

NESNE YÖNELİMLİ YAKLAŞIMLA ÖZELLİK TABANLI MODELLEME

İsmet ÇELİK*, Ali ÜNÜVAR **

Özellik-tabanlı modeller, Bilgisayar Destekli Tasarım (CAD) ve Bilgisayar Destekli Üretim (CAM)’ın bütünlendirilmesinde kullanılan en iyi araçlar olmuşlardır. Bu modellerde imalat işlemleri ile ilişkili olan özellikler kullanılarak tasarım gerçekleştiriliyor. CAD ve CAM’ın bütünlendirilmesi işlemi geometri, tolerans bilgisi ve imalat ile ilişkili veriyi içeren çeşitli veri tiplerinin bütünlüğe kapasitesine sahip geniş bir bilgi yapısı gerektirir. CAD ve CAM’i içine alan yoğun uygulamalarda nesne-yönelimli modelleme yaklaşımı, tasarımda ve oluşturululan verinin uygulanmasında son yıllarda önem kazanmıştır. Nesne-yönelim düşüncesi, büyük ve geniş veri ağaçları hiyerarşilerine ayrılrılarak bu sistemlerin karmaşıklığının giderilmesi ve belirginleşmesi için çok iyi bir araç sağlar. Bu çalışmada nesne-yönelimli veri tabanı yapısı kullanılarak oluşturulan özellik tabanlı bütünlük CAD/CAM sistemlerinin modellenmesi incelenmiştir. Önerilen nesne-yönelimli model, sistem tarafından istenen veri bütünlüğünü ve veri soyutlamasını destekler.

Anahtar sözcükler : Nesne-yönelimli modelleme, özellik-tabanlı modelleme, bilgisayar destekli tasarım, bilgisayar destekli imalat.

Feature-based models have been the best tools for integrating Computer Aided design (CAD) and Computer Aided Manufacturing (CAM). These models enable design using the properties related to production processes. The integration of CAD and CAM requires a wide information structure comprising data on geometry, tolerance, and manufacturing. Feature based modeling approach to intense applications involving CAD and CAM, has recently gained importance for design and application purposes. Object-oriented thought decomposed into a large and wide data tree hierarchy provides a very useful tool for simplifying and crystallizing these systems. This study investigates modeling of property based integrated CAD/CAM systems using the object oriented database structure. The proposed object-oriented model supports the data integrity and abstraction required by the system.

Keywords : Feature recognition, process planning, CAD/CAM object oriented modelling

* Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Makina Mühendisliği Ana Bilim Dalı

** Selçuk Üniversitesi Mühendislik Mimarlık Fakültesi, Makina Bölümü

GİRİŞ

Ünümüzün ileri teknoloji seviyesine sahip imalat ortamında rekabetçi olabilmek için parçanın tasarımından çıkartılan bilgiyi kullanarak bir otomatik işlem planlama sistemi oluşturmak gereklidir. Özellik modelleme bu konuda Bilgisayar Bütünlüklü Üretim (CIM) sistemleri için bir anahtar oldu [1]. Özellik tabanlı tanımlama, imalat bilgisinin ve teknolojik bilginin analizini ve tarifnamesini gerektirir [2]. CAD/CAM sistemlerinin kullanım amacı, dizayn ve imalat işlemlerinin çeşitli fonksiyonlarını birleştirmektir.

Parça tasarımcıları genelde, özellikleri, parçadan talaş kaldırma operasyonları tarafından uzaklaştırılacak hacimler olarak düşünürler. Bu özellikler tanımlanarak özellik-tabanlı modeller oluşturulurlar. Talaş kaldırma açısından bakıldığından parçalar dış şekillerine göre, dönel ve prizmatik olmak üzere iki sınıfa ayrılabilirler. Prizmatik parça sınıfı yine imalat açısından kanallar, delikler, kademeler, delikler, T-kanallar, U-kanallar, V-kanalları gibi belirgin özelliklere sahip parça sınıflarına ayrılabilirler. Bütün bu özellikler, koordinat eksenlerine dik, paralel ve belirli bir açıda yönelimleriyle benimsenirler [3].

CAD sistemleri geometri ve topolojiyi içeren nispeten düşük seviyeli bilgiyi içerirler. CAM sistemleri ise boyutlar, toleranslar, imalat işlemleri, malzeme bilgisi ve yüzey özelliklerini gibi yüksek seviyeli bilgi ile ilgilenirler. CAD ve CAM sistemleri arasındaki bu farklılığı kapatmak için, yukarıda bahsedilen "şekil özellikleri", "özelliklerin tasarımını" ya da diğer ismiyle "özellik-tabanlı tasarım ve modelleme" kavramlarını kullanma düşüncesi CAD/CAM bütünlüğü için bir çözüm sağladı. Özellik-tabanlı modelleme imalatçıya sadece geometrik ve topolojik bilgi sağlamak yerine bir parça modeli sunar. Bu model imalat için gerekli, geometrik ve topolojik bilginin yanında, CAM için gerekli diğer tüm bilgiyi de sunar.

Özellik tabanlı modelleme için, "ilişkisel" yaklaşım ve son yıllarda ortaya çıkan "nesne-yönelimli" yaklaşım modelleme teknikleri kullanılarak, özellikler için gerekli bilgi ve bu bilginin fonksiyonel yönünü veren davranışları tanımlanabilir. Bir özellik bir geometrik yüzeydir ve düzlemsel, açısal ve parametrik eşitlikler ile temsil edilebilir. Bir geometrik özelliğin temsili; özelliği oluşturan yüzeylerin yüzey listesi,

özellik sınırlarını oluşturan kenarların kenar listesi, koordinat sisteminde tanımlı köşelere ait noktaların köşe listesi olarak açıklanabilir. İmalat için gerekli tolerans bilgisi gibi diğer geometrik olmayan bilgilerde bu özellik geometrisine ilave edilebilirler[4]. Özellik-tabanlı modelleme ile böylece imalat için gerekli bütün bilgi, bir model olarak CAM'e sunulabilir.

Özellik-tabanlı modelleme kullanılarak yapılan parça modellemesinin, tam otomatik bir işlem planlama sisteminde uygulanabilmesi için belirli bir disiplin içinde yapılandırılması gereklidir. Çünkü, işlem planlama faaliyeti parça bilgisi, atölye bilgisi, tezgah bilgisi, takım bilgisi, tertibat bilgisi, yönetim bilgisi gibi çok geniş bilgiyi gerektirir. Bu bilgilerin parçalara ayrılp sınıflandırılmaları, aralarındaki ilişkilerin belirlenmesi, her parçanın kendi bilgisini bağımsız olarak içermesi, sistemin diğer parçaları ile iletişim kurabilmesi ve güncellenebilmesi gereklidir. Bu durum göz önüne alındığında özellik-tabanlı modellemenin sistematik olarak yapılandırılmadığında yetersiz kaldığı görülür. Çünkü o sadece tek tek özelliklerini ele alır ve modeller.

Özellik-tabanlı modellemenin nesne-yönelimli yaklaşım ile yapılması, son yıllarda tam otomatik işlem planlama sistemlerinin oluşturulması için ortaya atılmıştır. Nesne-yönelimli modelleme, nesneler şeklinde ayrıstırılan bir sistemde bilginin modellenmesini sağlayan bir metottur. Bu modelleme tekniğinde, bağımsız olarak oluşturulan nesneler bir kimlik, bir durum ve bir davranış belirtirler. Buna ek olarak, nesne yönelimli modelleme; nesnelerin diğer nesnelerden bilgi ve prosedürleri miras alabildiği (kalıtsallık) ve her bağımsız nesnenin kendisini aktif kılmak ve uygulamak için gerekli fonksiyonlarının ve değişkenlerinin tanımlanlığı bağımsız nesneler şeklinde bilgi ve prosedürlerinin çerçevelendiği (kapsülleme) geçtiği tarafından karakterize edilir [20].

Nesne-yönelimli modelleme sistemlerinde, sistemin parçaları, kendi "bilgilerini (datalar)" ve bu bilgilerin "davranışlarını (metotlar)" içeren, sistem varlıklarına dönüştürülürler [2]. Sistemin bu varlıklarını nesne olarak adlandırırlar. Her nesne kendine ait bilgileri içerir. Her

nesnenin kendi bilgisini diğer nesnelerden bir çerçeve ile ayırması "kapsülleme" olarak adlandırılmaktadır. Kapsülleme sayesinde sistemin her nesnesine diğer nesnelerden bağımsız olarak müdahale edilebilir. Bu durum nesnelerin güncellenmesine olanak sağlar. Sistemin parçaları olan nesneler arasındaki iletişim ve etkileşim ise "mesajlar" ile yapılır. Sistemin nesneleri alt sınıf/üst sınıf hiyerarşisi içinde kalıtsanarak, bir sınıf şeklinde oluşturulurlar.

Örneğin; bir imalat sistemi olarak atölyeyi ele alalım. Buradaki benzer özelliklere sahip tezgahlar bir nesneler sınıfını oluştururlar. Takım grupları, parça grupları ayrı bir nesneler sınıfıdır. Tezgahın adı, gücü, hassasiyeti vs, tezgahın özellikleri olarak datasını; yaptığı ilerleme, kesme hareketleri onun davranışlarını karakterize ederler. Bu şekilde bir tezgah kapsülleňerek tezgah nesnesi oluşturulur ve kendi bilgisine sahip olur. Ayrıca tezgah nesnesi diğer nesnelerle iletişim ve etkileşim için mesajlara da sahiptir. Bu konu ileriki bölümlerde daha ayrıntılı olarak açıklanacaktır.

Henüz tam otomatik bir işlem planlama sistemi oluşturulmamıştır. İmalatın belirli kademeleri için otomatik işlem planlama sistemleri vardır. Tam otomatik bir imalat ortamının oluşturulmasında CAD'den CAM'e geçiş için, özellik- tabanlı modellemenin yapılması kaçınılmazdır. İmalat ortamları birçok unsura sahiptir. Nesne-yönelimli modelleme, bu unsurların bir sistem içinde, sistemin parçaları olarak ve aralarındaki ilişkiler ile birlikte modellenmesi esasına dayanır. Nesne yönelimli özellik-tabanlı modelleme, üzerinde araştırmaların sürdürü ve tam otomatik işlem planlama sistemlerinin oluşturulması için çözüm olacak bir yaklaşım olarak düşünülmektedir.

LITERATÜRE BAKIŞ

Tasarım ve işlem planlama arasında bağlantı yapmak için iki yaklaşım ortaya atılmıştır. Birincisi Chang tarafından tanımlanan, parçanın görüşünerine göre imalat özelliklerinin tanımlanmasıdır [5]. Böylece CAD ve CAM arasında doğrudan bir hat yapılabılır. Diğer yaklaşım

CAD modelinden imalat özelliklerinin tanımlanması şeklindedir. Gu, Dong ve Parsei, öncelikli olarak tasarım özelliklerinin imalat özellikleri şekline dönüşümü üzerine yoğunlaştılar [6,7]. Dave ve Sakurai, parçayı hücreler şeklinde delta hacimlerine ayırdılar ve bu hücreleri imalat özellikleri ile birleştirdiler [8]. Regli ve arkadaşları şekil özelliklerinin imal edilebilirlikleri üzerinde geometrik bir yorumlama yaptılar [9]. İkinci yaklaşım ürün tasarımcısının ihtiyaç duyduğu en iyi tasarım araçlarını sağlar. Mühendislik tasarımcısı ürünün fiziksel ve fonksiyonel özelliklerini kavramallaştırır ve planlar.

Sheu ve arkadaşları tarafından şekil-özellikleri, şekil temsili ve özellik işleme mekanizması olarak tanımlandı. Sheu modeline göre; şekil özellikleri genelde hacimler veya kapalı hacimlerin yüzeyleri olarak tanımlanırlar. Örneğin hacim temsilinde bir cep rıjıt bir katıdır. Yüzey temsilinde ise bir cep dört ya da beş yüzey tarafından tanımlanır. Hacim temsili daha üstündür çünkü özellikler daha kolay belirlenebilir. Toleranslar Sheu modelinin çekirdek elemanlarındandır [10].

Wu, Nesne yönelimli modelleme üzerinde çalışmalar yapmış ve Hiyerarşik ve Nesne Yönelimli İmalat Sistemleri Analizi (HOOMSA) adını verdiği bir sistemi geliştirmiştir. Sistem, varlıklarını ve aralarındaki ilişkileri soyutlamaktadır. Varlıklar nesneler olarak sınıflandırılırlar ve her nesne kendi bilgisini ve diğer nesnelerle iletişim kurmasını sağlayan metotlarını içerir [3].

Aslan, AutoLISP programları dili komutları ile silindir, koni, pah, kanal, kavis, vida gibi dönel parçalar için özelliklerini tanımlar. Planlamacı ekran menüsünden işlemin özelliğine göre çizimi yapacağı komutu seçer. Seçilmiş olan komutun yardımıyla iş profili tanımlanır. Geliştirilen sistem profil özelliklerini belirleyerek iş profillerini tanır ve son işlemci tasrarlar. Sistem CAD ürün bilgisini DXF (Data Exchange File) dosya yapısı olarak kabul eder ve çalışır [19].

Yıldız prizmatik parçalar için bir işlem planlama sistemi geliştirmiştir ve geliştirdiği sistemde özellik tabanlı modelleme yaklaşımını kullanmıştır. Sistem ürün modelini STEP (STandarts for Exchange of Product Data)

standartlarına uygun olarak kabul eder ve bilgi tabanında tanımlı kanal, kademe, delik gibi özellikleri, ürün modelinden tanıyarak bunlar için yine bilgi tabanında tanımlı işlem planını üretir. CAD sistemlerinin çıktısı olarak STEP standartlarını kullanan sistemlerinin, ürünün tüm modelini tanımlamada ve CAD, CAM ve Bilgisayar Destekli İşlem Planlama (CAPP) iletişiminde yeterli olduğunu savunmuştur [18].

Schaler ve Mellor tarafından geliştirilen Nesne Yönelimli Sistem Analizi (OOSA) metodu, sistemin yapısal ve fonksiyonel bir tarifnamesini sağlar [3]. Metot üç önemli seviyeden oluşur (Şekil 1):

- Bilgi Modeli
- Durum Modeli
- İşlem Modeli

Bilgi Modeli üç seviyeye ayrılmaktadır; ilgilenilen konunun alanları olarak bölünen alt sistemler olarak *Sistem Modeli*; her alt sistemi nesneler olarak tanımlayan *Alan Modeli*; ve nesneleri, durumlar olarak tanımlayan *Bilgi Modeli*.

Durum Modeli, işlemlere ayrıstırılan ve Durum-Olay matrisi ile ilişkili durumları ve işlemleri tanımlar. İşlem Modelini somutlaştırır.

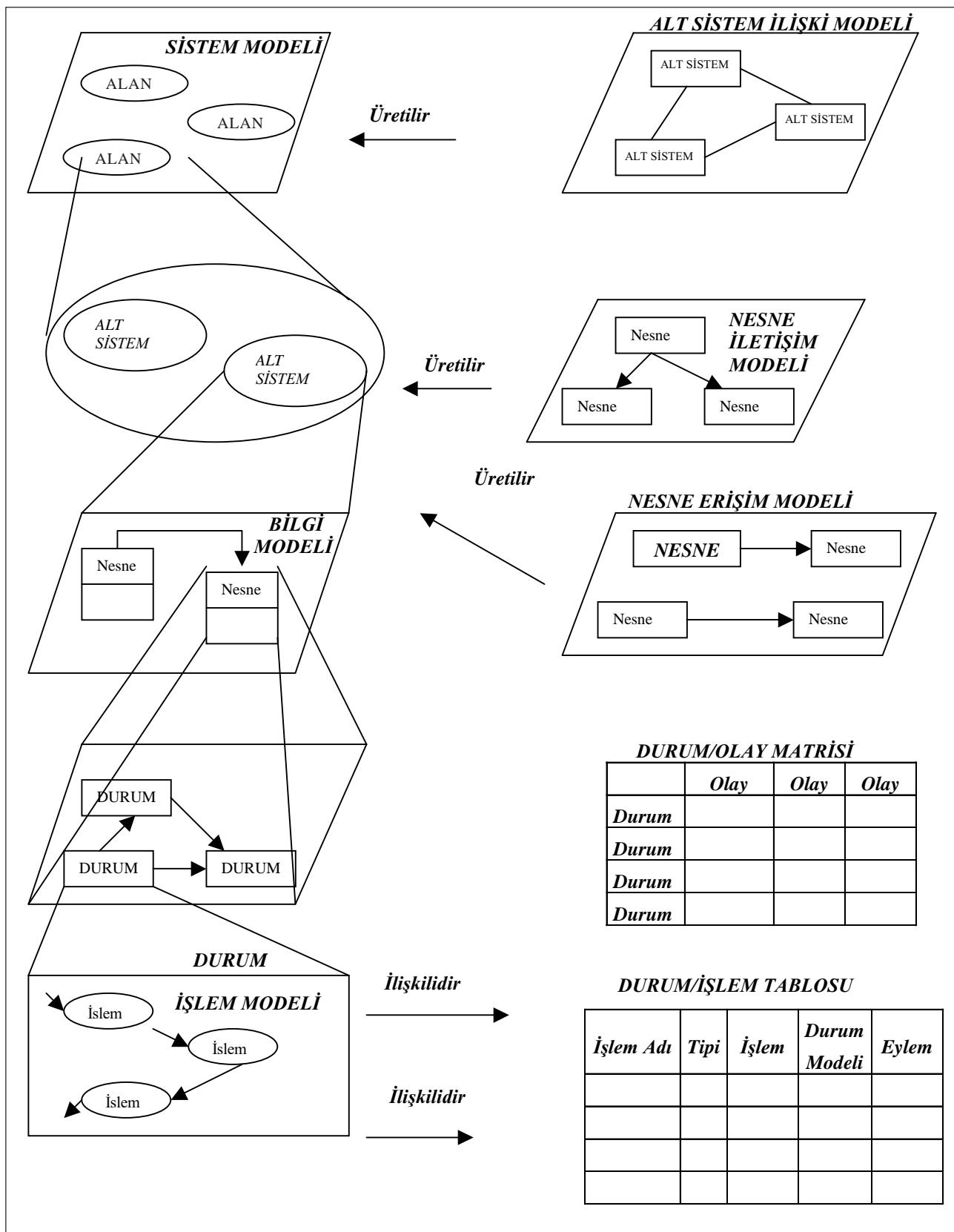
İşlem Modeli, çeşitli durumlar ile ilişkili her faaliyet için bir eylem çerçevesi tanımlar.

Model İçindeki Alt Sistem İlişki Modeli, alt sistem bileşenleri arasındaki ilişkiyi tanımlar.

Nesne İletişim Modeli, Alan Modeli ile ilişkili olup nesneler arasında eş zamanlı olmayan ilişkileri tanımlamaktadır. Nesneler arasındaki mevcut hatlardır.

Nesne Erişim Modeli, Alan Modelinden çıkartılır ve nesneler arasındaki eş zamanlı iletişimini tanımlar. Nesneler arasındaki mevcut ilişkiye sebep olan ve bu ilişkiye etkileyen bilgiyi içerir.

Usher ve Fernandes işlem planlama için takım seçiminin nesne yönelimli bir uygulaması üzerinde çalışmalar yapmışlardır. Geliştirilen "Nesne Yönelimli Takım Seçme Uygulaması (OATS)" adını verdikleri takım seçme sistemi, PARIS işlem planlama sisteminin bir parçasıdır. Model bir ISO STEP uygulama protokolü



Şekil 1. Schlaer ve Mellor'un Nesne Yönelimli Sistem Analizi Metodu

olan AP224'e göre nesne yönelimli parça tanımlaması yapar ve parça temsili için nesne yönelimli bir hiyerarşiyi gösterir. Parça; boyutlar, toleranslar ve özelliklere göre talaş kaldırma özellikleri ile ilişkilendirilerek tanımlanır. Her özellik için nesneler kapsüllenerek, o özelliği işlemek için kullanılabilecek takım tiplerini tanımlar [11].

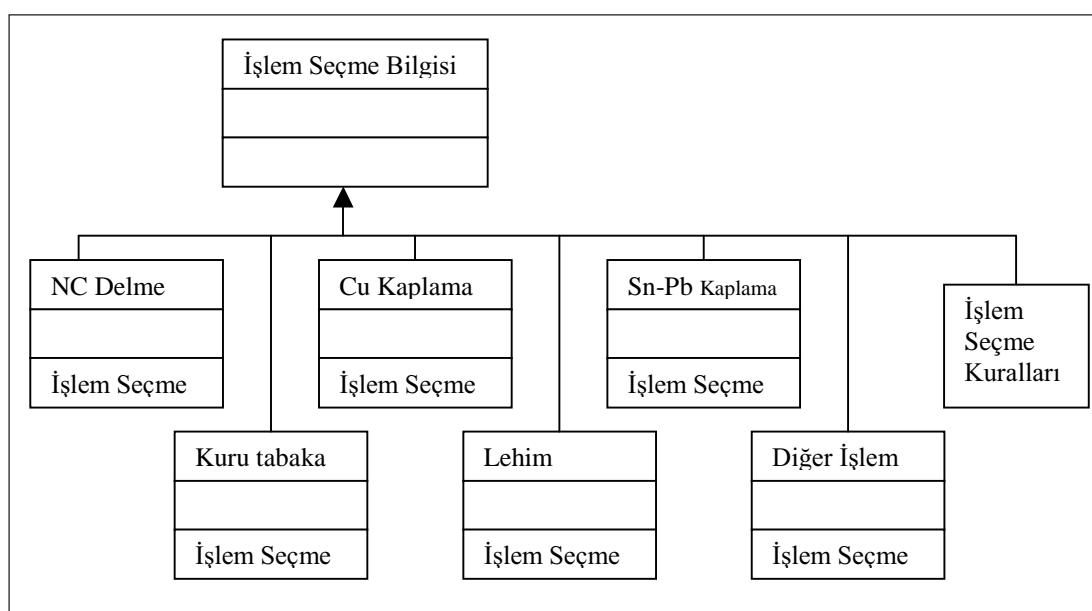
Hang ve arkadaşları, nesne yönelimli yaklaşım ile bir bilgi tabanlı sistem oluşturarak devre levhalarının imalatı için işlem planlama sistemi oluşturmuşlardır. Nesne Modelleme Tekniği (OMT) kullanılarak, sistem bileşenlerini ve gereksinimlerini modellediler. Sistemde model kısıtlamaları, planlama bilgisi, sonuç çıkarma mekanizması ve kullanıcı arayüzü nesneler olarak tanımlandılar. Ayrıca sistemin fonksiyonel bir modeli de oluşturulmuştur. Şekil 2 oluşturulan işlem planlama bilgisine ait "işlem seçme bilgisi" nin nesne modelini göstermektedir [12].

- Prosedürel programlama
- Modüler programlama
- Nesne-yönelimli programlama

Yapısal olmayan programlama, bir temel programdan oluşan ve program boyunca global olarak değişen komutları ve ifadeleri içerir.

Prosedürel programlamada bilgi program içinde oluşturulan prosedürler tarafından işlenir. Başka bir ifadeyle temel program, prosedürlere ayrılar ve bilgi işlenir.

Modüler programlama tekniğinde ortak fonksiyonelliklere sahip prosedürler ayrı modüller şeklinde gruplanarak bir bütünü oluştururlar. Her grup kendi verisini ve prosedürlerini içeren ve modül adı verilen birimlerdir. Nesne yönelimli programmanın temelini oluşturur.



Şekil 2. Devre Levhası İçin Oluşturulan Nesne Yönelimli İşlem Seçme Bilgisi Modeli

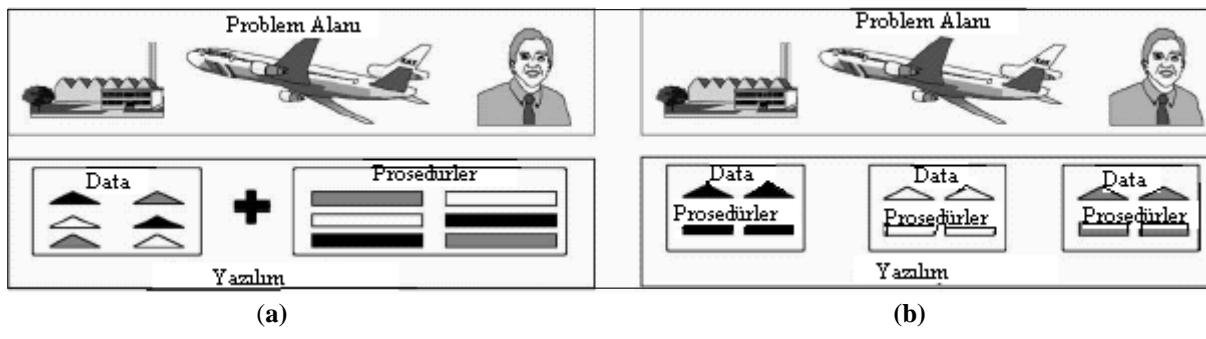
NESNE YÖNELİMLİ MODELLEME

Bilgisayar programlarının oluşturulma tekniklerine bakıldığından, genel olarak dört gruba ayrıldıkları görülür. Bunlar;

- Yapısal olmayan programlama

Nesne yönelimli programlamada, sistemi oluşturan unsurların her biri bir nesne olarak düşünülür. Bu unsurlar aslında modüllerdir. Her nesne değişkenler olarak verilerini ve metodlar olarak prosedürlerini içerecek şekilde çerçevelenir. Bu nesne yönelimli programlama tekniğinde "kapsülleme" olarak adlandırılmaktadır. Ayrıca ortak veri

ve metotlara sahip nesneler sınıfları oluştururlar. Nesneler arasında alt sınıf ve üst sınıf ilişkileri belirlenir. Şekil 3, Geleneksel yaklaşım ve Nesne yönelimli yaklaşım arasındaki farklılığı göstermektedir.



Şekil 3. Geleneksel (a) ve Nesne Yönelimli Yazılım (b) Yapısı

Nesne yönelimli modelleme, ilgilenilen problemin sistematik yapısını oluşturacak şekilde bilgisayar programının yazılması ve sistem modeli oluşturulmasıdır. Sistem, üzerinde çalışan programlama konusudur.

Nesne Yönelimli Modelleme Kavramları

Nesne yöneliminin anlaşılabilmesi için nesne, mesaj, sınıf, kalit ve arayüz kavramlarının bilinmesi gereklidir.

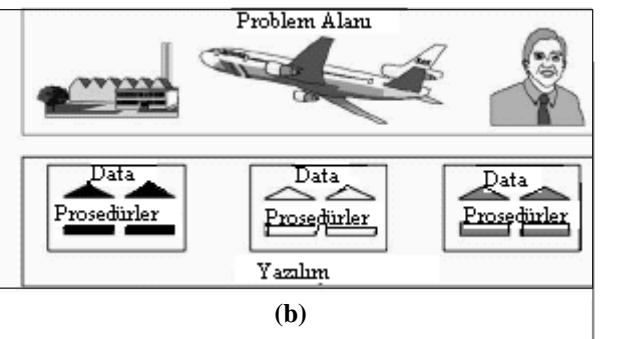
Nesne

Nesneler, nesne-yönelimli teknolojiyi anlamak için anahtarlarıdır. Etrafımızda gerçek dünyanın birçok nesne örneğini görebiliriz. Örneğin; bir köpek, televizyon seti, bisiklet vs. birer nesnedirler. Nesne; bilgiye dayalı, sistemin bir veya birkaç özellikle karakterize edilebilen elemanıdır [17].

Gerçek dünyanın nesneleri iki karakteri paylaşır: durum (state) ve davranış (behavior). Örneğin köpek durum olarak bir ada, renge, cinse ve açlık tokluğunu; davranış olarak havlamaya, alıp getirmeye ve kuyruk sallamaya sahiptir. Bisiklet ise durum olarak dişliye, pedala, iki tekerlege; davranış olarak frenlemeye, ivmelenmeye ve vites değiştirmeye sahiptir.

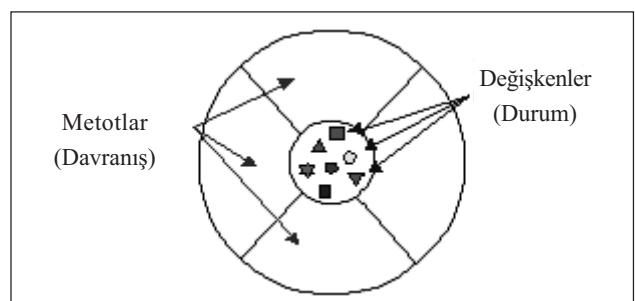
Yazılım nesneleri de gerçek dünya nesneleri gibi durumları ve davranışları oluşturulduktan sonra modellenirler. Bir yazılım nesnesi bir veya daha çok değişken (variable) şeklinde kendi durumunu içerir.

Değişken, bir tanımlayıcı tarafından isimlendirilmiş bir veri nesnesidir. Bir yazılım nesnesi metotlar ile kendi davranışlarını belirtir. Bir metot, nesne ile ilişkili bir fonksiyondur (alt iş programıdır). Şekil 4.



Mesaj

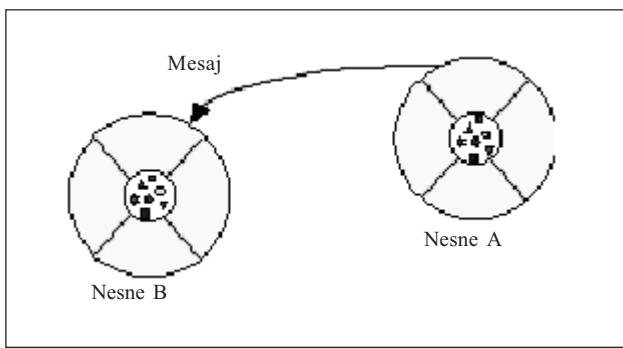
Bir nesne genelde yalnız başına çok kullanışlı değildir. Çünkü bir nesne, genelde diğer birçok nesneyi içeren geniş bir programın veya uygulamanın bir bileşeni olarak görülür. Nesnelerin etkileşimi sayesinde programcılar daha yüksek tertipteki fonksiyonları ve karmaşık davranışları modelleyebilirler. Garajdaki bir bisiklet yalnızca çelik ve lastikten oluşur; bisikletiniz kendisi tarafından herhangi bir aktivite için yeteneksizdir. Bisiklet sadece onun ile ilişkili başka bir nesne (kullanıcı) olduğunda faydalıdır.



Şekil 4. Nesne

Yazılım nesneleri mesajlar göndererek birbirlerini etkilerler ve iletişim kurarlar. A nesnesi, B nesnesinde B'nin metotlarından birisini yapmak istedığında, A nesnesi B nesnesine bir mesaj gönderir. Şekil 5.

Nesneleri kavramak, ne yapacaklarını tam olarak



Şekil 5. Nesneler Arası Mesaj Geçme

bilmek için daha çok bilgi gereklidir; örneğin bisikletinizde vites değiştirmek istediğinizde hangi vitesi istediğinizinizi belirtmelisiniz. Bu bilgi parametre olarak mesajın yanında belirtilmelidir.

Bir mesaj üç parçadan oluşur. Bunlar;

1. Gönderilecek mesaj nesnesi (bisiklet)
2. Yapılacak metodun adı (dişli değiştirme)
3. Metodun ihtiyaç duyduğu herhangi bir parametre (kaçinci vites, vs)

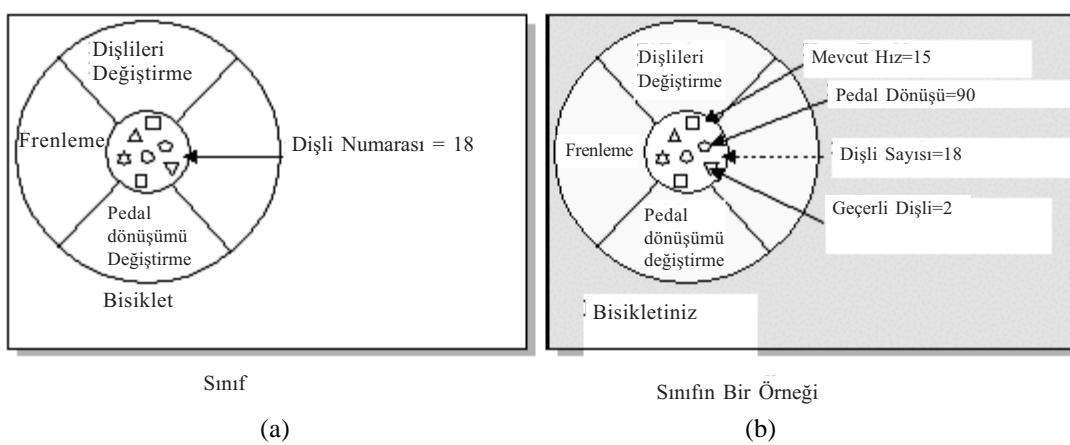
Sınıf

Gerçek dünyada genelde birçok benzer tip nesne vardır. Örneğin; bir bisiklet bütün dünyadaki birçok bisikletten sadece birisidir. Nesne yönelimli terminoloji kullanarak, bisikletimiz, bisikletler olarak bilinen nesneler sınıfının bir örneğidir. Bisikletler genelde mevcut dişli, mevcut pedal dönüsü, iki tekerlek gibi bir takım

durumlara ve vites değiştirme, frenleme gibi davranışlara sahiptir. Bununla birlikte, her bisikletin durumu bağımsızdır ve diğer bisikletlerden farklı olabilir. Bir sınıf benzer özelliklerini, operasyonlarını, ilişkileri ve ağları paylaşan nesneler setinin bir tanımlanmasıdır [20].

Bisiklet örneğimiz için sınıf, her bisiklet nesnesi için geçerli dişliyi, geçerli pedal dönüşünü vs. içeren gerekli örnek değişkenlerini bildirecektir. Sınıf aynı zamanda Şekil 6'da görüldüğü gibi vites değiştirme, frenleme ve pedal dönüşünü değiştirme için biniciye olanak sağlayan örnek metodların uygulamalarını bildirecek ve sağlayacaktır.

Nesne değişkenlerine ilave olarak sınıflar, sınıf-değişkenlerini tanımlayabilir. Bir sınıf değişkeni sınıfın bütün örnekleri tarafından paylaşılan bilgiyi içerir. Örneğin, bütün bisikletlerin aynı dişli sayısına sahip olduklarını varsayar. Bu durumda, dişilerin aynı sayısını gösteren bir örnek değişkeni tanımlamak yeterlidir; her varlık değişkenin kendi kopyasına sahip olacaktır fakat değeri her örnek için aynı olacaktır. Böyle durumlarda, dişli sayısını veren bir sınıf değişkeni tanımlayabilirsiniz. Bütün nesneler bu değişkeni paylaşırlar. Eğer bir nesne değişkeni değiştirirse, değişken bu tip diğer bütün nesneler için değişir. Bir sınıf aynı zamanda sınıf_metotlarını kapsar. Belirli bir örnek üzerinde örnek metodları istemek gerekiğinde doğrudan sınıfın bir sınıf metodu istemelisiniz.



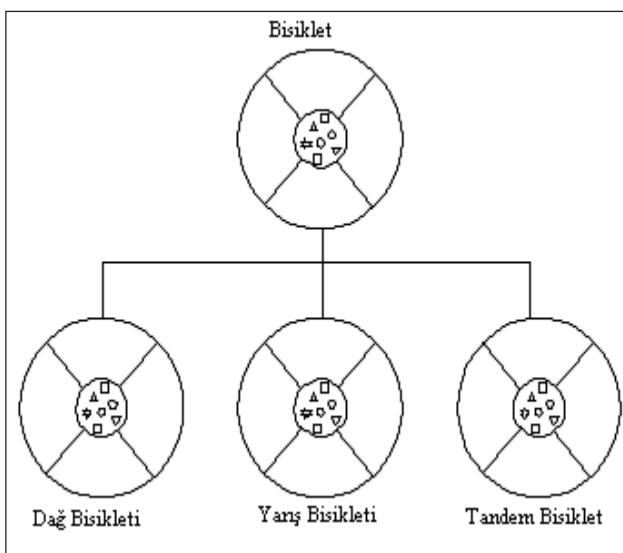
Şekil 6. Sınıf (a) ve Sınıfın nesnesi (b)

Kalıt

Genel bir prensip olarak, nesneler sınıflara göre tanımlanırlar. Sınıfını biliyorsak bir nesne hakkında çok şey biliyoruz. Eğer torna tezgahının ne olduğunu bilmeyeseniz bile, eğer size onun bir bisiklet olduğunu söyleyirse, siz onun iki tekerlege, bir çubuk direksiyona ve pedallara sahip olduğunu düşünürsünüz.

Nesne yönelimli sistemler daha ileri bir durum alırlar ve diğer sınıflara göre sınıfların tanımlanmasını da sağlarlar. Örneğin, dağ bisikleti, yarıs bisikleti ve tandem bisiklet, bisikletlerin çeşitli türleridirler. Nesne yönelimli terminolojide, dağ bisikletleri, yarıs bisikletleri ve tandemler, genel bisikletler sınıfının alt sınıfıdır.

Benzer şekilde bisiklet sınıfı, dağ bisikletleri, yarıs bisikletleri ve tandem bisikletlerin üst sınıfıdır. Bu ilişki Şekil 7'de gösterildi.



Şekil 7. Bisiklet Sınıfı

Her alt sınıf, üst sınıfın durumları (değişkenler olarak) olarak kalıtsar. Dağ bisikleti, yarıs bisikleti ve tandemler bazı durumları paylaşır: pedal dönüsü, hız ve benzeri. Aynı zamanda her alt sınıf, üst sınıfın metotlar kalıtsar. Dağ bisikleti, yarıs bisikleti ve tandemler bazı davranışları paylaşır: örneğin frenleme ve pedal dönüş hızını değiştirmeye gibi.

Arayüz

İngilizce'de bir arayüz, ilişkisiz varlıkların etkileşim için kullandıkları bir araç veya bir sistemdir. Bu tanımlamaya göre bir uzaktan kumanda siz ve televizyon setiniz arasında bir arayüzdür, Türkçe iki insan arasında bir arayüzdür, ve askerlikte yapılan davranış protokollerini farklı sınıftaki insanlar arasında bir arayüzdür. Java programlama dilinde bir arayüz, ilişkisiz nesnelerin birbirleri ile etkileşim için kullandıkları bir araçtır. Gerçekte nesne-yönelimli diller arayüzlerin fonksiyonelliğine sahiptirler ve arayüzlerinin protokollerini çağırırlar.

Arayüz, bilgilerin soyutlanmasıyla oluşturulan nesnelerin dışarıya verdikleri operasyonların setidir. Nesnelerin bilgi yapısına tanımlanan operasyonlar ile erişilir. Nesnenin dışarıdan görülen kısmı arayüzüdür.

Bisiklet diğer dünya varlıklarını ile etkileşim kurar. Örneğin bir depoda ki bir bisiklet bir arayüz olarak bir envanter programı tarafından yönetilebilecektir. Bir envanter programı, her bisikletin fiyatı ve seri numarası gibi kesin bilgileri verir. Bağımsız parçalar için sınıf ilişkilerinden ziyade, envanter programı bir iletişim protokolü sağlar. Bu protokol bir arayüz içinde bulunan sabit ve metot tanımlamalarının bir seti formundadır. Envanter arayüzü düzen, perakende satışı belirleme ve montaj, seri numarası tahsis etme gibi metotları tanımlayacak fakat uygulamayacaktır.

NESNE YÖNELİMLİ MODELLEME YAKLAŞIMIYLA ÖZELLİK TABANLI MODELLEME

Bir özellik-tabanlı modelde CAD datası özellikler olarak tanımlanabilir. Örneğin prizmatik parçalar için imalat özellikler; açık kanallar, kör kanallar, kademeler, cepler, delikler, vs. dirler [16]. Özelliklerin kolay tanımlanması için nesne-yönelimli düşünuceleri kullanarak bir özellik-tanımlama birimi düşünmek gereklidir. Özellik tanıma, veri tabanında depolanan CAD geometrik modelini test ederek, bir geometrik modelde bulunan

yüzeyler, kenarlar, köşeler ve ilkel katılar gibi basit varlıklardan kanallar, delikler, cepler ve çıkışları gibi şekil özelliklerini çıkartmaya çalışır. Özellik tabanlı tasarımında özellikler ve tüm diğer imalat bilgileri, tasarım işlemi sırasında belirlenir [18]. Özellik modelleme, bilgi modellemenin özel bir uygulamasını temsil eder [21].

Nesne-yönelimli özellik modelleme birimi, nesne-yönelimli düşüneleri kullanarak temel parça-konstrüksiyon elemanları şeklinde parça özelliklerinin bir grubunu sağlar. Aynı zamanda özellik-tabanlı parça tanımlama için bu birim içinde bir hibrit B-rep/CSG temsil şeması önerilir. Parça ve özellik tanımlamaları için nesne-yönelimli düşüneler uygulamalı olarak Şekil 8'de gösterilmiştir [3].

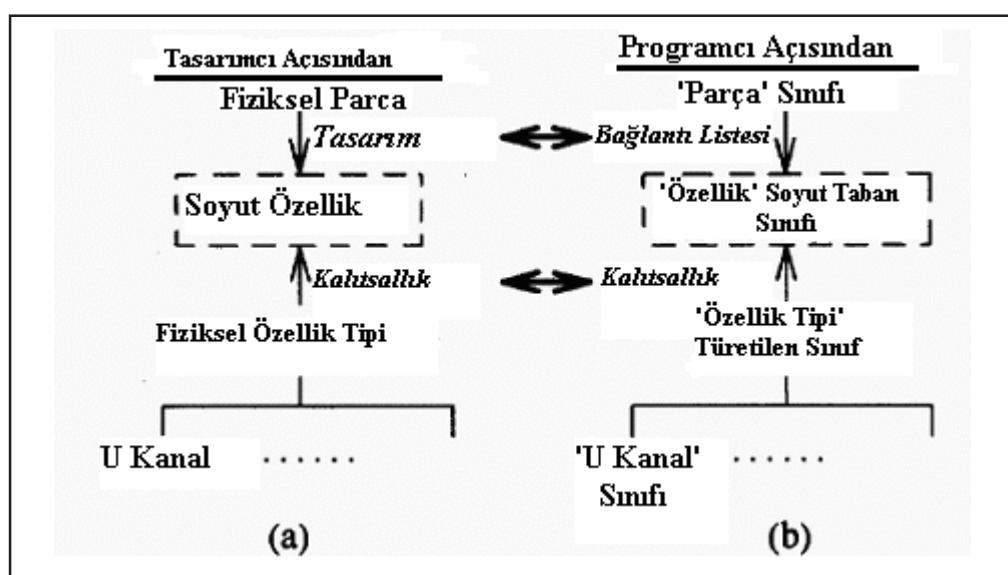
Tasarımcı Açısından Nesne-Yönelimli Özellik Tanımlama

Nesne-yönelimli düşünce olarak Şekil 8(a)'da, fiziksels parça ve soyut özellik arasındaki ilişki ve soyut özelliği oluşturan fiziksels özellik-tipleri arasında ki "kalıtsallık" ilişkisi görülmektedir. Parça tasarımcıları genelde, özellikleri genelde talaş kaldırma operasyonları tarafından uzaklaştırılacak hacimler olarak; ve ham parçadan

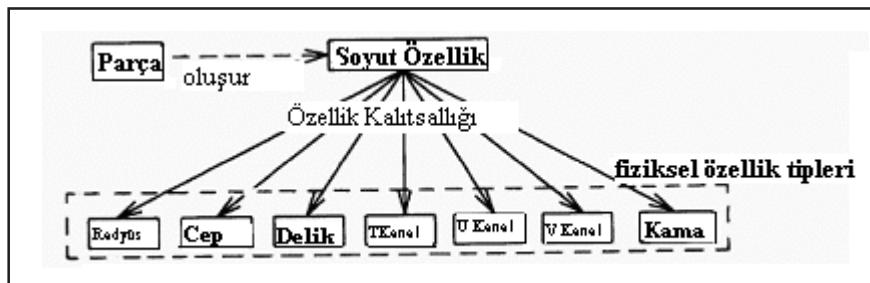
çıkartılacak özellikleri nesne kabul ederek bir özellik-tabanlı parçayı tanımlarlar. Böylece konstrüktif özelliklerinden biraz farklı olan hacimsel nesneler olarak özellikler tanımlanır [13]. Daha sonra fiziksels parça, soyut özelliklerden çıkarılan hacimler olarak ele alınabilir.

İmalat açısından parçaları, dış şekillerine göre genel olarak dönel ve prizmatik parçalar olmak üzere iki sınıfa böölünebiliriz. prizmatik parça sınıfları eğriler, kanallar, delikler, T-kanallar, U-kanallar, V-kanallar ve Kama kanallarını özelliklerine sahip olarak düşünürler [14]. Bütün özelliklerin yönelimleri koordinat eksenlerine paralel, dik veya belirli açıda kabul edilirler. Soyut özellikler yedi fiziksels özellik tipi tarafından kalıt olarak alınırlar. Parça-özellik ilişkilerinin bu yapı diyagramı Şekil 9'daki gibi tanımlandı.

Şekil 10, özellikler tarafından oluşturulan bir prizmatik parça örneğini gösterir. Her özellik, parça tasarımcısının bakışıyla bir nesne olarak ele alınır. Şekil 10'daki basit parça, Şekil 8(a)'da tanımlanan fiziksels parça olarak tanımlanır ve genel form içinde sekiz özellikten oluşan belirlenir. Fiziksels özellik-tipi ile ilgili nesneler Eğri 1, Eğri 2, Eğri 3, Eğri 4, Delik 1, Delik 2, U kanal 1, ve U kanal 2' dirler.



Şekil 8. Parça Tasarımcı Açısından (a) ve Programlamacı Açısından (b), Nesne-Yönelimli Düşünceler ile “Nesne” ve “Sınıf” Arasındaki İlişkiler



Şekil 9. Parça-Özellik İlişkisinin Yapısal Diyagramı

Programlamacı Açısından Nesne-Yönelimli Özellik Tanımlama

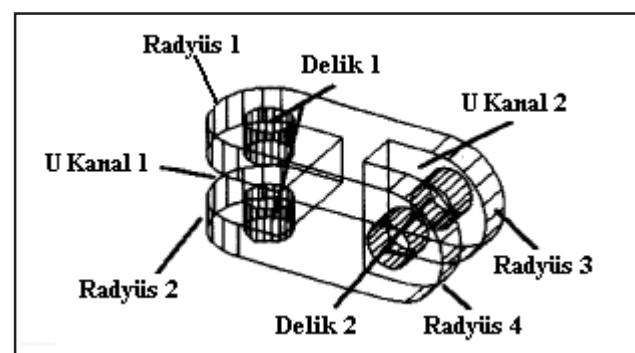
C++ programlama dili, fiziksel parçalar ve soyut özellikler arasındaki ilişkilerin; ve soyut özellikler ve fiziksel özellik tipleri arasındaki kalıtsallığın tanımlanmasını güçlü bir şekilde destekler [3]. Şekil 8(b) de görüldüğü gibi konstrüksiyon ilişkisi, soyut özelliklerin birleşiminden oluşan "Parça" isimli bir sınıf olarak fiziksel parça tanımlanabilir. Diğer taraftan, "Özellik", soyut temel sınıf olarak adlandırılır (Şekil 11(a)). "U Kanal" örneğinin türetilen bir fiziksel özellik-tipi, kalıtsallık-ilişkisi uygulamasını açıklamak için Şekil 11(b) de verildi. U-Kanal örneğinin türetilen fiziksel özellik-tipi örneği, özel özelliklerden ve gerçek fonksiyonlardan oluşan bir nesne olarak oluşturulurken; temel sınıf, genel özelliklerden ve gerçek fonksiyonlardan oluşan bir nesne olarak uygulanır. Türetilen sınıf, temel sınıfın genel özelliklerini miras alacaktır. Diğer fiziksel özelliklerin tipleri benzer şekilde tanımlanabilirler. "U Kanal" özellik-tipi örneği tarafından "Özellik", soyut temel sınıfının kalıtsanız özelliğinin operasyonu ve gerçek fonksiyonları aşağıdaki gibi açıklanır.

Şekil 11'de "sınıf" kelimesi "Özellik" isimli yeni bir bilgi tipi tanımlamak için kullanılır. Bir sınıf, bir nesnenin statik özelliklerinin (soyut özellikler) ve dinamik davranışlarının (gerçek fonksiyonlar) toplamıdır. "U kanalı" gösteren Şekil 11(b) deki "public özellik", temel sınıfın genel özelliklerini kalıtsar ve sınıf türetilir. Yani "Özellik" sınıfının datası, türetilmiş "U Kanal" sınıfı tarafından kalıt olarak paylaşılabilir.

(1) Genel Özellikler

Özellikleri inceleyerek, her fiziksel özellik-tipinin özellik

tipi adı, referans noktası, yön ve ilişkili Boolean operatörleri gibi bazı genel özelliklere sahip olduğunu görürüz. Bu genel özellikler, sistem içinde "Genel Özellik" ismiyle temel sınıf içinde toplanır. Böyle bir uygulama özellik-nitelığının kalıtsama işlemini kolaylaştırır [15].



Şekil 10. Eğri, Delik ve U-Kanal Özellikleri Tarafından Tanımlanan Bir Prizmatik Parça Örneği

(2) Özel özellikler

Genel özelliklerin yanında her fiziksel özellik-tipi kendi özel özelliklerine sahip olabilir. Örneğin, "Delik" özelliği çap ve derinliğe sahip olurken "U-Kanal" özelliği genişlik, yükseklik ve derinlik özel özelliklerine sahip olabilir. Yukarıda söz edilen genel özellikler farklı özellik-tipi nesneleri tarafından miras alınabilir ve yeniden kullanılabilir fakat özel özelliklere sadece bu özel özellikleri tanımlayan özellik-tipi nesnesi tarafından erişilebilir. Özellik kütüphanesinin sürdürülme, genişleme imkanı olarak yeni özellik-tiplerinin ek özel özellikleri eklenebilir.

(3) Gerçek fonksiyonlar

Tanımlanan bir özellik sınıfının gerçek fonksiyonları,

nesnenin özellik-tipine bağlıdır. Fiziksel özellik-tipi sınıfında tanımlı gerçek fonksiyon, giriş özellik parametreleri, çıkış özellik parametreleri, 3D B-rep data üretme fonksiyonları vs. içerir. Gerçek fonksiyonların kullanılması, özellik-tanımlama program tasarımını kolaylaştırır ve çok yönlü program yapmaya yardım eder [15].

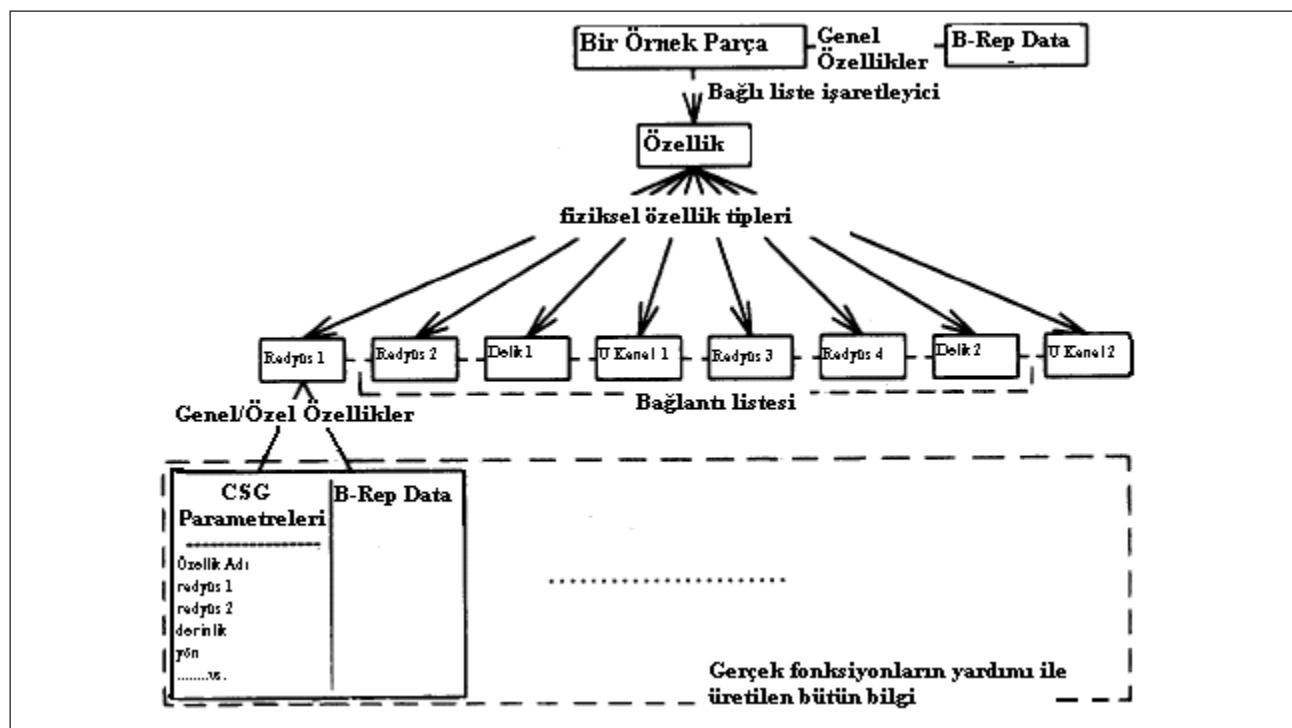
Parçanın geometrik tanımlaması bir hibrit B-rep/CSG formunda yapılabilir. CSG parametreleri tanımlanan her özellik için genel/özel özellikleri içerirken, parça ve özelliklerini içeren B-rep datası ACIS formatında temsil edilir. Şekil 10'daki basit parçanın Şekil 12'de bir programcı açısından hibrit B-rep/CSG temsil şeması gösterilir. Fiziksel parçalar ve özellik-tipleri

Simif özelliği/* fiziksel özellik-tipi nesnesinin Genel özelliklerini içeren bir genel temel sınıfı*/ {	U Kanal sınıfı: public Özellik /* 'Özellik' temel sınıfının genel özelliklerini kalıtsayan ve türetilen bir fiziksel özellik-tipi*/ {
Soyut özellik	Özel özellik
özellik adı referans noktası yön boolean operatörü özellik B-rep gösterimi parça B-rep gösterimi özellik çizgisi liste gösterimi	genişlik yükseklik derinlik tür
Gerçek fonksiyon	Gerçek fonksiyon
özellik parametreleri girişi B-rep datası üretme özellik dosya çıkışı	özellik parametreleri girişi B-rep datası üretme özellik dosyası çıkışı
};	};

(a) soyut taban sınıfı "Özellik" tanımlama

(b) türetilmiş 'U Kanal' tanımlama

Şekil 11. Soyut Taban Sınıfı "Özellik" ve türetilmiş "U Kanal" İ Tanımlama İçin Örnek Tanımlamalar



Şekil 12. Şekil 10'daki Basit Parça İçin Bir Hibrit CSG/B-Rep Temsil Şeması

böylece nesne-yönelimli programlama düşünelerini kullanarak basit bir şekilde uygulanırlar.

SONUÇ

Bu çalışmada nesne-yönelimli özellik tabanlı parça modelleme konusunda yapılan çalışmalar ve konu ile ilgili kavramlar tanıtıldı. Programlama teknikleri incelendi ve nesne-yönelimli programların farklılığı ve avantajları örnek parça üzerinde gösterilerek vurgulanmaya çalışıldı. Nesne yönelimli özellik tabanlı modelleme, özellikle Bilgisayar Destekli İşlem Planlama alanında başarılı olarak uygulanabilir. Oluşturulan modeller sistematik olarak parça, imalat ve gerekli diğer bilgilerin tam temsilini sağlayabilir ve modeller güncellenebilir. Nesne yönelimli yaklaşımla yapılan modelleme aynı zamanda bir CAD/CAM ortamında yapılan özellik tabanlı modelleme sisteminin karmaşıklığının çözülmesi için mükemmel bir araç sağlar. Esneklik, sistemin iç uyumu, özelliklerin kalıtsallığı, bilginin soyutlanması, bilginin ve yöntemlerin kapsülleme nesne-yönelimli modellemeyi üstün kılan başlıca konulardır.

KAYNAKÇA

1. **Case, K. And Gao, J.**, "Feature Technology an Overview", Symposium on Future Based Approaches to Design and Process Planning", Loughnorrough, U.K., 1991, pp.1-14.
2. **Chep, a., Tricarico, L., Bourdet, P. and Galuntuccce, L.**, "Design of Object Oriented Database for The Definition of Machining Operation Sequences of ^d Workpieces", Computers Industrial Enginering, Vol.34, No.2, U.K., 1998, pp.257-279.
3. **Pern, D.B. and Chang, C.T.**, "A New Feature-Based Design System with Dynamic Editing", Computer Industrial Enginering, Vol.32, No.2, U.K., 1997, pp.383-393.
4. **Motavalli, S., Cheraghchi, S.H. and Shamsaasef, R.**, "Feature-Based Modelling; An Object Oriented Approach", Computer Industrial Enginering, Vol.33, No.1-2, U.K., 1997, pp.349-352.
5. **Chang, C.T.**, "Expert Process Planning for Manufacturing", 1990
6. **Gu, P.**, "A Feature Representation Scheme for Supporting Integrated Manufacturing", Computer Industrial Enginering, Vol.26, U.K., 1994, pp.55-71.
7. **Dong, J. and Parsaei, H.R.**, "Design and Implementation of a Feature-Based Process Planning", Computer Industrial Enginering, Vol.27, No.1-4, U.K., 1994
8. **Dave, P. and Sakurai, H.**, "Maximal Volume Decomposition and Recognition of Interseeting", Proc. ASME Computers in Engineering Conf., 1995, pp.553-568.
9. **Regli, W.C., Gupta, S.K. and Nad, D.S.**, "Feature Recognition for Manufacturability Analysis", In ASME Computers in Engineering Conf., 1994, pp.93-104.
10. **Terpenny, J.P., Bartholomew, O.N.**, "Feature-Based Design Evaluation for Machine/Tool Selection for Sheet Metal", 2nd Industrial Engineering Research Conference Proceedings, HE, pp.26-30
11. **Usher, J.M. and Fernandes K.J.**, "An Object-Oriented Application of Tool Selection in Dynamic Process Planning", International Journal of Production Research, Vol.37, No.13, 1999, pp.2879-2894.
12. **Law, H.W., Tam, H.Y., Chan, A.H.S. and Hui, I.K.**, "Object-Oriented Knowledge-Based Computer-Aided Process Planning System for Bare Circuit Boards Manufacturing", Computers In Industry, Elsevier, 2001.
13. **Butterfield, W.B., Green, M.K., Scott, D.C. and Stoker, W.J.**, "Part Features for Process Planning", Report-R86-PPP-01, CAM-I, Incorporated, 611 Ryan Plaza Drive, Suite 1107, Arlington, TX 76011, 1986.
14. **Pern, D.B., Chan, Z. and Li, R.K.**, "Automated 3D Machining Feature Extraction from 3D CGS Solid Input", Computer Aided Design, 1990
15. **Lippman, S.B.**, "C++ Primer, 2nd end.", Addison-Wesley, Reading, MA., 1991
16. **Tseng, Y.J. and Jiang, B.C.**, "Evaluating Multiple Feature-Based Machining Methods Using an Activity Based Cost Analysis Model", International of Journal Advanced Manufacturing Tcehnology, No.16, 2000, pp.617-623.
17. **Allahverdi, N.**, "Uzman Sistemler", Atlas Kitabevi, 2002
18. **Yıldız, S.**, "Uzman Sistem Yaklaşımıyla Prizmatik Parçalar İçin İşlem Planlama Sistemi", Doktora Tezi, Selçuk Üniversitesi, 1996.
19. **Aslan, E.**, "Uzman Sistem Yaklaşımıyla Son İşlemci Tasarımı", Doktora Tezi, Gazi Üniversitesi, 1995.
20. **Hvam, L., Riis, J. and Hansen, B.J.**, "CRC Cards for Products Modelling", Computer in Industry, No.50, 2003, pp.57-70.
21. **Ma, Y.-S. and Tong, T.**, "Associative Feature Modelling for Concurrent Engineering Integration", Computers In Industry, February, 2003.