL point;  
END_ENTITY;  
(*
```

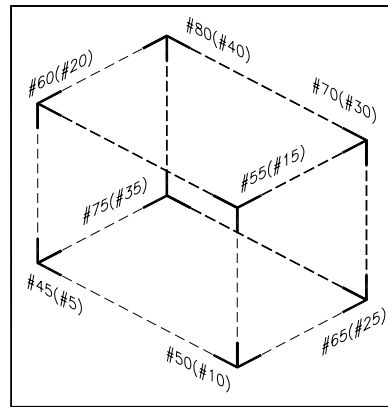
(*vertex\_point* köşe ile ilgili geometrik bir noktadır.)

### 2.1.3. Kenar Tanımlama

Bir kenarı tanımlamadan önce, kenarı yerleştirmek için, koordinat uzayında ilgili geometrik eğri tanımlanmalıdır. Burada çalışılan parçanın kenarlarına ait eğriler, birer doğrudur. Bir doğru, bir nokta ve yönü ile tanımlanan sınırlandırılmamış bir eğridir. Doğrunun pozitif yönü, yön vektörünün yönündedir. Her doğru bir yön vektörüne sahip olduğundan, doğrudan önce bir yön vektörü tanımlanmalıdır. Yön vektörü EXPRESS'te aşağıdaki şekilde tanımlanır.



Şekil 3. Nokta tanımlama



Şekil 4. Köşe tanımlama

```
*)  
ENTITY vector  
  SUPERTYPE OF (ONE OF (direction,  
    vector_with_magnitude));  
  SUBTYPE OF (geometry);  
END_ENTITY;  
(*
```

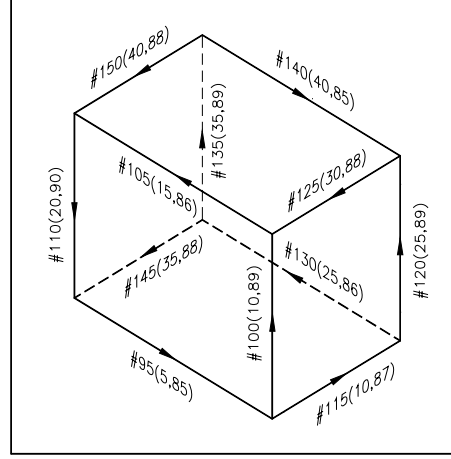
```
*)  
ENTITY direction  
  SUBTYPE OF (vector);  
  X: REAL;  
  Y: REAL;  
  Z: OPTIONAL REAL;  
END_ENTITY;  
(*
```

X,Y ve Z sırasıyla X,Y ve Z eksenlerinin yönlerinin elemanlarıdır. Örnek parçaya ait yön vektörleri aşağıda verilmiştir.

#85(1,0,0), #86(-1,0,0), #87(0,1,0), #88(0,-1,0), #89(0,0,1), #90(0,0,-1)

Yön vektörü belirlenen bir doğru, EXPRESS'te aşağıdaki şekilde temsil edilir. Örnek parçaya ait doğrular Şekil 5'te gösterilmiştir.

```
* )  
ENTITY curve  
  SUPERTYPE OF (ONE OF (line, conic,  
    bounded_curve, curve_on_surface,  
    offset_curve));  
  SUBTYPE OF (geometry);  
END_ENTITY;  
(*  
  
* )  
ENTITY line  
  SUBTYPE OF (curve);  
  pnt: cartesian_point  
  dir: direction;  
DERIVE  
  dim: INTEGER : =  
  coordinate_space(pnt);  
WHERE  
  WR1: coordinate_space(pnt) =  
  coordinate_space(dir);  
  WR2: (NOT EXIST  
  (pnt.local_coordinate_system))  
  AND(NOT EXIST  
  (dir.local_coordinate_system)) OR  
  pnt.coordinate_system =  
  dir.local_coordinate_system  
END_ENTITY;  
(*
```



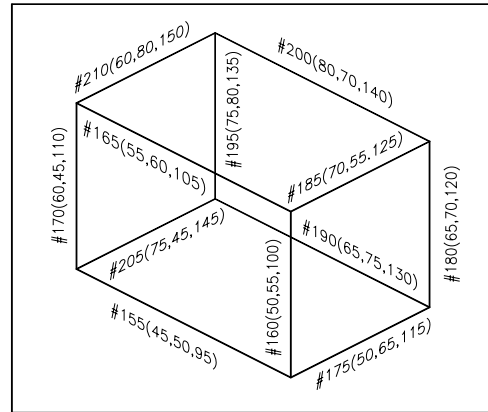
Şekil 5. Doğru tanımlama

(*pnt* doğrunun konumu, *dir* yönü, *dim* koordinat uzayının boyutsallığıdır. *WR1* Pnt ve dir, her ikisi de 2-D veya 3-D varlıklardır. *WR2* Pnt ve dir, aynı koordinat sistemini kullanarak tanımlanır.)

Buna göre kenarlar, ilgili köşeleri ve doğruları referans olarak tanımlanır. Örnek parçaya ait kenarlar Şekil 6'da gösterilmiştir. EXPRESS dilinde kenar tanımı şu şekildedir:

```
* )  
ENTITY edge  
  SUBTYPE OF (topology);  
  dge_start : vertex;  
  dge_end : vertex;  
  dge_curve : OPTIONAL curve_  
    or_LOGICAL;  
END_ENTITY;  
(*
```

*edge\_start* kenarın başlama noktası (köşe),  
*edge\_end* son noktası (köşe),  
*edge\_curve* isteğe bağlı eğri veya mantıksal  
varlıktır. Bu bir *eğri* veya  
*eğri\_mantıksal\_yapısı* olabilir.



Şekil 6. Kenar tanımlama

### 2.1.4. Kenar Çevrimi Tanımlama

*Komşu kenarlar* veya *kenar\_mantık\_yapıları* (edge\_logical\_structures) listesiyle temsil edilir ve sınırsız bir yüzeyi sınırlandırmak için kullanılır. Bir *kenar* listede verilirse, bu durum kenarın pozitif yönü ile kenar çevriminin pozitif yönünün aynı yönde olduğunu gösterir. Eğer kenarın pozitif yönü, çevrimin pozitif yönüne karşı ise, o zaman bu FALSE mantık sembolü içeren *kenar\_mantık\_yapısında* ifade edilir. Şekil 5'te iki komşu yüzey tarafından paylaşılan ve parçayı oluşturan sekiz adet doğru, yönleriyle beraber tanımlanmıştır. Buna göre bir yüzeyin kenar çevrimi, ortak kenarların yönüyle ters olmadığı sürece, *kenar\_mantık\_yapısı* kullanılmaz. Örnek parçada Y1 yüzeyine ait kenarların yönleri, kenar çevrim yönü ile aynı olduğundan, burada *kenar\_mantık\_yapısı* kullanılmamıştır. Bu açıklamalara göre, tüm yüzeylere ait kenar\_mantık\_yapıları, Şekil 7'de; kenar çevrimleri de, Şekil 8'de verilmiştir. EXPRESS'te kenar çevrimi aşağıdaki şekilde yapılır.

```
*)  
ENTITY edge_loop  
  SUBTYPE OF (loop); loop_edges: List [1:#] OF edge_or_logical;  
END_ENTITY;  
(*
```

*loop\_edges* kenar veya mantıksal listedir. *loop\_edges* kenar veya *kenar\_mantık\_yapısı* listesi olarak yerine getirilir. Şekil 8'de görüldüğü gibi her yüzeyin kenar çevrimi saat ibresinin tersine hareket ederek yüzeyin kenar çevrimini kapatmaktadır.

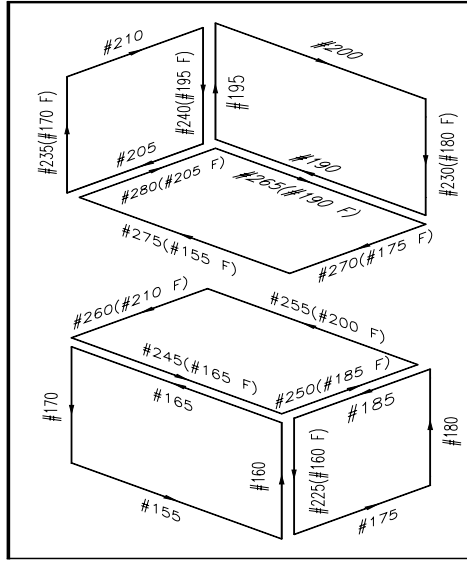
### 2.1.5. Sınırsız Düzlem Yüzey ve Yüzey Tanımlama

Geometrik bir yüzey, birden fazla yüzey tarafından kullanılabilir veya başka bir yüzey ile birleşik olabilir. *Sınırsız düzlem yüzey*, sınırsız yüzey üzerindeki bir nokta ve kendi normali ile tanımlanır. Bir düzlem yüzey, bir *nokta* ve bir *eksen* (axis2\_placement) varlığı ile temsil edilir.

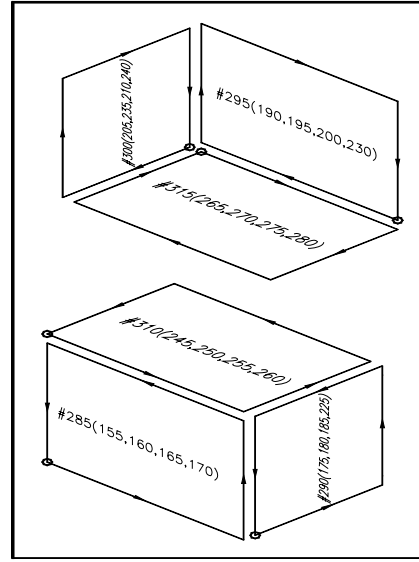
<pre>*) ENTITY plane   SUBTYPE OF (elementary_surface);   position: axis2_placement; END_ENTITY; (* <i>position</i> yüzeyin uyumu ve konumudur.</pre>	<pre>*) ENTITY axis2_placement   SUBTYPE OF (axis_placement);   location : cartesian_point;   axis : OPTIONAL direction;   ref_direction:OPTIONAL direction; END_ENTITY; (*</pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

(*location* ilgili lokal koordinat sisteminin orijini ve referans noktası, *axis* lokal Z ekseninin yönü, *ref\_direction* lokal X ekseninin yönüdür.)

Burada Y1 yüzeyi -Y yönündedir. Şayet Y1 veya öteki yüzeylerden herhangi birisine bir özellik yerleşirse, *location*, *axis* ve *ref\_direction* parametreleri de kullanılır. Örnek parçanın yüzeylerine ait düzlemler Şekil 9'da verilmiştir.



Şekil 7. Kenar mantık yapısı



Şekil 8. Kenar çevrimi

Bir düzlem yüzey, kendi sınırlı çevrimi ile temsil edilir. Yüzeyin kenar çevrimi, yüzeyin çevresinde saat ibresinin tersinedir. Her yüzey dış sınır, sınırlar, sınırlı ve sınırsız yüzeyler takımı ile tanımlanır. EXPRESS'te yüzey varlığı aşağıdaki şekilde tanımlanır:

## 2.2. Şekil Özelliği Tanımlama

Dört niteliğe sahip bir şekil özelliği, EXPRESS dilinde aşağıdaki şekilde tanımlanır.

```
*)  
ENTITY form_feature  
  feature_type : STRING;  
  implicit_reps : SET [0:#] OF implicit_form_feature;  
  pattern_rep : OPTIONAL implicit_form_feature_pattern;  
  replicate_rep : OPTIONAL replicate_form feature;  
END_ENTITY;  
(*
```

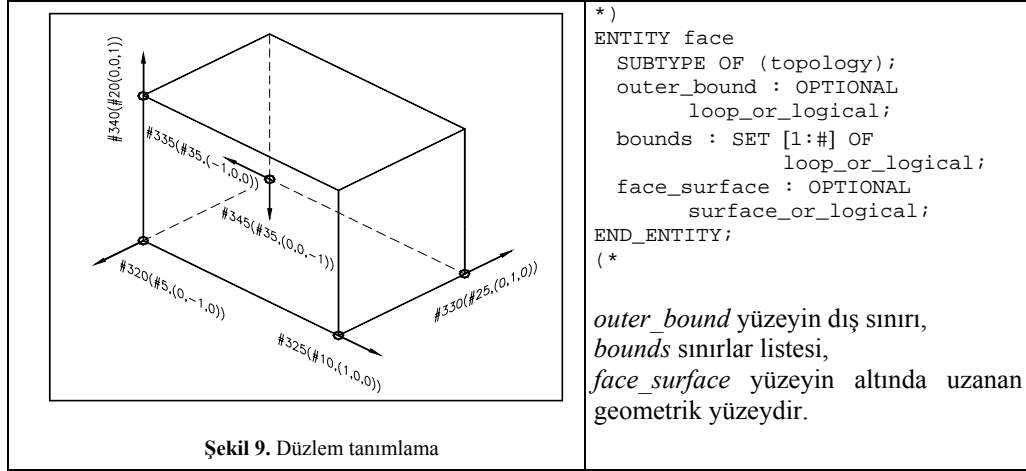
(feature type özelliğin (havşa, delik, kanal, kademe, cep vs) tipinin spesifikasyonu, implicit\_reps özelliğinin kapalı temsilini, pattern\_rep kapalı şekil özellik model temsilini, replicate\_rep özelliğinin replicate temsilidir.)

Delikler, kademeler, kanallar ve oyuklar gibi çeşitli özellikler STEP'te özellik hacmi temsili kullanılarak tanımlanır. Özellik hacmi, aşağıda gösterildiği gibi, özellik süpürme ve özellik yönlendirmeden oluşur.

```
*)  
ENTITY feature_volume  
  SUPERTYPE OF (feature_sweepXOR feature_ruling);  
END_ENTITY;  
(*
```



*feature\_ruling* kapalı bir pasaj, çıkıntı veya girintiyi belirlemek için kullanılır, *feature\_sweep*, bir düzlem profil ile bu profilin süpüreceği uzunlaşmasına bir yol ve bir veya iki uçlu şekil tanımlama içerirse, özellik süpürme prosedürel bir tanımdır. EXPRESS'te özellik süpürme aşağıdaki gibi ifade edilir.



Şekil 9. Düzlem tanımlama

```
*)
ENTITY feature_sweep
  SUPERTYPE OF (along_feature_sweep XOR axisymmetric_feature_sweep XOR
    in_out_feature_sweep)
  SUBTYPE OF (feature_volume);
  location : axis2_placement;
END_ENTITY;
(*
  (location özellik süpürme yerini belirlemek için eksen yerleştirmedir.)
```

### 2.2.1. Delik Özelliği Tanımlama

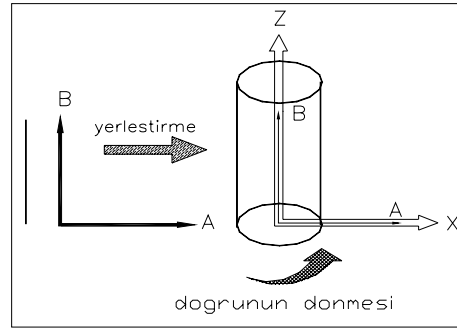
Delik özelliği *simetri\_eksenli\_ozellik\_süpürmede* tanımlanır. Simetri eksenli özellik süpürme, düzleme dik eksen etrafında, düzlem eğriyi 360° döndürmeyle meydana getirilir. Süpürme eğrisi, kapalı veya açık tanımlanabilir. Simetri eksenli özellik süpürme aşağıdaki gibi tanımlanır.

```
*)
ENTITY axisymmetric_feature_sweep
  SUPERTYPE OF (constant_diameter_axisymmetric_feature_sweep XOR
    tapered_axisymmetric_feature_sweep XOR
    other_axisymmetric_feature_sweep)
  SUBTYPE OF (feature_sweep);
  sweep_length : size_parameter;
  sweep_end : OPTIONAL_axisymmetric_feature_sweep_end;
END_ENTITY;
(*
  (sweep_length süpürülen profilin uzunluk ölçüsü, sweep_end istenen uç şeklinin tanımlanmasıdır.)
```

### 2.2.1.1. Sabit Çaplı Simetri Eksenli Özellik Süpürme

Bu profil, B eksenine paralel bir doğrudan oluşan, simetri eksenli özellik süpürme profilidir. Lokal B eksenini Z eksenine çakıştıktan sonra, doğru, bir silindiri oluşturmak üzere Z eksenini etrafında döndürülür (Şekil 10). Sabit çaplı simetri eksenli özellik süpürme, EXPRESS'te aşağıdaki gibi tanımlanır.

```
*)  
ENTITY(constant_diameter_  
  axisymmetric_feature_sweep  
  SUPERTYPE OF  
(axisymmetric_  
  feature_sweep);  
  sweep_size :  
  size_parameter;  
END_ENTITY;  
(*  
  sweep_size süpürülen özelliğin çapıdır.
```



Şekil 10. Doğrunun dönmesiyle silindir elde edilmesi

Simetri eksenli özellik süpürme ile tanımlanan kapalı şekil özelliğinin uç şekli, *simetri eksenli özellik süpürme* ucu ile belirlenir. Bu, düz, konik ve küresel uçlu bir delik için ortak uç şekillerini modellemek üzere kullanılabilir. Uç şekli belirtilmezse özelliğin düz uçlu olduğu kabul edilir. Uç şekli aşağıdaki şekilde belirlenir.

```
*)  
ENTITY axisymmetric_feature_sweep_end  
  SUPERTYPE OF (axisymmetric_feature_sweep_flat_end XOR);  
  axisymmetric_feature_sweep_spherical_end XOR  
  axisymmetric_feature_sweep_conical_end);  
  blend : OPTIONAL implicit_edge_blend;  
END_ENTITY;  
(*
```

### 2.2.1.2. Simetri Eksenli Düz Uçlu Özellik Süpürme

Simetri eksenli özellik süpürmenin uç şekli düz olup, bu, aşağıdaki şekilde temsil edilir. Simetri eksenli düz uçlu özellik süpürme düzleminde lokal,  $Z=0$ 'dır.

```
*)  
ENTITY axisymmetric_feature_sweep_flat_end  
  SUBTYPE OF (axisymmetric_feature_sweep_end);  
END_ENTITY;  
(*
```

### 2.2.2. Kademe Özelliği Tanımlama

Boyan Boya Özellik Süpürme, özellik süpürmenin bir alt tipi olup ilk şeklin sınırları içerisinde bulunan (hava/malzeme ara yüzünde) bir yoldaki özellik süpürmedir. Kademe,

kanal gibi bazı ortak özellikler bu şekilde tanımlanır. Boydan boya özellik süpürme varlığının EXPRESS'te tanımlanması aşağıdaki gibidir.

```
*)  
ENTITY along_feature_sweep  
  SUBTYPE OF (feature_sweep);  
  sweep_path : feature_sweep_path;  
  sweep_profile : open_feature_sweep_profile;  
  sweep_ends : SET [0:2] OF along_feature_sweep_end;  
END_ENTITY;  
(*
```

(*sweep\_path* süpürmeyi belirleyen yoldur. *sweep\_profile* yol boyunca süpürülecek profilin özelliğidir. *sweep\_ends* süpürülen şeklin uç tipini açıklar.)

### 2.2.2.1. Boydan Boya Özellik Süpürme Ucu

Kapalı şekil özelliğinin uç şekli, boydan boya özellik süpürme ucu olarak adlandırılan varlıkla tanımlanmıştır. Bu, sadece süpürme ucu olduğu zaman kullanılabilir. Eğer özellik süpürme işlemi parçanın bir ucundan girip öteki ucundan çıkıyorsa, herhangi bir özellik süpürme ucu söz konusu değildir. Bir örnek olarak bu varlık, frezelenmiş kanalın yuvarlatılmış ucunu modellemek için kullanılabilir. Boydan boya özellik süpürme ucu aşağıdaki şekilde tanımlanır.

```
*)  
ENTITY along_feature_sweep_end  
  SUPERTYPE OF (along_feature_sweep_flat_end XOR;  
                along_feature_sweep_radiused_end);  
  sweep_end : feature_end_types;  
END_ENTITY;  
(* (sweep_end başlangıç veya sonda bir uç olup olmadığını gösterir. )
```

### 2.2.2.2. Boydan Boya Özellik Süpürme (düz uçlu)

Boydan boya özellik süpürme ucunun düzlem olduğunu göstermek için kullanılır.

```
*)  
ENTITY along_feature_sweep_flat_end  
  SUBTYPE OF (along_feature_sweep_end);  
  end_blend: OPTIONAL implicit_edge_blend (end_blend istenen uç birleşme yerinin tanımıdır.)  
END_ENTITY;  
(*
```

### 2.2.2.3. Özellik Süpürme Yolu

Açık form özelliğinin şeklini tanımlamak için boydan boya süpürülen bir özellik süpürme profili eğrisidir.

```
*)  
ENTITY feature_sweep_path  
  SUPERTYPE OF (circular_feature_sweep_path XOR  
                spiral_feature_sweep_path XOR  
                surface_conforming_feature_sweep_path_ XOR  
                other_feature_sweep_path_ XOR  
                linear_feature_sweep_path);  
END_ENTITY; (*
```

#### 2.2.2.4. Doğrusal Özellik Süpürme Yolu

Doğrusal bir çizgi olarak tanımlanır. Özelliğin LKS'nin +Z eksenini boyunca, Z=0'dan Z=L'ye kadar uzanır. Burada L= yol uzunluğudur. Doğrusal süpürmede kullanılan profillerin A ve B eksenleri sırasıyla özelliğin LKS'nin X ve Y eksenlerine yerleştirilir.

```
*)  
ENTITY linear_feature_sweep_path  
  SUBTYPE OF (feature_sweep_path);  
  path_lenght : size_parameter;          (path_lenght süpürme yolunun uzunluk boyutudur.)  
END_ENTITY;  
(*
```

#### 2.2.2.5. Özellik Süpürme Profili

Bir şekil özelliğini tanımlamak için, bir özellik süpürme yolunun, alan boyunca süpürülmesiyle oluşan bir düzlem eğri veya birleşik eğriler setidir. Profiller, yolun tipine göre özelliğin LKS'ne yerleştirilen lokal AB uzayında tanımlanır. AB uzayında profilin uyumu, profilin tipine göre belirlenir.

```
*)  
ENTITY feature_sweep_profile  
  SUPERTYPE OF (closed_feature_sweep XOR  
                open_feature_sweep_profile);  
END_ENTITY;  
(*
```

#### 2.2.2.6. Açık Özellik Süpürme Profili

Kapalı olmayan eğri veya bileşik eğriden meydana gelen özellik süpürme profilidir. Profilin açık uçlarının malzeme/hava arayüzüne kadar uzatıldığı kabul edilir. Örneğin "U" şeklindeki bir profil için, sadece genişlik ve kenar birleşme yerleri tanımlanır. Düşey kolların uzunluğu profilin yerleştirilmesiyle belirlenir.

```
*)  
ENTITY open_feature_sweep_profile  
  SUPERTYPE OF (circular_arc_feature_sweep_profile XOR  
                rounded_u_feature_sweep_profile XOR  
                vee_feature_sweep_profile XOR  
                square_u_feature_sweep_profile XOR  
                tee_feature_sweep_profile XOR  
                ell_feature_sweep_profile XOR  
                half_obround_feature_sweep_profile XOR  
                other_open_feature_sweep_profile);  
SUBTYPE OF (feature_sweep_profile);  
END_ENTITY;  
(*
```

#### 2.2.2.7. L (Kademe) Özellik Süpürme Profili

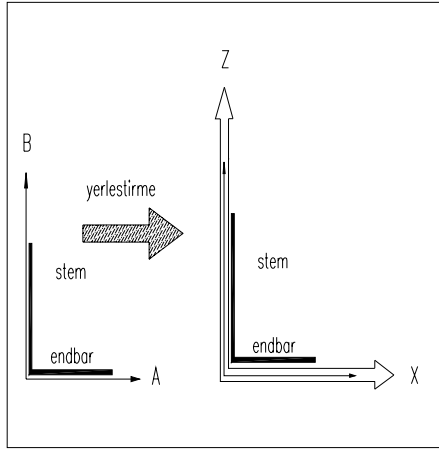
L şeklindeki özellik süpürme profilidir. Bu bir dikey *endbar* ve *stem*'i içerir. Stem, endbarla uçları tespit edilmiş iki paralel yarı sonsuz doğru iken, Endbar bir dikdörtgendir. Endbar, stem'e zıt kenarlı pozitif B yarım düzleminde ve A eksenini üzerindedir. (Şekil 11).

Endbar'ın yüksekliği ve stem'in genişliği sıfıra eşit olduğunda, L özelliğini yerleştirme Şekil 12'de görüldüğü gibi yapılır. L özellik süpürme profili EXPRESS'te şu şekilde yazılır.

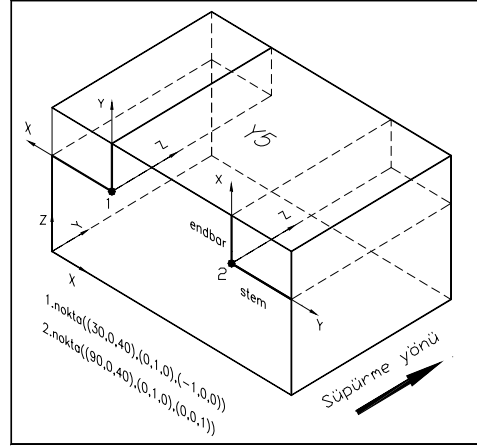
```
*)  
ENTITY ell_feature_sweep_profile  
  SUPERTYPE OF open_feature_sweep_profile);  
  stem_width      : size_parameter;  
  endbar_width    : size_parameter;  
  endbar_height   : size_parameter;  
  ell_orientation : ell_orientation_types;  
  stem_endbar_blend : OPTIONAL implicit_edge_blend;  
  endbar_blend1   : OPTIONAL implicit_edge_blend;  
  endbar_blend2   : OPTIONAL implicit_edge_blend;  
  endbar_blend3   : OPTIONAL implicit_edge_blend;  
END_ENTITY;  
(*
```

*stem\_width* stem'in genişliği (X boyutu), *endbar\_width* endbarın genişliği (X boyutu), *endbar\_height* endbarın yüksekliği (Y boyutu), *ell\_orientation* endbarın ucunun pozitif veya negatif X yarım düzleminde olup olmadığını gösterir. *stem\_endbar\_blend* stem ve endbar arasında kalan konkav köşenin kavisidir. *endbar\_blend1*, *endbar\_blend2*, *endbar\_blend3* L'nin köşelerinin birleşim yeridir. *L-uyum tipleri* aşağıdaki şekilde tanımlanır:

```
*)  
TYPE ell_orientation_types = ENUMERATION OF  
  (ell_positive;  
  ell_negative);  
END_ENTITY;  
(*
```



Şekil 11. Kademe özelliğinin A-B ekseninden X-Y eksenine yerleştirilmesi



Şekil 12. Örnek parçada L kademe özelliği süpürme

Kademe özelliği için varlık bilgisi, fiziksel dosyaya yazılmadan önce, özellik Şekil 12'de görüldüğü gibi, LKS'ye yerleştirilir. Tolerans, boyutlar ve bir parçayı özellik tabanlı

tasarım esaslarına göre tanımlayan diğer varlık tanımlamaları da EXPRESS dili kullanılarak benzer şekilde yerine getirilir.

### 2.3. Fiziksel Dosyada Komple Bir Parçanın Temsil Edilmesi

Bu bölümde açıklanan varlık tanımlamalarını kullanarak Şekil 1'de verilen örnek parçanın, fiziksel dosyada STEP standarda göre temsili aşağıda verilmiştir.

```
/* STEP standard */
STEP;
HEADER
FILE_IDENTIFICATION('PART2','15.05.1997',
'Yaldiz-Celik','Sulcu University','02,Programme name');
FILE_DESCRIPTION('Part2-two steps and a hole');
IMP_LEVEL('LEVEL2');
CLASSIFICATION();
MAXSIG();
ENDSEC;
DATA;
#5 = POINT(0,0,0);
#10 = POINT(120,0,0);
#15 = POINT(120,0,60);
#20 = POINT(0,0,60);
#25 = POINT(120,80,0);
#30 = POINT(120,80,60);
#35 = POINT(0,80,0);
#40 = POINT(0,80,60);
#45 = VERTEX(#5);
#50 = VERTEX(#10);
#55 = VERTEX(#15);
#60 = VERTEX(#20);
#65 = VERTEX(#25);
#70 = VERTEX(#30);
#75 = VERTEX(#35);
#80 = VERTEX(#40);
#85 = DIRECTION(1,0,0);
#86 = DIRECTION(-1,0,0);
#87 = DIRECTION(0,1,0);
#88 = DIRECTION(0,-1,0);
#89 = DIRECTION(0,0,1);
#90 = DIRECTION(0,0,-1);
#95 = LINE(#5,#85);
#100 = LINE(#10,#89);
#105 = LINE(#15,#86);
#110 = LINE(#20,#90);
#115 = LINE(#10,#87);
#120 = LINE(#25,#89);
#125 = LINE(#30,#88);
#130 = LINE(#25,#86);
#135 = LINE(#35,#89);
#140 = LINE(#40,#85);
#145 = LINE(#35,#88);
#150 = LINE(#40,#88);
#155 = EDGE(#45,#50,#95);
#160 = EDGE(#50,#55,#100);
#165 = EDGE(#55,#60,#105);
#170 = EDGE(#60,#45,#110);
#175 = EDGE(#50,#65,#115);
#180 = EDGE(#65,#70,#120);
#185 = EDGE(#70,#55,#125);
#190 = EDGE(#65,#75,#130);
#195 = EDGE(#75,#80,#135);
#200 = EDGE(#80,#70,#140);
#205 = EDGE(#75,#45,#145);
/* nokta tanımlama */
/* köşe tanımlama */
/* vektör tanımlama */
/* doğru tanımlama */
/* kenar tanımlama */
```

```
#210 = EDGE(#60,#80,#150);
#225 = EDGE_LOGICAL_STRUCTURE(#160,.F.);
#230 = EDGE_LOGICAL_STRUCTURE(#180,.F.);
#235 = EDGE_LOGICAL_STRUCTURE(#170,.F.);
#240 = EDGE_LOGICAL_STRUCTURE(#195,.F.);
#245 = EDGE_LOGICAL_STRUCTURE(#165,.F.);
#250 = EDGE_LOGICAL_STRUCTURE(#185,.F.);
#255 = EDGE_LOGICAL_STRUCTURE(#200,.F.);
#260 = EDGE_LOGICAL_STRUCTURE(#210,.F.);
#265 = EDGE_LOGICAL_STRUCTURE(#190,.F.);
*/
#270 = EDGE_LOGICAL_STRUCTURE(#175,.F.);
#275 = EDGE_LOGICAL_STRUCTURE(#155,.F.);
#280 = EDGE_LOGICAL_STRUCTURE(#205,.F.);
#285 = EDGE_LOOP(#155,#160,#165,#170);
#290 = EDGE_LOOP(#175,#180,#185,#225);
#295 = EDGE_LOOP(#190,#195,#200,#230);
#300 = EDGE_LOOP(#205,#235,#210,#240);
#310 = EDGE_LOOP(#245,#250,#255,#260);
#315 = EDGE_LOOP(#265,#270,#275,#280);
#320 = PLANE(AXIS2_PLACEMENT(#5,(0,-1,0),));
#325 = PLANE(AXIS2_PLACEMENT(#10,(1,0,0),));
#330 = PLANE(AXIS2_PLACEMENT(#25,(0,1,0),));
#335 = PLANE(AXIS2_PLACEMENT(#35,(-1,0,0),));
#340 = PLANE(AXIS2_PLACEMENT(#20,(0,0,1),));
#345 = PLANE(AXIS2_PLACEMENT(#35,(0,0,-1),));
#350 = FACE(#285,(#285),#320);
#355 = FACE(#290,(#290),#325);
#360 = FACE(#295,(#295),#330);
#365 = FACE(#300,(#300),#335);
#370=AXIS2_PLACEMENT((60,40,0),(0,0,1),(1,0,0)); /
#375= FEATURE_SWEEP
SCOPE
#382= AXISYMMETRIC_FEATURE_SWEEP
SCOPE
#384=CONSTANT_DIAMETER_AXISYMMETRIC_FEATURE_SWEEP(20);
ENDSCOPE (60.); /* 60 süpürme uzunluğu.
ENDSCOPE (#370);
#385=AXIS2_PLACEMENT((30,0,40),(0,1,0),(-1,0,0));
#390= LINEER_FEATURE_SWEEP_PATH(80);
#395=L_FEATURE_SWEEP_PROFILE(0,30,0, L_POZITIVE., , , );
#400= FEATURE_SWEEP
SCOPE
#405= ALONG_FEATURE_SWEEP(#390,#395,());
ENDSCOPE (#385);
#410=AXIS2_PLACEMENT((90,0,40),(0,1,0),(0,0,1));
#415= LINEER_FEATURE_SWEEP_PATH(80);
#420=L_FEATURE_SWEEP_PROFILE(0,20,0, L_POZITIVE., , , );
#425= FEATURE_SWEEP SCOPE
#430=ALONG_FEATURE_SWEEP(#415,#420,()); ENDSCOPE (#410);
#435= FACE
SCOPE
#440= FORM_FEATURE('HOLE',(#375), , );
#445= FORM_FEATURE('STEP',(#400), , );
#450= FORM_FEATURE('STEP',(#425), , );
ENDSCOPE (#310,(#310),#340);
#455 = FACE(#315,(#315),#345);
#460 = TOLERANS_RANGE(0.01,0.01);
#462 = TOLERANS_RANGE(0.02,0.02);
#464 = TOLERANS_RANGE(0.05,0.05);
#466 = TOLERANS_RANGE(0.2,0.2);
#468 = TOLERANS_RANGE(0.1,0.1);
#470 = TOLERANS_RANGE(0.09,0.09);
#472 = TOLERANS_RANGE(0.1,0.1);
#474 = TOLERANS_RANGE(0.6,0.6);
#480 = LOCATION_DIMENSION(#472,#355,#365,.T.,.T., );
#485 = LOCATION_DIMENSION(#468,#360,#350,.T.,.T., );
```

/\* Y6'nın kenarları Şekil 5'te tanımlanan yönlere terstir

/\* kenarları referans alan kenar çevrimi \*/

/\* düzlem yüzey tanımlama\*/

/\* üç nitelikli yüzey tanımlama -dış sınır, sınır listesi,düzlem yüzey- \*/

\* deliği yerleştirmek için Z,X yönleri \*/

Simetri eksenli özellik ucu tanımsız \*/

/\* L-kademe için şekil tanımlama \*/

/\*özellik süpürme ucu tanımsız\*/

/\* üzerine yerleşen özellikleriyle beraber Y5 yüzeyi \*/

/\* Şekil özelliği tanımlama \*/

```
#490 = LOCATION_DIMENSION(#468,#455,#435.,T.,T.);
#495 = LOCATION_DIMENSION(#466,#440,#365.,T.,T.);
#500 = LOCATION_DIMENSION(#466,#440,#350.,T.,T.);
#505 = LOCATION_DIMENSION(#464,#450,#455.,T.,T.);
#510 = LOCATION_DIMENSION(#464,#450,#365.,T.,T.);
#515 = LOCATION_DIMENSION(#464,#445,#365.,T.,T.);
#520 = LOCATION_DIMENSION(#466,#445,#455.,T.,T.);
#525 = SIZE_DIMENSION(#440,#468.,TAM.);
#530 = CIRCULARITY(#440),0.05);
#535 = STRAIGHTNESS(#440),0.6,.,REGARDLESS.,T.,60);
#540 = CLOSED_SHELL(#350,#355.,#360,#365,#435,#455);
#545 = MANIFOLD_SOLID_PREP(#540,());
#550 = PART_MODEL(#545,'mm',(#440,#445,#450),(),());
#555 = MATERIAL_PROPERTY('1040',.);
#560 = AXIS2_PLACEMENT((0,0,0),(0,0,1),(1,0,0));
#565 = PART_MODEL_STRUCTURE(#550,#560);
ENDSEC;
ENDSTEP;
```

### 3. SONUÇ

BDT/BDÜ sistemlerinin bütünleşmesinde esas olarak tespit edilmiş çözülmesi gereken iki konu vardır. Bunlardan birincisi BDT ve BDÜ için parçanın tam olarak tanımlanması ikincisi ise BDT ve BDÜ sistemleri arasında parça bilgisinin ve üretimle ilgili bilgilerin transferidir. STEP standardına göre yapılan parça bilgisini tanımlama ve dönüşümü BDT/BDÜ bütünleşmesi çalışmalarında umut verici bir gelişmedir. Parçanın hem geometrik hem de teknolojik bilgileri STEP standartlarında mevcuttur. Bu formatta oluşturulan parça tasarım bilgisi farklı BDT sistemleri arasında ve BDÜ sistemleri arasında tam bilgi temsili ve parça bilgisi transferi amacıyla kullanılabilir. STEP, ISO tarafından geliştirilmiş, parça temsili ve parça bilgisinin transferini sağlama amacıyla geliştirilmiş bir dünya standardıdır.



## KAYNAKLAR

- [1] A. Chep, L. Tricarico, 1999; Object-Oriented Analysis and Design of a Manufacturing Feature Representation, *International Journal of Production Research*, Vol.37, No.10, pp.2349-2379.
- [2] M.E. Ssemakula, J.S. Gill, 1989; CAD/CAPP Integration Using IGES, *AME-Advanced Manufacturing Engineering*, Vol 1, No 5, pp. 264-270.
- [3] P. Smith, 1990; CAD/CAM Data Exchange, *CIM Review The Journal of Computer Integrated Manufacturing Management*, Vol 6, No 2, pp. 31-33, Winter.
- [4] STEP-Standard for the Exchange of Product Model Data, 1998; *The First Working Draft of STEP Version 1.0*, ISO TC 184/SC4/ WG 1, pp. 1-546.
- [5] S. Joshi, and T. Chang, 1990; Feature Extraction and Feature Based Design Approaches in the Development of Design Interface for Process Planning", *Journal of Intelligent Manufacturing, London*, Vol. 1, No. 1, pp 1-15.
- [6] C.S. Chen, 1998; Developing a Feature Based Knowledge System for CAD/CAM Integration, *Computers Ind. Eng.*, Vol 15, No.1-4, pp.34- 40.
- [7] D.H. Lee, D. Kiritsis, P.Xirouchakis, 2001; Branch and Fathoming Algorithms for Operation Sequencing in Process Planning, *International Journal of Production Research*, Vol.39, No.8, pp.1649-1669.
- [8] D. An, H.R. Leep, H.R. Parsaei, and A.P. Nyaluke, 1995; A Product Data Exchange Integration Structure Using PDES/STEP for Automated Manufacturing Applications, *Computer Industrial Engineering*, Vol. 29, No.1-4, pp.711-715, Elsevier.
- [9] P. Mangesh, R.N. Bhandarkar, 2000; STEP-Based Feature Extraction from STEP Geometry for Agile Manufacturing, *Computers In Industry*, Vol.41, pp.3-24, Elsevier.
- [10] X.W. Xu, H. Wang, J. Mao, S.T. Newman, T.R. Kramer, F.M. Proctor, and J.L. Michaloski, 2005; STEP-compliant NC research: the search for intelligent CAD/CAPP/CAM/CNC integration, *International Journal of Production Research*, Vol.43, pp.3703-3743.
- [11] ISO 10303-11 Description Methods: The EXPRESS Language Reference Manual.
- [12] X.G. Ming, K.L. Mak, J.Q. Yan, 1998; A PDES/STEP-Based Information Model for Computer-Aided Process Planning, *Robotics and Computer-Integrated Manufacturing*, Vol.14, pp.347-361.
- [13] K. Hilary, F. Nick, W. Alan, W. Nigel, 2001; A Generic Framework for Transforming EXPRESS Information Models, *Computer Aided Design*, Vol.33, pp.501-510, Elsevier.