T.C

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SAÇ KESME İŞLEMLERİ İÇİN BİR BİLGİSAYAR DESTEKLİ TASARIM VE ÜRETİM SİSTEMİNİN GELİŞTİRİLMESİ

Fazıl Saray

Yüksek Lisans Tezi

Elektronik Ve Bilgisayar Sistemleri Anabilim Dalı

Konya, 2007

T.C

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SAÇ KESME İŞLEMLERİ İÇİN BİR BİLGİSAYAR DESTEKLİ TASARIM VE ÜRETİM SİSTEMİNİN GELİŞTİRİLMESİ

FAZIL SARAY

YÜKSEK LİSANS TEZİ

ELEKTRONİK VE BİLGİSAYAR SİSTEMLERİ ANABİLİM DALI

Bu tez 09.03.2007 tarihinde aşağıdaki jüri tarafından oybirliği / ~~oyçokluğu~~ ile kabul edilmiştir.

Prof.Dr. Süleyman YALDIZ	Prof.Dr. Novruz ALLAHVERDİ	Doç.Dr. Hacı SAĞLAM
(Danışman)	(Üye)	(Üye)

ÖZET

Yüksek Lisans Tezi

SAÇ KESME İŞLEMLERİ İÇİN BİR BİLGİSAYAR DESTEKLİ TASARIM VE ÜRETİM SİSTEMİNİN GELİŞTİRİLMESİ

Fazıl SARAY

Selçuk Üniversitesi Fen Bilimleri Enstitüsü

Elektronik ve Bilgisayar Sistemleri Eğitimi Anabilim Dalı

Danışman : Prof.Dr.Süleyman YALDIZ

2007, 109 Sayfa

Jüri : Prof.Dr.Süleyman YALDIZ

Prof.Dr.Novruz ALLAHVERDİ

Doç.Dr.Hacı SAĞLAM

Sac metaller birçok alanda kullanılmaktadır. Sac metaller mamul haline gelinceye kadar farklı işlemlerden geçirilmektedir. Bunlar üzerinde yapılan işlemlerden biri de kesmedir. Bu çalışmada sac metallerin sayısal kontrollü tezgâhlarda kesilmesi işlemlerinde, üretim için planlanmasından imal edilmesi aşamasına kadar gereken tüm işlemlerin yapılmasını sağlayacak bir bilgisayar programı geliştirilmeye çalışılmıştır. Sac metal ürünlerinin geliştirilmesi işlemlerinde işlem planlama, çizim, üretim ve nümerik kontrollü tezgâh sistemlerini etkin bir şekilde bir araya getirilmesi amaçlanmıştır. Sayısal kontrollü tezgâhların yaygınlaşması ve hızlanması ile üretimde en çok zaman alan işlem üretimin yapılmasından çok öncesinde yapılan hazırlık çalışmalarıdır. Bu çalışmalar farklı programlar aracılığıyla yapıldığında daha çok zaman almakta ve programlar arasında veri transferlerinde hatalar ve kısıtlamalar oluşmaktadır. Üretim yapılacak alan ile ilgili bütün işlemlerin tek bir programda birleştirilmesi üretimi daha hızlı ve etkin hale getirecektir. Aynı zamanda hatları ve karmaşayı önleyecektir. Bu çalışmadaki işlemler; çizim, yerleştirme, işlem planlama adımlarından oluşmaktadır. Bu adımlarda farklı sayılarda alt adımları içermektedir.

Anahtar Kelimeler: bilgisayar destekli tasarım, bilgisayar destekli üretim, yerleştirme, sac kesim, ölçülendirme, CAD/CAM yazılımı, yerleştirme algoritmaları yazılımı.

ABSTRACT

Master Thesis

DEVELOPMENT OF A COMPUTER AIDED DESIGN AND MANUFACTURING SYSTEM FOR SHEET METAL CUTTING

Fazıl SARAY

Selçuk University

Graduate School of Natural and Applied Sciences

Department of Computer Systems

Supervisor : Prof.Dr.Süleyman YALDIZ

2007, 109 pages

Jury: Prof.Dr.Süleyman YALDIZ

Prof.Dr.Novruz ALLAHVERDİ

Assoc.Prof.Dr.Hacı SAĞLAM

Sheet metals are used in many areas. Sheet metals are to be passed different processes until to become product. One of the process to produce sheet metals is to cut them. In this work during this process, in the works of cutting the sheet metals on numeral control workbench, a computer programme has been tried to develop to make progress from the process of planing to producing. In the process of improving the sheet metals; process plan, drawing (CAD), production (CAM)and numeral control workbench (CNC) systems are aimed to work interactively. With the aid of the widespreading and the acceleration of numeral control workbench, the preperation takes more time than the production process. This process takes more time when it is done with different programmes, there can be defects and contraints in data transfer between programmes. All of the process related to production area combining all the programmes in one will make the production more quick and active. In addition, it can prevent defects and confusions. The process comprises of drawing, setting, process planing. These steps include different kinds of substeps.

Key Words : computer aided design, computer aided manufacturing, nesting, sheet metal cutting, measurement, CAD/CAM Software, nesting algorithm and software.

ÖNSÖZ

Genel manada üretim tarihsel olarak dört safhadan geçmiştir. Bunlar elle üretim, mekanizasyon, otomasyon ve bütünleşmedir. Günümüzde bilgisayarla bütünleşik üretim sistemleri kullanılmaktadır. Bilgisayarla bütünleşik üretim, üretime bilgisayar teknolojisinin uygulanması olarak düşünülebilir. Tasarımdan üretime kadar bütün işlemler bilgisayarlarda ilgili yazılımlar aracılığıyla yapılmaktadır. Bu yazılımlar sıfır hata ile işlem yaparak elle yapılamayacak zorluktaki problemleri güvenli bir şekilde çözebilmeli, kullanıcı dostu olmalı, yani kullanıcının yapacağı işlemleri kolaylaştırmak için bütün araçları içermeli, kullanılacağı alanda diğer programlara ait veri dosyalarını ve çizimleri açabilmeli ve bunlar üzerinde işlem yapabilmeye izin vermelidir. Genellikle yazılımlar tasarım için ayrı, üretim için ayrı, tasarlanarak kullanılmaktadırlar. Bu durumda tasarlanan şeylerin, üretim işleminin yapılmasını sağlayan yazılımlar tarafından hatasız olarak alınabilmesi ve sıfır hata ile üretimi yapacak olan tezgâhlara aktarımını yapması gerekmektedir.

Bu çalışma, parça sacların hazır halde bulunan tabaka veya rulo şeklindeki bütün saclar üzerine en uygun şekilde yerleştirilerek, kesimi yapacak olan tezgaha doğrudan aktarılabilecek bir veri formatına dönüştürülmesini amaçlamaktadır. Çalışma dört kısım olarak ele alınmıştır. 2 boyutlu CAD (Computer Aided Design-BDT Bilgisayar Destekli Tasarım), CAM (Computer Aided Manufacturing-BDT Bilgisayar Destekli Üretim), Yerleştirme (Nesting), Kütüphane (Library).

Bilgisayar destekli tasarım programları yaygın olarak günümüzde kullanılmaktadır. Ancak bu programların yazılması ve çalışması konusunda yeterince kaynak ve örnek program bulunmamaktadır. Bu da başlangıç aşamasında çalışmamız açısından bazı zorluklara neden olmuştur. Bu zorluklar, çizimin bilgisayarda nasıl ifade edileceği? Nasıl gösterileceği? Nasıl dosyada saklanacağı? vb. şeklinde karşımıza çıkmıştır. Bu soruların cevabı bu çalışmada bulunmuş gerekli yapı geliştirilmiştir. Yapılan kısım zor olan sorulara cevap vermekte ve nasıl olacağını uygulamalı olarak göstermektedir. Kalan kısımlar bu yapıya birer alt işlem olarak

rahatlıkla yerleştirilebilir. Bilgisayar destekli üretim konusunda da gerekli araştırmalar yapılmıştır. Bazı modüller geliştirilmiştir. Son aşama olan yerleştirme konusunda da gerekli araştırmalar yapılmış algoritmalar ve programa uygulanması için gerekli yapı ortaya konmuştur. Yerleştirme için gerekli bütün alt işlemler programda yapılmıştır. Ayrıca programda bunların dışında bir kütüphane yapısı oluşturulmuştur. Kütüphane yapısı kullanıcıya zamanla dosya sayısı fazlasıyla artacağı için hazır bir şema saklama düzeni sunaktadır. Ayrıca buraya kaydettiği dosaları rahatlıkla alıp programa ekleyebilmesine imkân sağlamaktadır.

İÇİNDEKİLER

ÖZET ..... **İ**

ABSTRACT ..... **İİ**

ÖNSÖZ ..... **İİİ**

İÇİNDEKİLER ..... **V**

ŞEKİLLER TABLOSU ..... **Vİİ**

1. GİRİŞ **1**

2. KAYNAK ARAŞTIRMASI **5**

3. MATERYAL VE METOT **9**

3.1. Çizim Alanı **9**

3.2. Nesnelerin Hafızada Tutulması **10**

3.3. Koordinat Sistemi **13**

3.4. Ölçülendirme **14**

3.5. Ağaç Yapısı **16**

3.6. Kütüphane Yapısı **18**

3.7. Dosya Yapısı **20**

4. GELİŞTİRİLEN TASARIM VE ÜRETİM SİSTEMİ PROGRAMI **23**

4.1. Ana Form Ekran Görünümü **23**

4.2 Dosya Menüsü **24**

4.2.1. Yeni komutu ve akış çizelgesi 25

4.2.2. Aç komutu ve akış çizelgesi 26

4.2.3. Kaydet komutu ve akış çizelgesi 28

4.2.4. Farklı kaydet komutu ve akış çizelgesi 29

4.2.5. Kütüphaneye kaydet komutu ve akış çizelgesi 30

4.2.6. Kütüphaneden ekle komutu ve akış çizelgesi 31

4.2.7. Dxf al komutu ve akış çizelgesi 32

4.3. Düzen Menüsü 39
4.3.1. Yeniden çiz komutu 39
4.3.2. Tümünü sil komutu 41
4.3.3. Uygulama tercihleri komutu 43

4.4. Çiz Menüsü 45

4.5. Komut Menüsü 51
4.5.1. Serbest çevir komutu 51
4.5.2. Yatay ters çevir komutu 57
4.5.3. Dikey ters çevir komutu 62
4.5.4. Sola 90° çevir komutu 64
4.5.5. Sağa 90° çevir komutu 65
4.5.6. Açılı döndür komutu 66

4.6. Pencere Menüsü 72
4.6.1. Kütüphane 72
4.6.2. Ağaç 81
4.6.3. Komut kutusu 85
4.6.4. Çizim araç çubuğu 86
4.6.5. Durum çubuğu 88

4.7. Görünüm Menüsü 92
4.7.1. Izgara 92
4.7.2. İşaretçi 94

4.8. Kütüphaneden Ekle Formu 95

4.9. Metin Ekle Formu 98

5. UYGULAMA 100

6. TARTIŞMA 105

7. SONUÇ VE ÖNERİLER 107

KAYNAKLAR 108

ŞEKİLLER TABLOSU

Şekil 1.1 Projenin Amacı 4

Şekil 3.1 Timage nesnesi 9

Şekil 3.2 Tarama sureci 10

Şekil 3.3 Şekiller ve tutma noktaları 12

Şekil 3.4 Koordinat sistemi 13

Şekil 3.5 Ölçülendirme 15

Şekil 3.6 Nesnelerin ağaç yapısında gösterimi 17

Şekil 3.7 Kütüphane yapısının şeması 18

Şekil 3.8 Kütüphane penceresi 19

Şekil 3.9 Nesnelerin dosyada saklanışı (çizgi, daire, yay, pline) 21

Şekil 3.10 Nesnelerin dosyada saklanışı (poligon, yazı, grup) 22

Şekil 4.1 Ana form ekran görünümü 23

Şekil 4.2 Ana form pencereleri 24

Şekil 4.3 Dosya menüsü 25

Şekil 4.4 Yeni komutu akış çizelgesi 26

Şekil 4.5 Aç komutu akış çizelgesi 27

Şekil 4.6 Kaydet komutu akış çizelgesi 28

Şekil 4.7 Farklı kaydet akış çizelgesi 29

Şekil 4.8 Kütüphaneye kaydet komutu akış çizelgesi 30

Şekil 4.9 Kütüphaneden ekle akış çizelgesi 31

Şekil 4.10 Dosya Al (*.dxf) komutu akış çizelgesi 34

Şekil 4.11 Çizgi oku komutu akış çizelgesi 35

Şekil 4.12 Daire oku komutu akış çizelgesi 36

Şekil 4.13 Yay oku komutu akış çizelgesi 37
Şekil 4.14 Yay oku komutu akış çizelgesi 38
Şekil 4.15 Düzen menüsü..... 39
Şekil 4.16 Yeniden çiz komutu akış çizelgesi..... 40
Şekil 4.17 Tümünü sil komutu akış çizelgesi..... 42
Şekil 4.18 Uygulama tercihleri formu birimler sayfası ekran görünümü 43
Şekil 4.19 Uygulama tercihleri formu renk ayarları sayfası ekran görünümü..... 44
Şekil 4.20 Çiz menüsü ekran görünümü 45
Şekil 4.21 Çizim araç çubuğu ekran görünümü 45
Şekil 4.22 Çiz/Çizgi komutu akış çizelgesi..... 47
Şekil 4.23 Çizgi seçildikten sonraki fare tuşuna ilk ve ikinci kez basıldığında oluşan akış çizelgesi 48
Şekil 4.24 Çizgi seçilip fare tuşuna basıldıktan sonra fare hareket ettirildiğindeki akış çizelgesi..... 50
Şekil 4.25 Komut menüsü ekran görünümü..... 51
Şekil 4.26 Serbest çevir komutu ekran görünümleri 52
Şekil 4.27 Serbest çevir komutu akış çizelgesi 52
Şekil 4.28 Serbest çevir komutu seçildikten fare tuşuna basıldığındaki akış çizelgesi 54
Şekil 4.29 Serbest çevir komutu seçildikten fare çizim alanında hareket ettirildiğindeki akış çizelgesi..... 56
Şekil 4.30 Yatay ters çevir komutu akış çizelgesi..... 57
Şekil 4.31 F1 komutu ile yatay ters çevir işlemi akış çizelgesi 58
Şekil 4.32 Yatay ters çevir komutu işlevi 59
Şekil 4.33 Döndür sol alt programı akış çizelgesi 61
Şekil 4.34 Komut/Dikey ters çevir komutu akış çizelgesi 62

Şekil 4.35 F2 komutu ile dikey ters çevir işlemi akış çizelgesi 63
Şekil 4.36 Dikey ters çevir komutu işlevi 63
Şekil 4.37 Sola 90° çevir komutu akış çizelgesi 64
Şekil 4.38 Sağa 90° çevir komutu akış çizelgesi 65
Şekil 4.39 Açılı döndür komutu akış çizelgesi(1) 66
Şekil 4.40 Açılı döndür komutu akış çizelgesi(2) 68
Şekil 4.41 Açılı döndür komutunun düzlemde gösterimi 69
Şekil 4.42 Açılı döndür alt programı alış çizelgesi 71
Şekil 4.43 Pencere menüsü ekran görünümü 72
Şekil 4.44 Pencere menüsü ekran görünümü 73
Şekil 4.45 Kütüphane penceresinde tasarım anında yer alan nesneler ve ekran görünümü 74
Şekil 4.46 Kütüphane komutu akış çizelgesi 75
Şekil 4.47 Kütüphaneyi güncelle komutu akış çizelgesi 78
Şekil 4.48 İmaj nesnelerinde fare tuşuna basıldığına oluşan akış çizelgesi 80
Şekil 4.49 İmaj nesnelerinde fare tuşuna iki kere basıldığına oluşan akış çizelgesi .. 80
Şekil 4.50 Pencere/Ağaç komutu akış çizelgesi 81
Şekil 4.51 Ağaç penceresi ekran görünümü 82
Şekil 4.52 Ağaç sağ tuş menüsü ekran görünümü 84
Şekil 4.53 Komut kutusu ekran görünümü 85
Şekil 4.54 Komut kutusunda yer alan nesneler 85
Şekil 4.55 Çizim araç çubuğu ekran görünümü 87
Şekil 4.56 Çizim araç çubuğunda yer alan nesneler 87
Şekil 4.57 Durum çubuğu ekran görünümü ve gösterilen nesneler 88
Şekil 4.58 Fare işaretçisinin nesne üzerine geldiğinde durum çubuğundaki üçüncü panelin ekran görünümü 89

Şekil 4.59 Fare işaretçisinin nesne boyutlandırma kutusu üzerine geldiğinde durum çubuğundaki dördüncü panelin ekran görünümü 90

Şekil 4.60 Çizgi ve daire boyutlandırılırken durum çubuğu 5. panelin ekran görünümü 91

Şekil 4.61 Yay boyutlandırılırken durum çubuğu 5. panelin ekran görünümü..... 91

Şekil 4.62 Görünüm menüsü ekran görünümü 92

Şekil 4.63 Izgara komutu akış çizelgesi 93

Şekil 4.64 İşaretçi ekran görünümü 94

Şekil 4.65 Izgara komutu akış çizelgesi 95

Şekil 4.66 Kütüphaneden ekle formu ekran görünümü 96

Şekil 4.67 Çift tıklanan bir dosyanın kütüphaneden ekle formunda gösterilmesi 97

Şekil 4.68 Kütüphaneden ekle formundan sayfaya aktar komutu seçildikten sonraki ana form akran görünümü 98

Şekil 4.69 Yazı ekle formu ekran görünümü 99

Şekil 5.1 Parçanın AutoCAD ekran görünümü..... 100

Şekil 5.2 Dxf al komutu ile programa alınan çizimin ekran görünümü..... 100

Şekil 5.3 Grup seçildiğinde oluşan ekran görünümü 101

Şekil 5.4 Kütüphaneden ekle formunda tabaka sacın ekran görünümü. 102

Şekil 5.5 Tabaka sac ve nesnenin ekran görünümü 103

Şekil 5.6 Dxf parçanın yerleştirilişi 103

Şekil 5.7 Kütüphaneden eklenen yeni nesne..... 104

Şekil 5.8 İkinci yerleşim sonucu ekran görünümü..... 104

1. GİRİŞ

CAD bilgisayar dünyasında en çok kullanılan tasarım alanlarından biridir. Bilgisayar kullanımının ana amaçları arasında bilgiyi hızlı, hatasız ve istenilen şekilde işleme gelir. CAD programları, bütün teknik bilimler için bu ana amaçları destekler [1]. CAD programları, bilgileri kullanıcıdan aldığı komutlarla birlikte her çeşit şekle, teknik resimlere, tasarımlara, haritalara ve projelere dönüştürür. Bu işlevin sonucundaki verimliliği, kullanıcının bilgi ve tecrübesi belirler. CAD programları, hazır araçlar sunarak kullanıcıların etkin ve verimli çalışmasına imkân tanır. Ayrıca Cad programları, arka planda kalan hatasız ve doğru veri biçimlerinin yanında kullanıcılar için yaptığı işlemi ekranda açık ve net olarak görebilmesi doğrudan ekran üzerinde işlemlerini yapabilmesi gerekir. Görselliğin arkasında kalan birçok işlem kullanıcıya yazılım tarafından sunulmaktadır. Bu yazılımlarda bütün CAD alanları hedeflendiği gibi, özel amaçlı yazılımlarda da kullanılmaktadır.

Yazılımlar ister özel amaçlı bir yazılım olsun, ister genel amaçlı bir yazılım olsun birbirleri arasında mutlaka veri transferi yapmaları gerekmektedir. Yazılımları üreten firmalar bu verilerin bilgisayarda nasıl bir dosyada ve ne şekilde saklanacağını belirler. Dosyaların farklı programlar arasındaki geçişleri genellikle nötr dosya biçimleri aracılığıyla yapılır. Bu amaçla geliştirilen birçok nötr dosya biçimi vardır. Bunlardan en yaygın olarak kullanılanlardan biri dxf'tir. Dxf dosya, Autocad programını yazan Autodesk firması tarafından 1992 yılında geliştirilmiştir. Autocad programının standart veri biçimi dwg'dir. Dwg dosyası, standart text editörler tarafından açılamaz. Açmak ve üzerinde işlem yapabilmek için bu yapının çok iyi anlaşılması ve tüm metotlarıyla uygulanması gerekir. Ancak dxf dosyalar text türünde dosyalardır. Açması ve üzerinde işlem yapması daha kolaydır ve bu yapıda da çizime ait bütün bilgiler yer alır. Bu dosyaların tüm verileriyle açılıp, gösterilebilmesi ve üzerinde işlem yapılabilmesi için bu dosyayı açacak programın tüm metot ve verileri içermesi gerekir. Bu çalışmada, CAD açısından tüm gerekli geometrik şekillerin ekranda gösterimi fare ile kullanıcı tarafından çizilebilmesi ve üzerinde değişiklik yapılabilmesi sağlanmıştır. Bunun yanı sıra, çizilen nesneler bir ağaç yapısında gösterilmiş ve gerektiğinde değişikliklerin bu ağaçtan yapılması

amaçlanmıştır. Ayrıca nesneler fare ile doğrudan çizilebildiği gibi komutlarla çizilebilmesi de amaçlanmıştır. Ayrıca dxf dosya formatındaki çizimlerin programa alınarak gösterimi, düzenlenmesi de sağlanmıştır. Gerekli klavye kısa yolları da eklenmiştir. Bu çalışma CAD programlarının çalışmasından yola çıkarak benzer özelliklere sahip özel amaçlı (sac kesme) olarak kullanılması hedeflenen bir yazılımın geliştirilmesini hedeflemiştir. Bu hedefin gerçekleşebilmesi için öncelikle CAD programlarında yer alan hemen hemen tüm işlemlerin programa dahil edilmesini gerekmektedir. Bu konuda kaynakların kısıtlı olması ve var olan kaynaklarında açık örnekler içermemesi nedeniyle her şeyin düşünülüp gerçekleştirilmesi zaman almıştır. CAD programlama açısından birçok soruya cevap bulunmuştur. Çizim işlemlerinin tamamlanmasının ardından yapılacak olan işlem kesilmesi için çizilen parçaların belirtilen boyutta bir tabaka üzerine yerleştirilmesinin sağlanmasıdır. Burada farklı parçaların da aynı tabaka içerisine eklenip hepsinin bir arada kesilmesi gerekebilir. Bu amaçla kütüphane yapısı geliştirilmiştir. Kullanıcı, kütüphaneden görerek istediği parçayı çift tıklayarak yeni bir pencerede açılmasını sağlayabilir, açılan yeni pencereye parçadan kaç adet yerleştirileceğini seçebilir ve tabaka üzerine yerleştirilmesini sağlayabilmektedir.

Nesting yerleştirmek anlamına gelmektedir. Nesting, istenilen şekil ve boyuttaki parçaların tabaka halindeki bütün bir saca en az malzeme telefi olacak şekilde yerleştirilmesinin sağlanmasını amaçlamaktadır. Parçalar döndürülerek ve yerleri değiştirilerek en uygun şekilde yerleştirilmelidir. Yerleştirme yapılırken kesmeyi sağlayacak olan sayısal tezgâhın kesme hareketini en kolay ve hızlı şekilde yapabilmesi de amaçlanmalıdır. Yerleştirmenin ardından geçilecek cam-üretim aşamasında yapılan yerleştirmeye göre kesme kafasının izleyeceği yol simüle edilerek kullanıcı tarafından izlenmesi ve kesmeye ilişkin olası hataların düzeltilmesi de sağlanmalıdır. Yerleştirme işleminin başarılı şekilde yapılabilmesi için başarılı bir algoritma geliştirilmeli veya var olan algoritmalar kullanılmalıdır. Bu işlem oldukça karmaşık ve zor bir metot içerir. Ayrıca yerleştirme işleminde tabaka halindeki sacın özellikleri ve çeşidi de yerleştirme stratejisini etkileyecektir. Kesme işlemini yapacak olan sayısal tezgâha ait parametreler de yerleştirme işlemi için bir temel veridir.

CAM (Computer Aided Manufactoring-BDT Bilgisayar Destekli Tasarım) üretim yapan tezgahların doğrudan bilgisayara bağlanabilmesi ve bilgisayar ile kontrol edilebilmesidir [2]. Ayrıca CAM içerisinde üretilecek mamulün tezgâha aktarılmadan işleme aşamasındaki adımlarına tek tek bakılabilir ve bir animasyon ile işlemler gösterilebilir. Böylece malzeme kaybı olmadan çok az bir zamanda üretim aşamalarına bakılarak olası hatalar veya yanlış planlanan işlemler düzeltilebilir. Bu çalışmada hedeflenen işlemlerden biride çizilen nesnelerin gerekli yerleştirme işlemleri yapıldıktan sonra plazma kesme tezgâhlarına aktarılabilecek veri biçiminde dosyalara dönüşümün sağlanması ve son aşama olan üretime geçilmeden önce tezgâhın çalışmasının simüle edilmesi ile gerekli hata kontrollerinin yapılması amaçlanmıştır. En hızlı ve doğru şekilde kesimin yapılabilmesi için kesme kafasının hareketleri(yol) ekrandan izlenerek, düzenlemelerin yapılması amaçlanmıştır. Bu kısım ile ilgili sade bir modül eklenerek nasıl olacağı konusunda bir ön fikir verilmiştir.

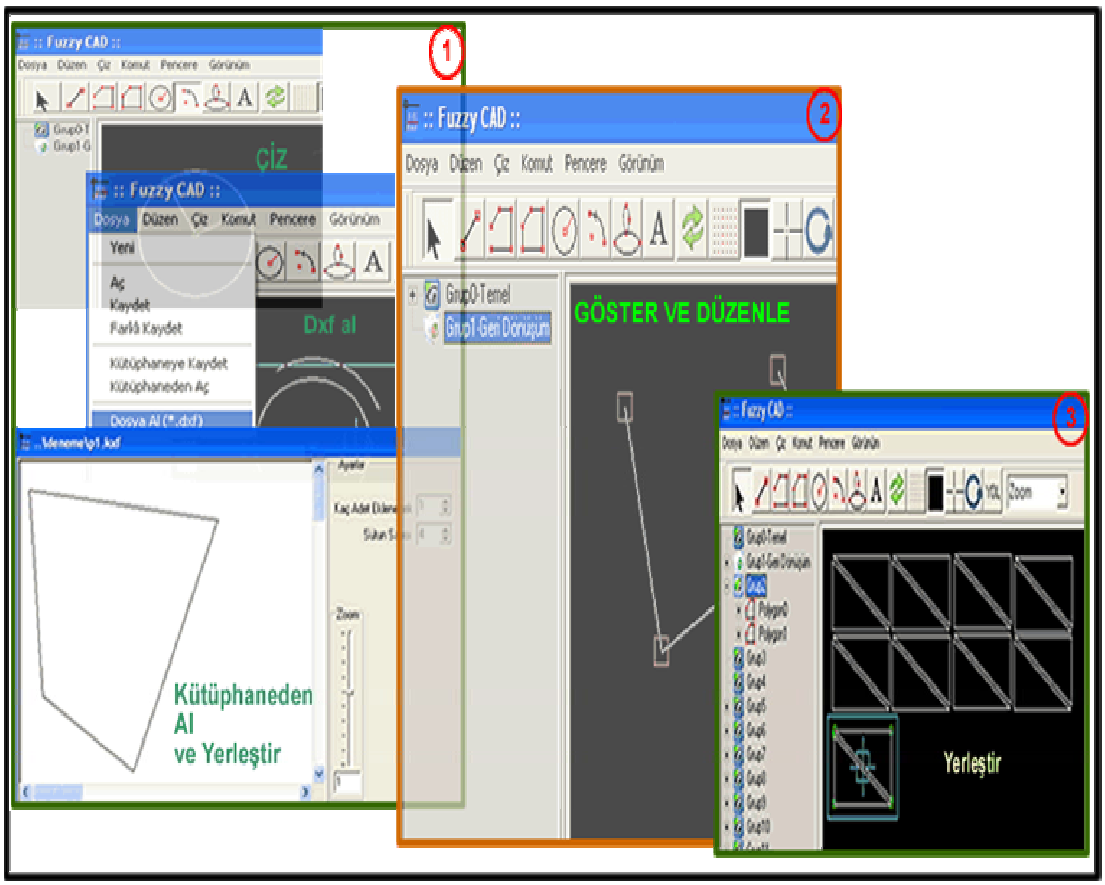
Nümerik kontrollü (NC ve CNC) takım tezgâhlarının endüstride kullanılmasıyla üretimde daha fazla hassasiyet ve esneklik sağlanabilmiştir. Ayrıca daha karışık ve hassas parçaların standart ve seri olarak üretilebilmesi sağlanmıştır.

Günümüzde kaliteli ve verimli üretim; Bilgisayar Destekli Tasarım (CAD); Bilgisayar Destekli İmalat (CAM) ve Bilgisayarlı Nümerik Kontrol (CNC) uygulamalarının birlikte uyum içinde kullanılması ile sağlanmıştır. Günümüzde CAD ve CAM yazılımlarının gelişmeleri ve yaygınlaşmalarından dolayı NC ve CNC tezgâhları daha verimli kullanılır hale gelmiştir.

CNC tezgâhlarında imal edilen parçaların çoğu karışıktır ve 3 eksen operasyonlarına ihtiyaç duyarlar. Böyle parçaların programını manüel olarak yazmak ve CNC ye yüklemek zor ve zaman alıcıdır. Hata yapma olasılığı hayli fazladır. Böyle durumlarda CAM yazılımları sayesine yüksek hassasiyette programlar oluşturmak, değişiklik yapmak ve farklı işleme operasyonlarını CAM üzerinde simüle ederek üretime başlamak zaman ve mükemmellik açısından gayet sağlıklıdır.

Hassas ve verimli üretim için CAD ve CAM yazılımları gerekli olduğu kadar NC programlamanın temelleri olan "G ve M " yani ISO Standart programlama kodlarının bilinmesi gereklidir.

Bu çalışmanın amacı Şekil 1.1'deki gibi özetlenebilir.



Şekil 1.1 Projenin Amacı

2. KAYNAK ARAŞTIRMASI

Sac metal ürünlerin imalatında, işlemlerin planlanmasından ürünün imal edilmesi aşamasına kadar farklı bilgisayar destekli (BD-CA-Computer, Aided) yazılım araçları kullanılır. Bilgisayar Destekli İşlem Planlama (CAPP-Computer Aide, Proces, Planning) sistemleri; en iyi araç yollarını, planlarını ve bilgisayar destekli üretim (BDÜ-CAM-Compute, Aide, Manufacturing) sistemleri için çeşitli imalat gereksinimlerini (delme veya kesme gibi) oluşturmalıdır. Kaynak araştırmalarımıza göre, sac metal ürünlerin geliştirilmesi işlemlerinde CAPP, CAM ve bilgisayarlı nümerik kontrol (CNC-Compute, Numerica, Control) sistemlerini etkin bir şekilde bir arada kullanabilen platform veya sistem yoktur. Her ne kadar bu işlemin nasıl uygulanacağı ISO 10303[1] standardının 207. bölümünde[2] açıklanmış olsa da uygulamada sınırlı kalmıştır (XIE,XU 2006).

Farklı programlar arasında çizim dosyalarının aktarılabilmesi amacıyla nötr dosya formatları geliştirilmiştir. Bu dosya yapıları IGES, VDAIS, VDAFS, SET ve DXF'dir.

IGES, 1979 yılında CAD veri değişimi için ilk uluslararası standart olarak ortaya konulmuştur. 10 ekim 1979 yılında da 200 kişinin katılımıyla yapılan açık bir toplantıda IGES (Initial Graphics Exchange Specifications) duyuruldu (Roger N. Nagel, Walt W. Braithwaite and Philip R. Kennicott). Daha sonra IGES v1.0 1981 yılında ANSI standardı olarak kabul edildi ve duyuruldu (Kemmerer 1999). IGES dosyası altı bölümden oluşur. Flag, Start, Global, Directory Entry, Parameter data ve Terminate. Nesneler directory entry ve parameter data entry kısmında tanımlanmıştır. Directory entry index veriyi tanımlayan öznitelikleri içerir. Parameter data kısmında tüm nesnelerin tanımının yapıldığı sabit uzunlukta kayıtlar içerir.

---

[1] ISO 10303 Endüstriyel otomasyon sistemleri ve bütünleştirme(STEP Standard for the Exchange of Product model data)) – Ürün verisi gösterimi ve değişimi standardıdır.

[2] ISO 10303 Bölüm 207 04.06.2004 tarihinde yürürlüğe giren, sac metal kalıplarının planlanmasında ve tasarımında kullanılacak uygulama protokolüdür.

VDAIS (Vereinung Deutsche Automobilindustrie IGES Subset) IGES'in Alman otomobil endüstrisinde tanımlanmış bir alt kümesidir. Hala Alman otomotiv endüstrisinde kullanılmaktadır.

VDAFS (Vereinung Deutsche Automobilindustrie Flächen Schnittstelle) yüzey geometrisinde veri değişimi için Alman nötr dosya yapısıdır. Çok karmaşık bir yapıya sahiptir (http://cadd.web.cern.ch).

SET (Standard Exchange et de Transfer) veri transferi ve arşivlemesi için oluşturulmuş Fransız standardıdır. 1983 yılında farklı CAD sistemleri arasında veri değişimi için nötr dosya yapısı olarak geliştirilmiştir (http://cadd.web.cern.ch).

Dxf dosya yapısı Autocad programını yazan Autodesk firması tarafından 1992 yılında oluşturulmuştur. DXF dosyası kod ve değerlerden oluşur. Kodlar grup kod olarak bilinir. Grup kodlar tamsayı tipindeki sayılardır. Bu sayılar takip eden verinin neyi ifade ettiğini gösterir. DXF dosyalarında her bölüm grup kod 0 (sıfır) ile başlar ve SECTION yazısı takip eder. Bunu grup kod 2 takip eder ve bölüm adını gösteren yazı takip eder (Örneğin, HEADER). Sonraki bölümde tanımlanmış elemanların grup kodları ve değerleri yer alır. Bölümün (Section) bitişi 0 (sıfır) ve onu takip eden ENDSEC metni ile olur (www.autodesk.com).

Yukarıda kısaca tanıtılan dosya türleri cad programları arasındaki veri değişimi ve cad programlarından cam programlarına tasarımların aktarılmasını sağlar. Bir cad-cam programının bu dosya formatlarından bir ya da birkaçını tanıması gereklidir. Nötr dosya formatlarının kullanılması bu çalışmada amaçlanan cad-nesting-cam bütünleşmesi için bulunması gereken özelliklerden biridir. Sonraki adım çizilen nesnelerin tabaka saclara en uygun şekilde yerleştirilmesidir. En uygun yerleşim için ölçütler şunlar olabilir; en az malzeme sarfiyatı, en hızlı ve doğru kesim yolu, kısa sürede yerleştirme vb. Yerleşim algoritmaları geleneksel programlama metotlarıyla düşünüldüğünde oldukça zor ve fazla hesaplama gerektiren işlemleri içermektedir. Bu işlem çok zaman almaktadır. Bu nedenle yerleştirme için değişik algoritmalar geliştirilmiştir.

Fujita ve ark. En uygun yerleştirme için kalıtım algoritması (GA-Genetic Algorithm) ve küçültme algoritmasını birleştirme yaklaşımını sunmuştur. Su ve ark. en uygun yerleştirme yöntemini kalıtım algoritmasını kullanarak geliştirmiştir.

İsteğe göre çizilen poligonların yerleştirilmesi problemleriyle ilgili olarak Li ve ark. kalıtımsal algoritma ve bulanık mantık kurallarını kullanarak karma bir metot sunmuştur. Kalıtımsal algoritmayı yerleştirme ve dönme sınırları stratejisini anlamak için kullanmışlardır. Bulanık mantık kuralları, kalıtımsal algoritma içindeki nesnelerin sıklığında kullanılacak meyil algoritması için uygun işlev değerlerini tek tek bulmakta kullanılan bir fonksiyon gibidir. Bu birleşim kalıtım algoritmasındaki arama kapasitesini arttırarak en uygun çözümün bulunmasını olasılığını yükseltmiştir.

İki boyutlu parçaların büyük saclardan en az malzeme sarfiyatı olacak şekilde kesilmesi birçok endüstride önemli bir iştir. Yerleştirme yöntemi birçok parametreyle ilişkilidir. Bunlar kesme işlemi, şekil ve farklı parçaların kesimi, sacın şekli ve boyutu ve sonraki işlemler için gerekli parçalar vb. Bundan dolayı en az malzeme sarfiyatı olacak şekilde farklı parçaların en uygun şekilde yerleşiminin bulunması önemli çaba gerektirir. Yerleştirme problemlerinin çözümünde farklı teknikler kullanılır. Matematiksel teknikler, probleme özgü buluşsal yöntemler ve yeni buluşsal yöntemler gibi. Çeşitli matematiksel teknikler; tamsayı programlama, doğrusal (lineer) programlama, devingen (dinamik) programlama ve tamsayı lineer programlama vb. metotları dikdörtgen biçimindeki parçaların dikdörtgen biçimindeki tabakalara yerleştirilmesinde kullanıldı. Dikdörtgen biçimindeki tabakalara dikdörtgen biçimindeki şekillerin yerleştirilmesinde dinamik programlamaya başvuran Beasley (1985)'dir. Bu metot, dikdörtgen biçimindeki parçaların hesaplamaya dayalı olarak yerleştirilmesinde tabakalar boyutu çok büyük olduğu zaman kullanışlı değildir. Reda ve Abd (1994) lineer programlamayı dörtgensel parçaların giyotin ile kesmede yerleştirme amacıyla kullanmışlardır. Tamsayı lineer programlama metodu çok sayıda dörtgenler ve tabakalar düşünüldüğünde yerleştirme için elverişli değildir. Çok sayıda eşitsizlik ve ikili değişkenlerin yerleştirme problemlerinin çözümünde var olmasından dolayı (Shpitalni ve Manevich 1996)

matematiksel metotlarla çok sayıdaki parçaların çok yönlü tabakalara yerleştirilebilmesi sınırlıdır.

Babu A. ve Babu N. Genetik ve kalıtımsal algoritmayı kullanıp karma bir algoritma geliştirerek uygulamışlardır. Yapılan çalışma sonucunda değişik dörtgensel parçaların çoklu dörtgensel tabakalara yerleştirilmesinde etkin olarak kullanılabileceğini ifade etmişlerdir.

BDT/BDÜ sistemleri arasındaki iletişimi sağlamak için bu güne kadar çeşitli Bilgisayar Destekli İşlem Planlaması sistemleri geliştirilmiştir. Ancak bunlar imalatçıları ve imalat gereksinimlerini tam olarak karşılayacak seviyeye ulaşamamıştır. Farklı bdt sistemleri arasındaki iletişim sorunlarını gidermek amacıyla DXF, IGES, VDAFS, STEP gibi çeşitli standartlar geliştirilmiştir. Bu çalışmada "Autocad'de parça resimleri çizilmiştir. BDT/BDÜ bütünleşmesini sağlamak için çizimler DXF formatına dönüştürülmüştür. TOR-CAPP adı verilen programla torna tezgâhında işlenecek parçaların simülasyonu ve parça işlem planı üretilmiştir. Daha sonra Sd kod dönüşümü yapılarak parça programları üretilmiştir. Bu işlemleri gerçekleştiren bilgisayar programı Qbasic dilinde yazılmıştır." (Çalik, Ünüvar ve Yaldız 1998)

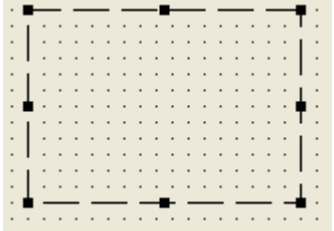
Yerleştirme konusunda yapılan çalışmalardan ve kullanılan yöntemlerden bazıları özetlenmiştir. Tasarımdan üretime kadar yapılması gereken bir diğer işlemde yerleştirmeden sonra yolların simüle edilerek gösterilmesidir. Bu simülasyonda kesme ucunun izleyeceği yol, parçanın kesmeye başlanacağı nokta ve kesim giriş yerleri gösterilir. Bu simülasyona bakıp, operatör olası hataları tespit eder düzeltilmesini sağlar. Yerleştirme ve yolların belirlenmesinden sonra kesme işini yapacak olan tezgâhın özelliklerine göre en son elde edilen çizimin CNC'nin kullanacağı kodlara dönüştürülmesi gerekir.

3. MATERYAL VE METOT

Eskiden programlama dilleri içerisinde yer alan grafik mod önce MAC'lerde daha sonra Windows'la birilikte kişisel bilgisayarlarda bir işletim sistemi ortamı haline gelmiştir. Windows'la birlikte bu işletim sisteminin getirdiği yenilikleri kullanabilen programlama dilleri görsel haliyle ortaya çıkmıştır. Bugün hayatımıza giren ve kolaylaştıran birçok program ve teknolojinin ortaya çıkması da bu dönemde olmuştur. Bu çalışmada Windows işletim sisteminde çalışacak bütünleşik metal sac kesiminde kullanılacak cad-nesting-cam içeren bir program yazılmıştır. Program Borland Delphi 7.0 ile yazılmıştır. Delphi de yer alan nesneler dışında hazır bir nesne ya da bileşen kullanılmamıştır. Hazır bileşen kullanılmaması konunun daha iyi anlaşılmasını sağlamış ve daha sonra istenildiği gibi gelişime açık bir yapı ortaya konması için çok yaralı olmuştur. Programda yer alan önemli nesneler ve bu nesnelerde kullanılan metotlardan bazıları şunlardır.

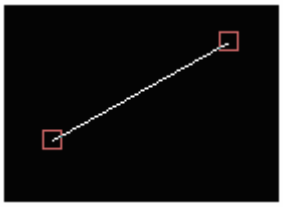
3.1. Çizim Alanı

Programda kullanılan çizim alanı Delphi' de Additional sekmesinde yer alan Timage nesnesidir. Timage nesnesi grafiksel görüntüleri form içerisinde göstermek için kullanılır. Timage nesnesinin canvas özelliği kullanılarak üzerine geometrik şekiller çizilebilir. Canvas özelliği Delphi' de form, shape, bevel gibi bazı nesnelerde de vardır. Ancak incelemeler sonucunda çizim alanı olarak bunlardan ya doğrudan form alanının kullanılması ya da Timage nesnesinin kullanılması en uygun olduğu görülmüştür. Bir forma eklenen Timage nesnesi Şekil 3.1'deki gibi görünür.

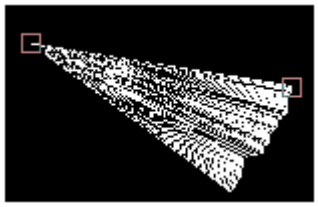


Şekil 3.1 Timage nesnesi

Çizimlerin fare ile yapılabilmesi için Timage nesnesinin Mouse.down, Mouse.move ve Mouse.up olayları kullanılmıştır. Çizim alanında fare ile çizim yapılırken önceki çizilen şeklin tekrar çizilmesi ile eski resmin gösterildiği pikseller tekrar taranmış böylece şekil silinmiştir. Şekil 3.2'de bu işlem bir çizginin boyutlandırılmasında şöyle gösterilebilir. Şekil 3.2.a'da taşıma esnasında farenin hareketine göre eski şekil silinmiş yenisi çizilmiştir. Şekil 3.2.b'de ise eski şekil silinmemiştir.



a. Tarama yapıldığında



b. Tarama kaldırıldığında

Şekil 3.2 Tarama sureci

3.2. Nesnelerin Hafızada Tutulması

Çizim nesnelerinin tanımlanmasında kayıt veri tipi kullanılmıştır. Nesnelerin çizim metotlarına göre gerekli koordinat ve diğer bilgileri bu kayıtta tanımlanmıştır. Nesneler teknik resimdeki metotlara yakın olarak analitik geometride olduğu gibi koordinatlarla ifade edilmiştir. Örneğin bir çizgi yi incelersek çizgi değişik şekillerde analitik olarak ifade edilebilir. Başlangıç-bitiş koordinatlarıyla, başlangıç koordinatları-eğim-uzunluk ile vb. Çizgi hangi metotla çizilirse çizilsin hafızada saklanması tek bir forma hesaplamalar ile sokulabilir veya her yöntem için ayrı bir kayıt tanımlanabilir. Programda çizgi aşağıdaki gibi temsil edilmiştir.

```
type
   ddline=record
         fp:TPoint;
         sp:TPoint;
         egimradyan:Real;
         secili:Boolean;
         silindi:boolean;
   end;
var
  mcizgi:array[0..100] of ddline;
```

Çizginin çizilmesinde ve hafızada ifade edilmesinde başlangıç-bitiş koordinatları kullanılmıştır. Ekranda nesneler piksel olarak gösterilmektedir bu nedenle de başlangıç ve bitiş noktaları Tpoint tipinde tanımlanmıştır. Tpoint değişkeni iki tamsayı değerinden oluşur. Tpoint tipinde tanımlanan değişkene veri şu şekilde aktarılabilir;

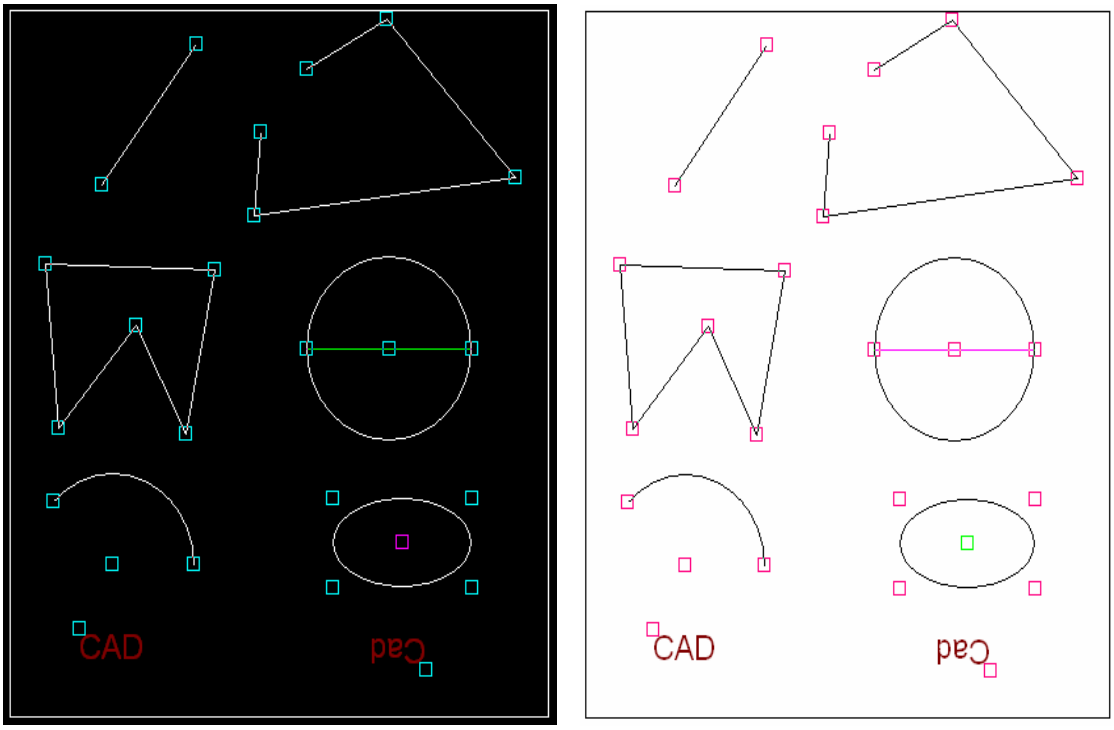
```
Procedure kaydet(x,y:Integer);
Var
   Konum:Tpoint;
   T1,T2:Integer;
Begin
      Konum:=point(x,y);//veya  konum.x:=x; konum.y:=y; // değer aktarma.
      T1:=konum.x; T2:=konum.y; //Değer alma
End;
```

Yukarıdaki program parçasında konum değişkenine x,y koordinatlarının değerleri aktarılmıştır. Aktarılan değerler daha sonra tamsayı tipinde tanımlanmış olan T1,T2 değişkenlerine aktarılmıştır.

Çizgi kaydında gerekli koordinatların yanında onun bir nesne gibi kullanılabilmesi amacıyla seçili ve silindi gibi mantıksal değişkenlerde tanımlanmıştır. Seçili değişkeni bir çizginin seçili olup olmadığını anlamamızı sağlamaktadır. Silindi değişkeni çizginin silinip silinmediğini göstermekte kullanılır. Silinen nesneler programdan tamamen atılmamakta geri dönüşüm kutusunda tutulmaktadır.

Diğer şekillerde gerekli verileri içeren kayıtlar olarak tanımlanmıştır. Daire merkez koordinatları-yarıçap, yay merkez koordinatları-yarıçap-başlangıç açısı-yay açısı, poligon ve polyline noktalarının koordinatları, elips çerçeveleyen dörtgenin sol-üst ve sağ-alt koordinatları, yazı başlangıç koordinatları şekilde tanımlanmıştır.

Şekil 3.3'de şekiller çizilerek seçilmiştir. Tutma noktaları şekillerin hangi bilgilerle saklandığı hakkında bilgi verebilir. Şekil 3.3'de çizilen şekiller ve bunların tutma noktaları gösterilmiştir. Şekil 3.3.a'da zemin siyah olarak çizimler gösterilmiştir. Şekil 3.3.b'de ise renkler ters çevrilerek aynı şekiller ve tutma noktaları gösterilmiştir.



a. Siyah ekranda b. Renkler ters çevrildiğinde

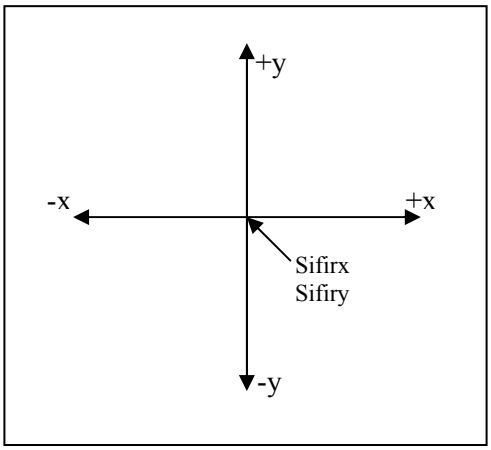
Şekil 3.3 Şekiller ve tutma noktaları

3.3. Koordinat Sistemi

Görsel programlarda sıfır noktası sol-üst köşedir. Koordinatlar Windows'ta SetViewportOrgEx API ile değiştirilebilir. Ancak fare ile doğrudan çizim yapılırken yapılan koordinat değişikliği etki etmez yine sıfır noktası aynı kalır. Bu API ancak dosyadan okuma işlemlerinde etkin olarak kullanılabilir. Bu durumda çizim esnasında değişiklik şu şekilde yapılabilir. Farenin hareketlerine göre meydana gelen olaylardan biri, mousemove olayıdır. Bu olay gerçekleştiğinde farenin değerleri de x,y değişkenlerinde prosedüre aktarılır. Alınan bu değerler global olarak tanımlanan sifirx ve sifiry değişkenleri ile aşağıdaki gibi işlem yapıldığında x,y değerleri sıfır noktasına göre hesaplanmış olur. Böylece koordinat sistemi ve bölgelerde analitik geometride olduğu gibi ayarlanmış olur. Şekil 3.4'de alınan x,y değerlerinin sıfır noktasına göre hesaplanmasından sonra oluşacak düzlem gösterilmiştir.

```
procedure TForm1.c_alanMouseMove(Sender: TObject; Shift: TShiftState; X,Y:
Integer);

Begin
        x:=x-sifirx;
        y:=sifiry-y;
        …….;
        …….;
 End;
```



Şekil 3.4 Koordinat sistemi

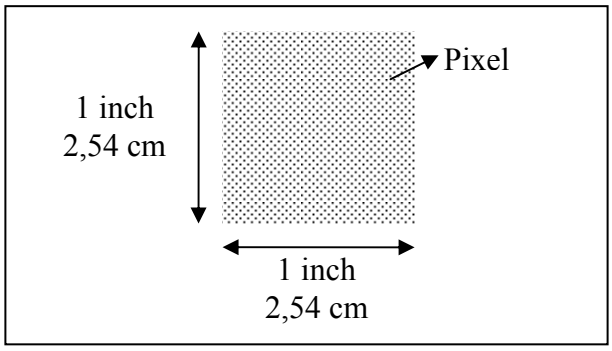
3.4. Ölçülendirme

Ekranda gösterilen en küçük nokta pikseldir. Onun dışında gösterilen değerler bir takım kabul edişlere dayanarak yapılan hesaplamalardır. Yani 1 piksel 0.1 mm olarak da 1 piksel 100 mm olarak da kabul edilebilir. 1 pikselin kaç cm veya mm olacağı Windows'un SetMapMode API kullanılarak değiştirilebilir. Ancak bu değişim bir farz ediş olduğu için SetViewportOrgEx komutunda olduğu gibi çizilmiş nesnelerin açılmasında işe yarar ve fare ile yapılan doğrudan çizimlerde gösterim piksel olmasına rağmen kullanıcıya pikselden dönüştürülen uzunluk değeri gösterilerek gerçek uzunluklarla çalışması sağlanır. Bu API parametre olarak şu değerleri alabilir (http://msdn.microsoft.com).

Tablo 3.1. SetMapMode komutunun parametreleri.

Değer	**Açıklama**
MM_ANISOTROPIC	Grafik arabirimine göre gösterim yapılır.
MM_HIENGLISH	Her mantıksal birim 0.001 inch olarak gösterilir. Pozitif *x* is sağa doğru; pozitif *y* üste doğru olur.
MM_HIMETRIC	Her mantıksal birim 0.01 mm olarak gösterilir. Pozitif *x* is sağa doğru; pozitif *y* üste doğru olur.
MM_ISOTROPIC	Her mantıksal birim varsayılan(piksel) uzunlukta olur.
MM_LOENGLISH	Her mantıksal birim 0.01 inch olarak gösterilir. Positif *x* is sağa doğru; pozitif *y* üste doğru olur.
MM_LOMETRIC	Her bir piksel 0.1 mm olarak gösterilir. Pozitif *x* is sağa doğru; pozitif *y* üste doğru olur.
MM_TEXT	Piksel olarak gösterilir. Varsayılan olandır. Pozitif *x* is sağa doğru; pozitif *y* alta doğru olur.
MM_TWIPS	Her mantıksal birim yirmidebir oranında gösterilir yazıcılardaki gibi (1/1440 inch, twip olarak ifade edilir). Pozitif *x* is sağa doğru; pozitif *y* üste doğru olur.

Görsel programlamada ekran çözünürlüğüne göre her formun PixelPerInch olarak belirlenen bir özelliği vardır. Bu özellik bir inch karelik alanın kaç noktadan oluştuğunu belirtir. Aşağıdaki şekilde bu tanımlama gösterilmiştir.



Şekil 3.5 Ölçülendirme

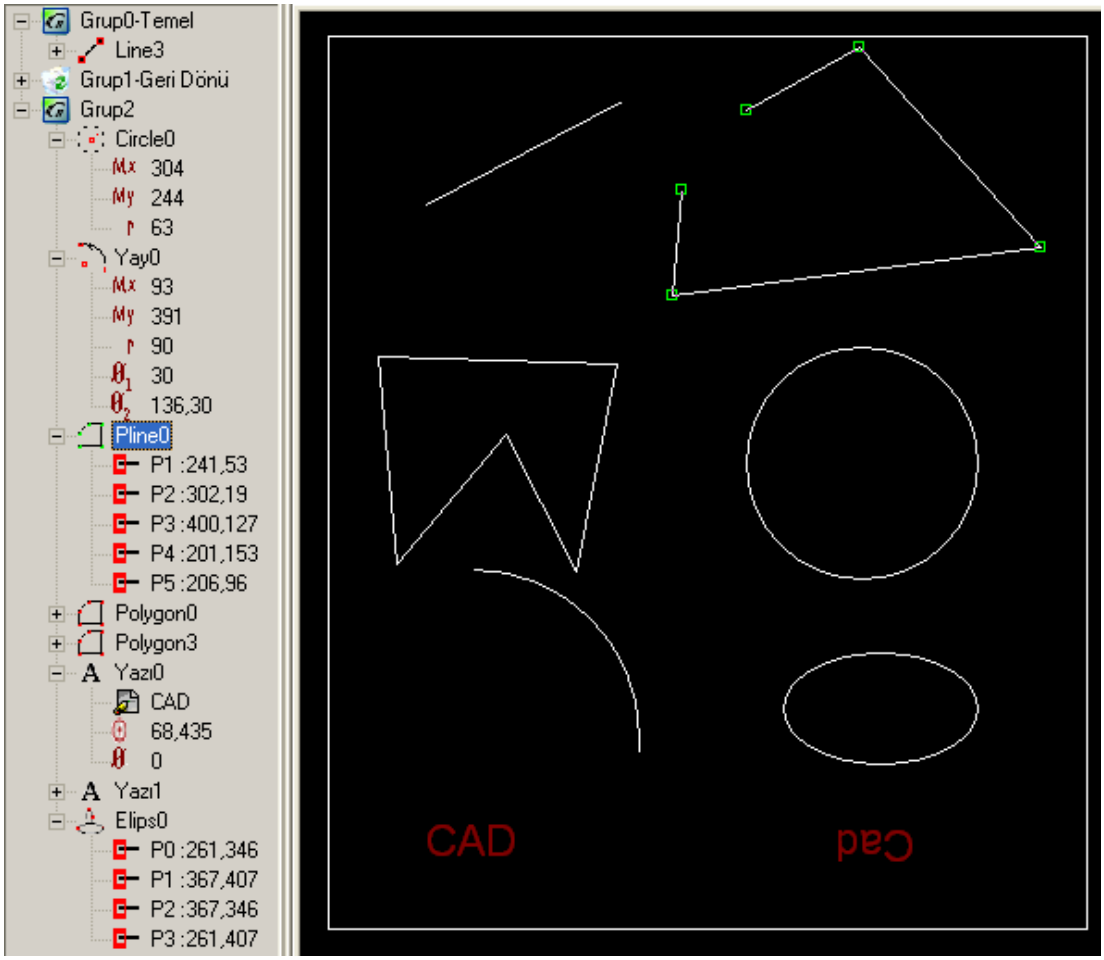
Bilgisayar destekli çizim programlarında ölçülendirmelerin günlük hayatta kullanılan uzunluklara göre yapılması gerekir. Bilgisayardaki çizimlerin gerçek uzunluklara dönüştürülmesi pixelperinch parametresi kullanılarak yapılabilir. Dönüşüm işleminde kullanılan kodların pikselden cm ye dönüşümü aşağıdaki gibidir.

```
FloattoStrF(((x/form1.PixelsPerInch)*25.4),ffNumber,10,7)
+ ' ||| ' +
FloattoStrF(((y/form1.PixelsPerInch)*25.4),ffnumber,10,7);
```

Programda farenin çizim alanındaki konumu durum çubuğunda gösterilmektedir. Konum piksel ve gerçek ölçülerle olmak üzere iki şekilde yapılmıştır. Gösterilecek gerçek ölçek Uygulama Tercihleri penceresindeki Birimler sekmesinden yapılmaktadır. Buradan ondalık basamak sayısı da değiştirilebilmektedir. Varsayılan pixelperinch değeri 96' dır. Yani 1 inclik uzunluk 96 noktadan oluşmaktadır. Şekil 3.5'de bu durum gösterilmiştir. 1 inch 25,4 mm olduğuna göre 1 piksel 0,264583 mm olarak bulunur. Bu değer seçilen hassasiyete göre yuvarlanarak ondalık basamak sayısı azaltılır. Bu da bize çizimlerdeki hassasiyetin bulunmasını sağlar.

3.5. Ağaç Yapısı

Çizilen şekiller bir gruba dâhil edilerek gösterilmiştir. Yani her şekil bir gruba üyedir. Buradaki grup yapısı CAD programlarındaki katmanların temsilidir. Katman yapısında her katmanın, o katmana ait olan tüm nesneleri etkileyen bir takım özellikleri vardır. Gizleme, kilitleme, çizgi biçimlerini değiştirmek vb. bu çalışmada gruplar için bu özellikler tanımlanmamıştır. Gruplar, kendisine ait olan nesnelerin döndürülmesinde çoğaltılmasında kullanılmıştır. Böylece bir nesneyi oluşturan çizimler bütün halinde bağımsız olarak seçilebilmekte ve işlem yapılabilmektedir. Programın geliştirilmesinde grup yapısı oldukça katkı sağlayacaktır. Program çalıştırıldığında otomatik olarak Grup 0- Temel ve Grup 1-Geri dönüşüm olmak üzere iki grup oluşturulmaktadır. Ağaç yapısının gösteriminde Delphi' deki Ttreeviev nesnesi kullanılmıştır. Şekil 3.3'deki nesnelerin ağaçta gösterimi Şekil 3.6'daki gibidir. Çizilen nesnelere ait özellikler fare ile çizim alanından değiştirilebildiği gibi treeviev üzerinden çift tıklanarak ya da sağ tuş menüsünden düzenle komutu seçilerek değiştirilebilir.

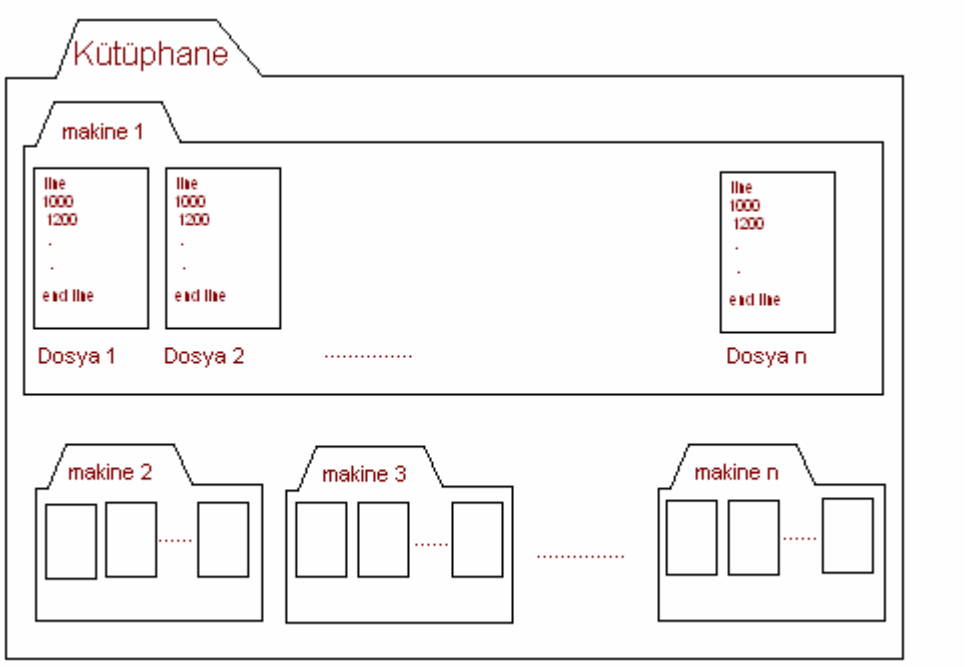


Şekil 3.6 Nesnelerin ağaç yapısında gösterimi

Şekil 3.6'dan da görüldüğü gibi Pline0 nesnesi ağaçtan seçildiğinde çizim alanında da seçilmesi sağlanmaktadır. Ağaç yapısın nesneler bir gruptan başka bir gruba taşınabilmektedir. Taşıma işlemi ctrl+shift+sol tuş birleşimi ile yapılmaktadır. Geri dönüşüm grubuna taşınan nesneler çizim alanından silinmektedir. Geri dönüşüm kutusunda yer alan nesneler diğer gruplara taşındığında da çizim alanında görüntülenmektedir. Ayrıca ağaç üzerinde sağ tuş menüsünden gruplar kopyalanıp yapıştırılabilmektedir.

3.6. Kütüphane Yapısı

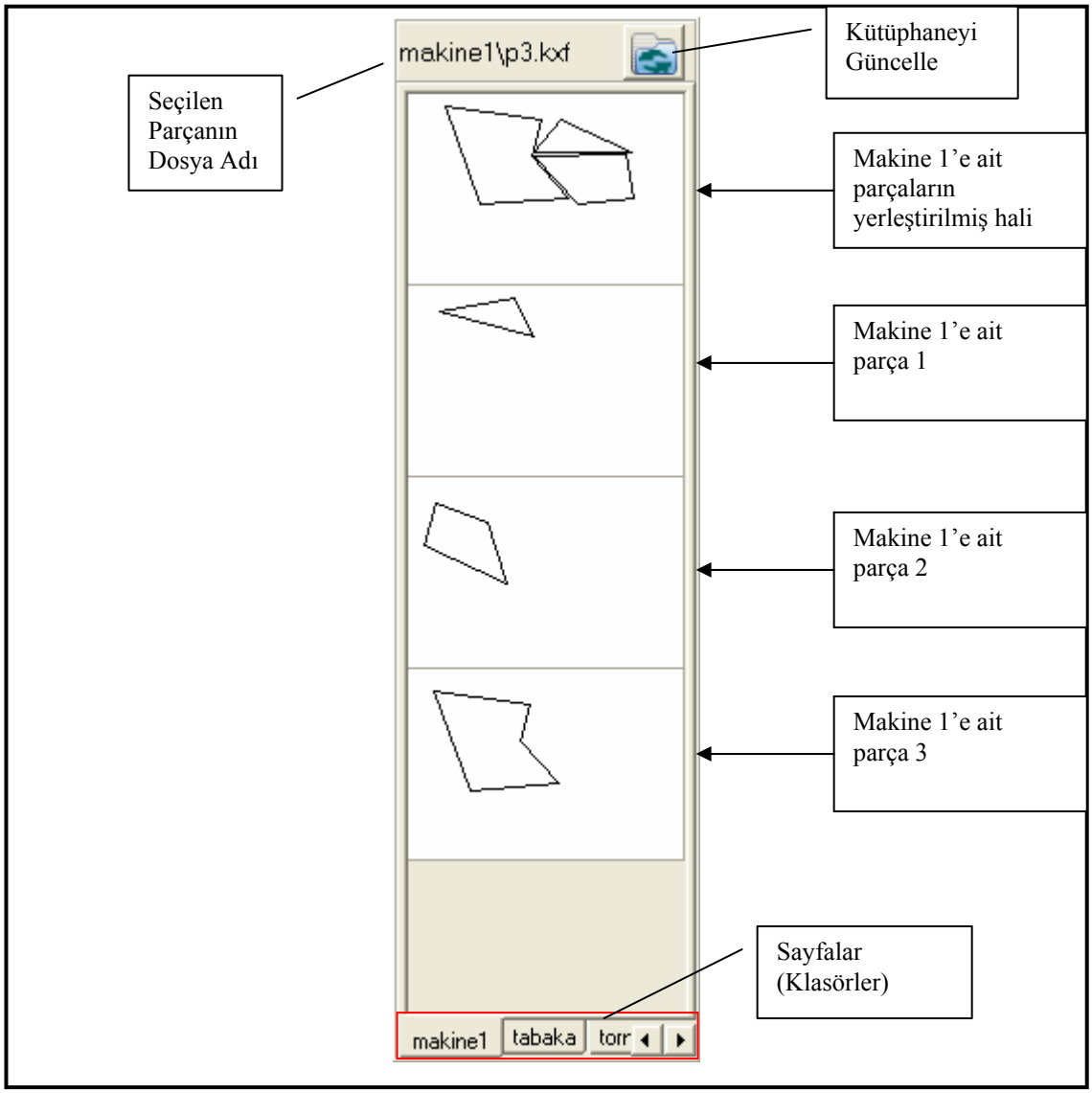
Kütüphane yapısında dosya sistemindeki ağaç yapısı kullanılmıştır. Bir makineyi ya da tezgâhı oluşturan parçalar kütüphane klasörü içine açılacak ilgili makineye ait klasörün içine bir dosya olarak kaydedilecektir. Şekil 3.7'de bu yapı şematik olarak gösterilmiştir. Kütüphane klasörünün programın yer aldığı klasörde oluşturulması gereklidir. Kütüphane klasörü içerisine de çizimi yapılan makineye ait klasör oluşturulmalıdır. Ancak bu yapı daha da iyileştirilerek kullanıcı tarafından istenilen düzenleme sağlanabilir. Bu düzenlemenin program yüklenirken ya da program içerisinden ayarlar menüsünden yapılması sağlanabilir.



Şekil 3.7 Kütüphane yapısının şeması

Bu yapıda bir makineyi oluşturan parçalar bir kez yerleştirilip ait olduğu makineye ait klasörün içine bir dosya olarak kaydedildiğinde o makineden kaç adet üretilecekse parça yerleşimi yapılan dosya tabaka saca yeniden yerleştirilerek kesime

aktarılabilir. Böylece yerleşim için harcanan zaman daha kısa olacaktır ve bir makinenin parçaları imalat için kusursuz hale geldiğinde en son yerleşim en doğru yapıyı içerdiğinden kusursuzluk devam edecektir. Kütüphanedeki klasör ve dosyalar Excel' deki gibi bir kitap yapısında sayfalar şeklinde gösterilmiştir. Kütüphane klasöründeki her klasör bir sayfayı bu klasördeki dosyalarda bir resim olarak ön izleme şeklinde açılmaktadır. Programda pencere menüsünden kütüphane seçeneği aktif yapıldığında kütüphane penceresi Şekil 3.8'deki gibi gösterilecektir.



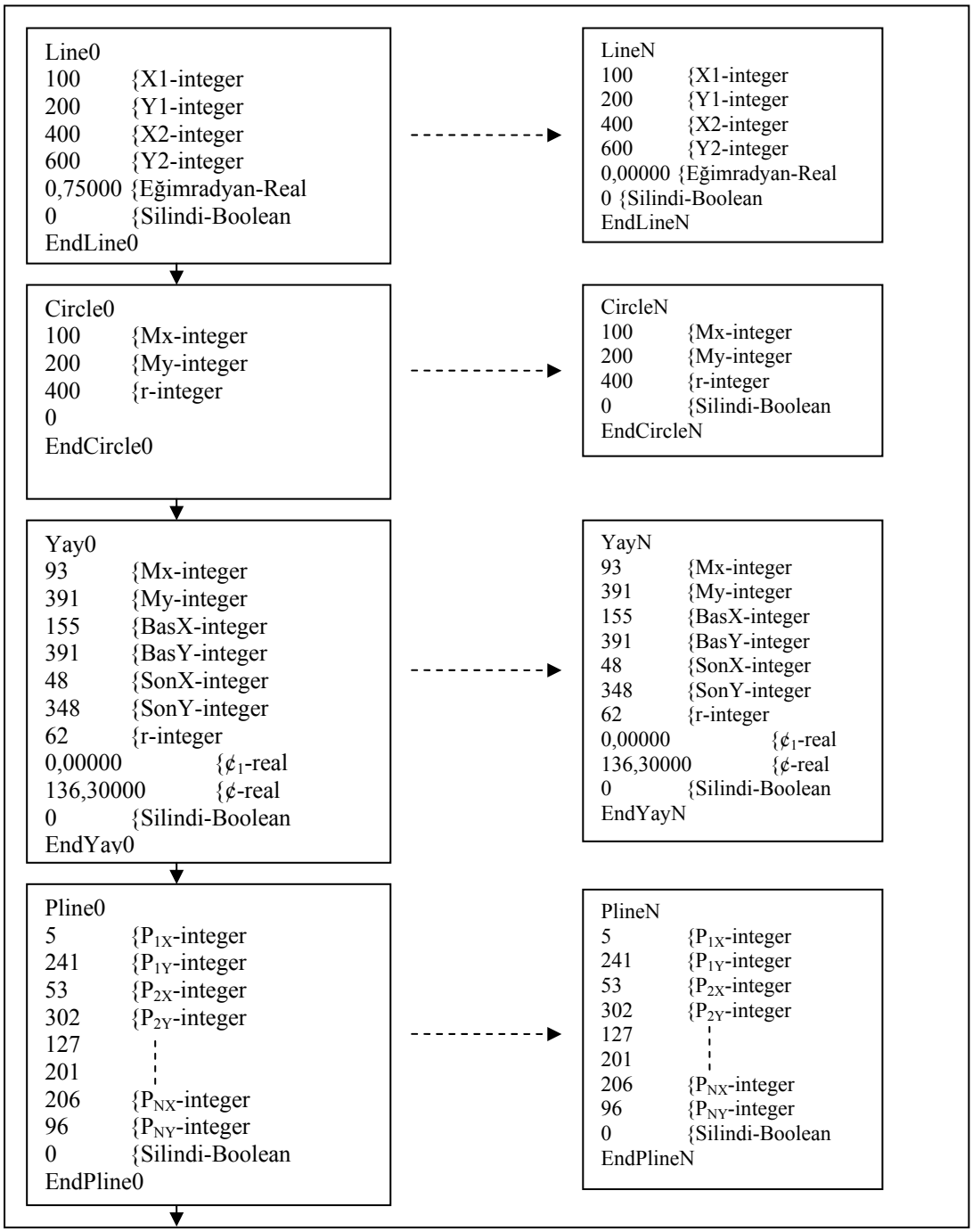
Şekil 3.8 Kütüphane penceresi

Kütüphane penceresinde bir şekil üzerine çift tıklandığında şekil yeni bir formda açılmaktadır. Bu formdan sayfaya aktar komutu seçilerek şekil istenilen sayıda sayfaya aktarılmaktadır.

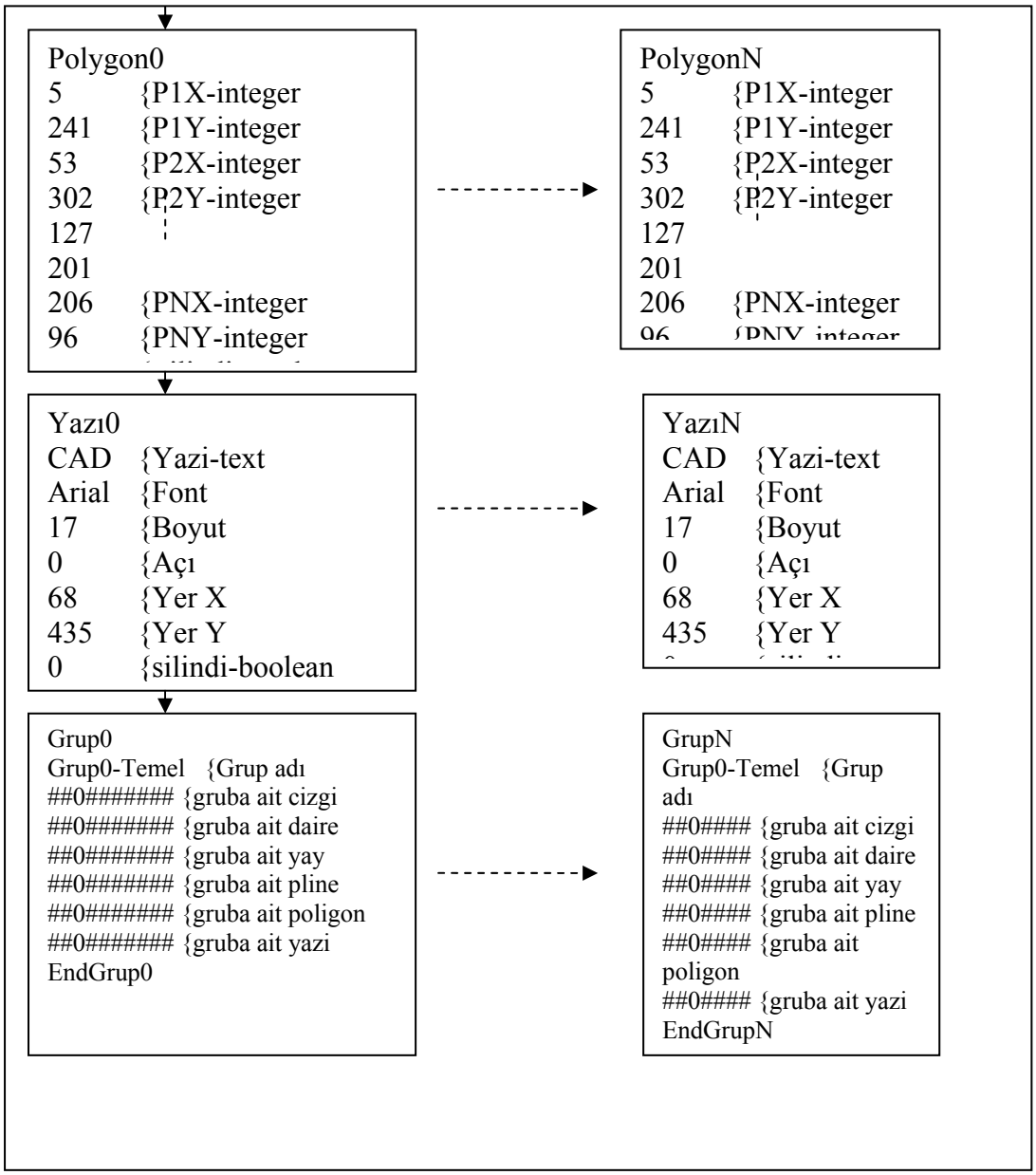
3.7. Dosya Yapısı

Bu çalışmada çizilen nesnelerin saklanmasında text türünde fxs uzantılı dosyalar kullanılmıştır. Bu dosyada çizilen şekillere ait gerekli bilgiler ve grup yapısı kaydedilmektedir. Nesneler isim ve numaraları ile birbirinden ayrılmaktadır. Örneğin line0'ın açılımı; line şeklin çizgi olduğunu, 0 ise çizginin numarasını göstermektedir. Dosyaya kaydedilen nesneler; ismiyle başlamakta ve end komutuyla sonlanmaktadır. Şekil 3.9 ve Şekil 3.10'da programda tanımlanan nesnelerin dosyada hangi bilgilerle saklandığı gösterilmiştir. Nesnelerden sonra grup bilgileri dosyaya kaydedilmektedir. Grup tanımlamasında o gruba ait olan nesneler numaraları ile her bir nesne üç karakter olacak şekilde kaydedilmiştir. Nesneleri numarası 0,1,2,…,n şeklinde gittiğinden her numara üç karaktere tamamlanmıştır. Örneğin line0 grup1'e ait ise grup1 dosyaya kaydedildiğinde line tanımında bu bilgi "##0" şeklinde ifade edilmiştir. Bu şekilde üç karaktere tamamlanmasının nedeni grup1'e ait çizgilerin tek bir satırda saklanmasıdır. Okunurken de üçer üçer gruplanarak # işaretleri kaldırılır ve metin tamsayı biçimine dönüştürülür. Şekil 3.10'da gruba ait bilgilerin saklanışı ve ne anlama geldikleri gösterilmiştir.

Kütüphaneye kaydedilen çizimler yine text türünde kxf uzantılı olarak kaydedilmektedir. Kütüphane dosyalarında grup yapıları kaydedilmemiştir çünkü çizilip kütüphaneye kaydedilen nesneler bir makineye ait parça olarak düşünülmüştür. Ayrıca kütüphaneden yerleşim yapıldığında veya yeni bir çizime eklendiğinde bir grup olarak eklenmektedir. Çoğaltma, döndürme gibi işlemler bu grup üzerinden yapılmaktadır.



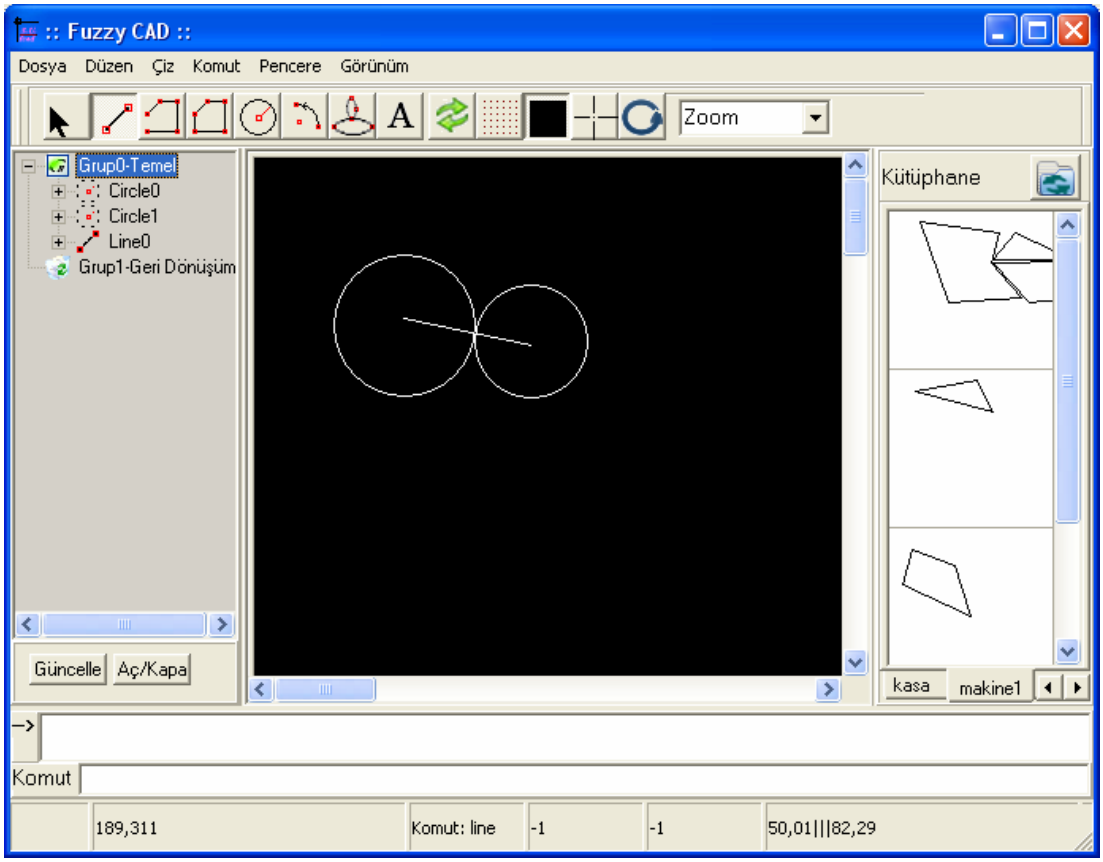
Şekil 3.9 Nesnelerin dosyada saklanışı (çizgi, daire, yay, pline)



Şekil 3.10 Nesnelerin dosyada saklanışı (poligon, yazı, grup)

4. GELİŞTİRİLEN TASARIM VE ÜRETİM SİSTEMİ PROGRAMI

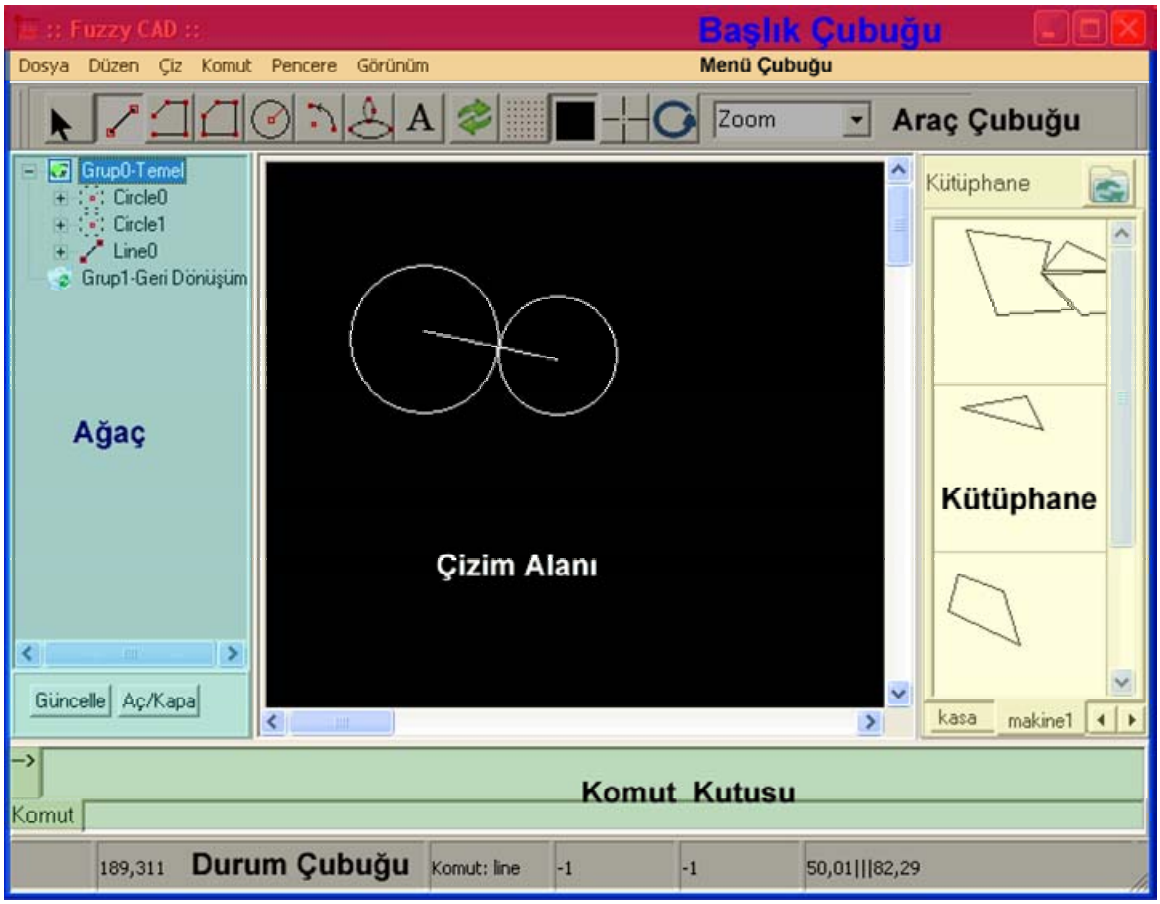
4.1. Ana Form Ekran Görünümü



Şekil 4.1 Ana form ekran görünümü

Programın ana form görünümü Şekil 4.1'deki gibidir. Program dört ana bölümden oluşmaktadır. Ortadaki kısım çizim alanıdır. Çizim alanı dışında kalan kısımlar pencere menüsünden gizlenebilir. Sol tarafta ağaç yapısı gösterilmekte çizilen nesneler buraya bir node olarak eklenmektedir. Çizilen nesnelere ait bilgiler bu node'a alt node olarak eklenmekte ve gösterilmektedir. Sağ taraftaki kısım kütüphaneye kaydedilmiş dosyaların gösterildiği kısımdır. Alt kısım kullanıcı ile

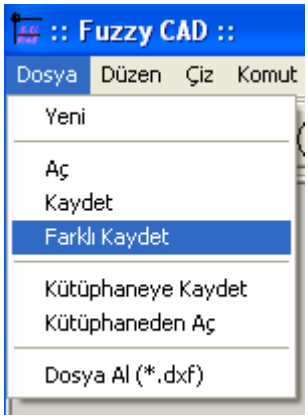
program arasındaki etkileşimi sağlamak için konulmuştur. Kullanıcının çizimleri fare yerine klavyeden yapabilmesi, gerektiğinde bazı değerleri girebilmesi ve programın vereceği mesajları görebilmesini sağlamaktadır. Şekil 4.2'de ana formda yer alan pencereler gösterilmiştir.



Şekil 4.2 Ana form pencereleri

4.2 Dosya Menüsü

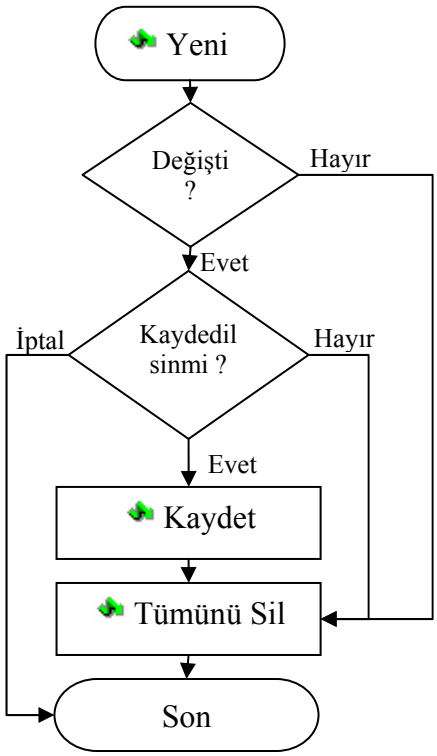
Dosya menüsü Şekil 4.3'de gösterilmiştir. Dosya menüsünde yeni, aç, kaydet, farklı kaydet, kütüphaneye kaydet, kütüphaneden aç ve dosya al(dxf) komutları bulunmaktadır.



Şekil 4.3 Dosya menüsü

4.2.1. Yeni komutu ve akış çizelgesi

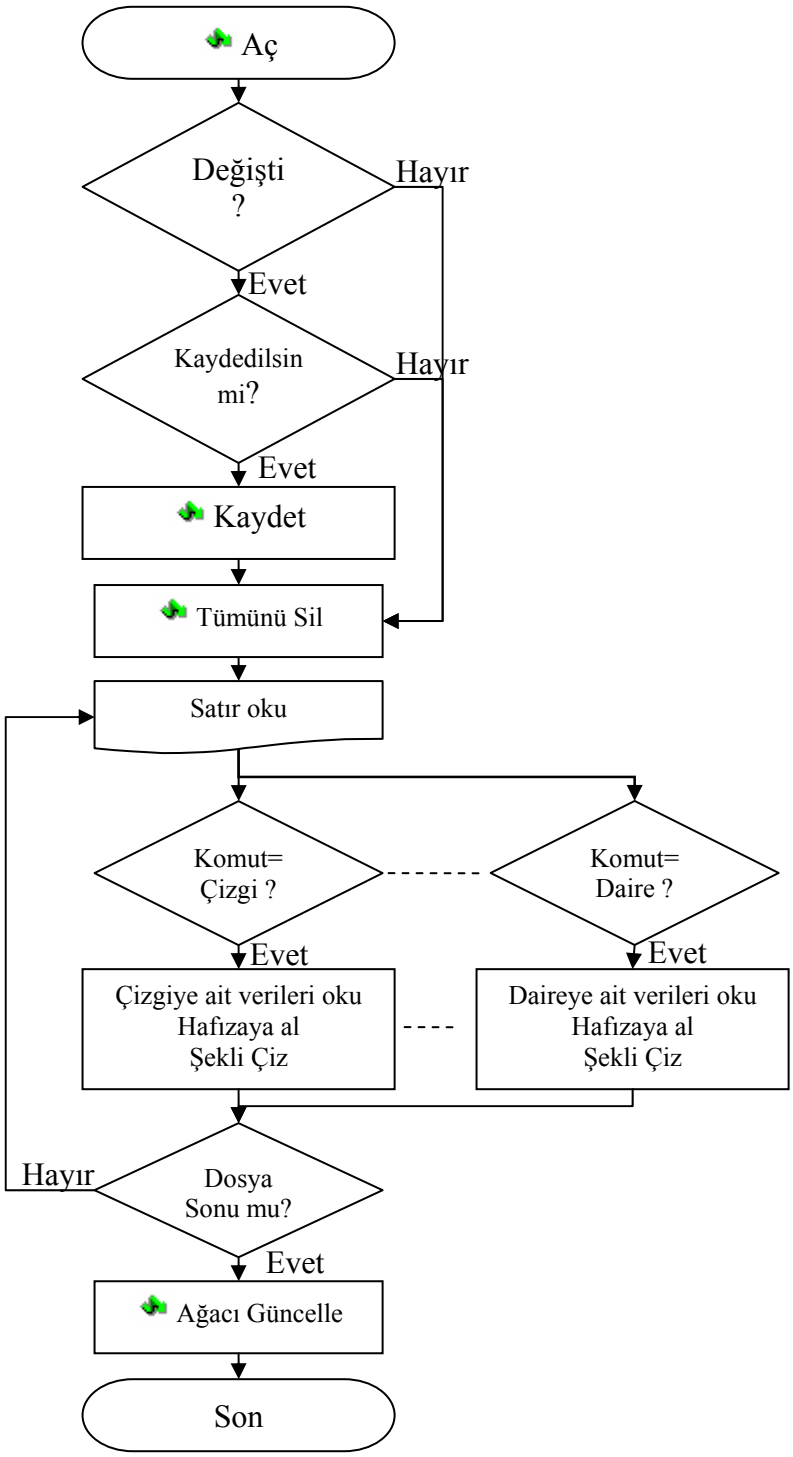
Yeni komutu seçildiğinde bir çizim yapıldıysa veya çizimde değişiklik yapıldıysa mevcut çizimin kaydedilip kaydedilmeyeceği sorulmaktadır. Kaydet seçilirse işlem kaydet alt programına yönlendirilmektedir. Kaydetme işleminden sonra tümünü sil alt programına yönlendirilerek çizim alanının temizlenmesi sağlanır. Hayır seçildiğinde sadece tümünü sil alt programı çalıştırılarak çizim alanının temizlenmesi sağlanmaktadır. Yeni komutunun algoritması Şekil 4.4'deki gibidir.



Şekil 4.4 Yeni komutu akış çizelgesi

4.2.2. Aç komutu ve akış çizelgesi

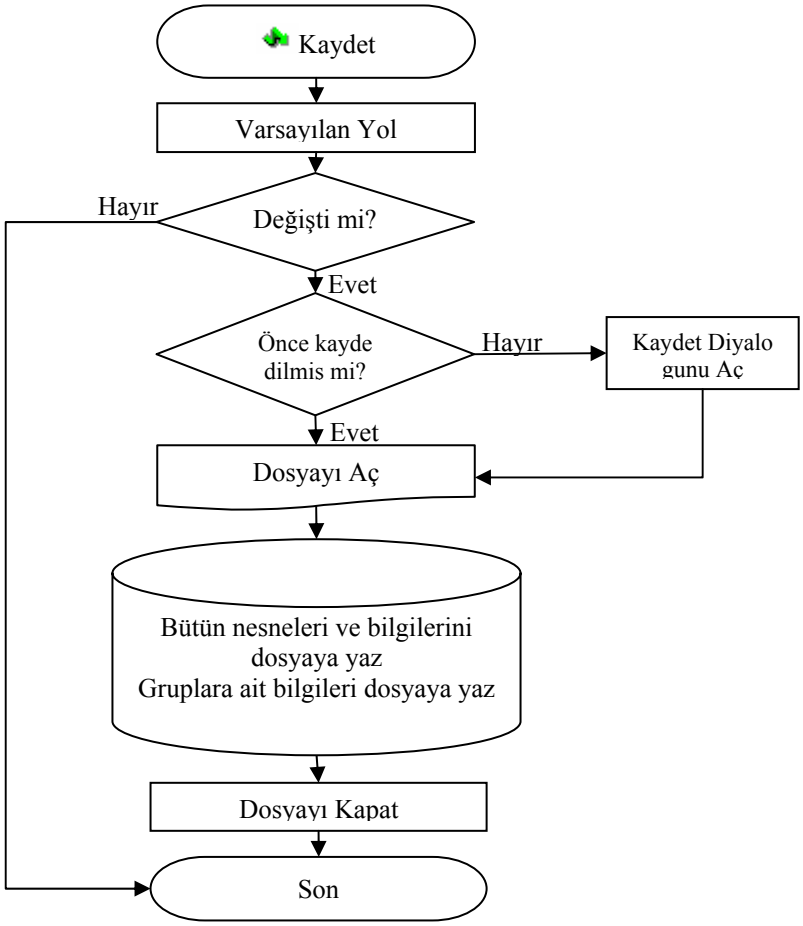
Aç komutu seçildiğinde sayfada değişiklik yapılıp yapılamadığı test edilmektedir. Değişiklik yapıldıysa kullanıcıya değişikliklerin kaydedilip kaydedilmeyeceği sorulmaktadır. Kaydet seçilirse kaydet alt programı çalıştırılmaktadır. Hayır, seçilirse çalışan programın yolu alınarak programın bulunduğu klasörde yer alan çizimler klasörü başlangıç klasörü olarak Aç penceresinde gösterilmektedir. Aç penceresinin fxs uzantılı dosyaları süzmesi sağlanmaktadır. Kullanıcı bu pencereden istediği dosyayı seçerek aç butonuna bastığında çizim alanı temizlenerek dosya okunur, kaydedilmiş olan şekiller çizilerek ekranda gösterilir. Dosyadan okunan grup bilgilerine göre ağaç yapısı oluşturulur. Aç komutunun akış çizelgesi şekil 4.5'deki gibidir.



Şekil 4.5 Aç komutu akış çizelgesi

4.2.3. Kaydet komutu ve akış çizelgesi

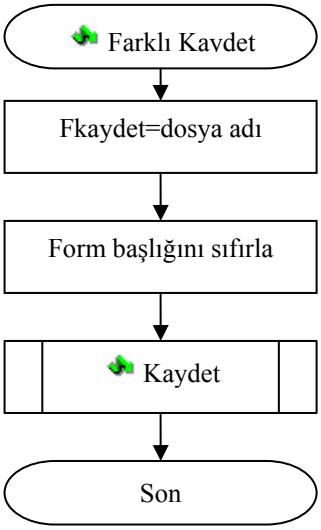
Kaydet komutu seçildiğinde dosya daha önce kaydedilmiş ise doğrudan ilgili dosyaya değişiklikler kaydedilmektedir. Eğer dosya daha önce kaydedilmemiş ise kaydet diyalog penceresi açılır. Kaydet diyalogu açılırken varsayılan klasör olarak programın bulunduğu klasörde yer alan Çizimler dizini varsayılan klasör olarak diyalog penceresinde gösterilir. Kullanıcı dosya ismini ve kayıt yerini seçerek kaydet düğmesine bastığında hafızada tutulan bilgiler belirtilen dosyaya kaydedilir. Dosyalar "fxs" uzantısı ile kaydedilmektedir. Kaydet komutunun akış çizelgesi şekil 4.6'daki gibidir.



Şekil 4.6 Kaydet komutu akış çizelgesi

4.2.4. Farklı kaydet komutu ve akış çizelgesi

Farklı kaydet komutu seçildiğinde form'un başlık çubuğu sıfırlanmaktadır ve Kaydet komutu çalıştırılmaktadır. Kaydet komutunun başında form başlığı test edilmektedir. Bu işlem Şekil 4.6'daki "Önce Kaydedilmiş mi?" işlemine karşılık gelmektedir. Bu işlemde çizim daha önce kaydedilmişse formun başlığında program adından sonra yer alan dosya adına bakılmaktadır. Eğer program adından sonra dosya adı varsa daha önce kayıt yapıldığını yoksa kayıt yapılmadığını belirtmektedir. Bu nedenle de farklı kaydet komutu seçildiğinde form başlığını sıfırlanması bu dosyanın ilk defa kaydedildiği anlamına gelir. Kaydet komutu çalıştığında global olarak tanımlanan fkaydet değişkenine bakar eğer fkaydet değişkeninde bir değer varsa kaydet diyalog kutusunun başlığı Farklı Kaydet olarak değiştirilir. Kaydet komutunda açılan diyalog kutusu sadece başlığı değiştirilerek açılmış olur. Farklı kaydet komutunun akış çizelgesi şekil 4.7'deki gibidir.



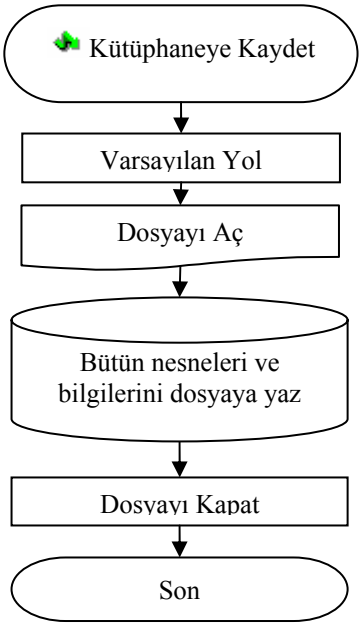
Şekil 4.7 Farklı kaydet akış çizelgesi

4.2.5. Kütüphaneye kaydet komutu ve akış çizelgesi

Kütüphaneye kaydet komutu seçildiğinde; çalışan programın bulunduğu dizinde yer alan Kütüphane klasörü, kaydet diyalog kutusunda varsayılan klasör olarak açılmaktadır. Kullanıcı kaydedilecek dizini seçip dosya ismini verdikten sonra çizilen şekiller belirtilen dosyaya kaydedilmektedir. Kaydet komutundan farklı olan özellikleri şunlardır;

- ✓ Silinen nesneler dosyaya kaydedilmez
- ✓ Dosyaya grup bilgileri kaydedilmez

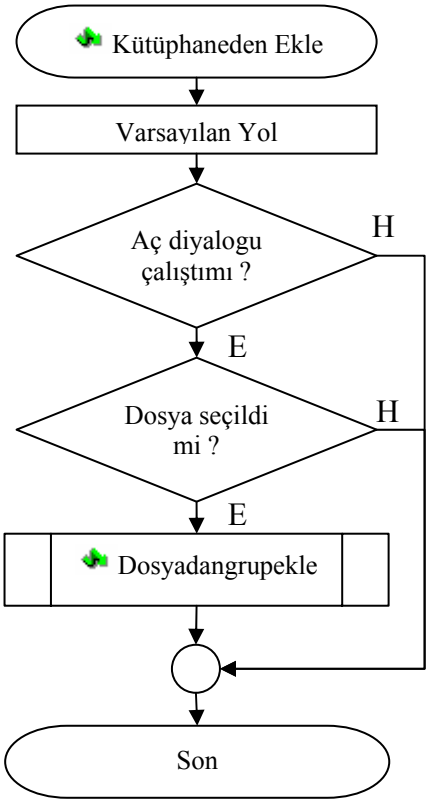
Kütüphaneye kaydedilen dosyalar "kxf" uzantısı ile kaydedilmektedir. Şekil 4.8'de kütüphaneye kaydet komutuna ait akış çizelgesi gösterilmiştir.



Şekil 4.8 Kütüphaneye kaydet komutu akış çizelgesi

4.2.6. Kütüphaneden ekle komutu ve akış çizelgesi

Kütüphaneden ekle komutu seçildiğinde aç diyalog kutusunda varsayılan klasör olarak kütüphane klasörü gösterilir. Buradan kullanıcının seçtiği dosya ağaç yapısında bir grup olarak gösterilerek çizim alanında nesneler gösterilir. Eklenen grupta yer alan nesnelerin konumlarına göre grup çerçevesi çizilir ve grup seçili hale getirilir. Şekil 4.9'daki dosyadangrupekle alt programı kütüphane konusunda tanıtılacaktır. Kütüphaneden ekle komutunun akış çizelgesi şekil 4.9'daki gibidir.



Şekil 4.9 Kütüphaneden ekle akış çizelgesi

4.2.7. Dxf al komutu ve akış çizelgesi

Dxf al komutu seçildiğinde aç diyalog kutusu ekrana gelmektedir. Buradan açılmak istenen dxf dosya seçilir. Dosyanın açılması ve programda görüntülenebilmesi için dxf dosyanın yapısının anlaşılması gerekir. Dxf dosya yapısı kısaca şöyledir;

DXF dosyası kod ve değerlerden oluşur. Kodlar grup kod olarak bilinir. Grup kodlar integer tipteki sayılardır. Bu sayılar takip eden verinin neyi ifade ettiğini gösterir. DXF dosyalarında her bölüm grup kod 0 (sıfır) ile başlar ve SECTION stringi takip eder. Bunu grup kod 2 takip eder ve bölüm adını gösteren string takip eder (Örneğin, HEADER). Sonraki bölümde tanımlanmış elemanların grup kodları ve değerleri yer alır. Bölümün (Section) bitişi 0(sıfır) ve onu takip eden ENDSEC metni ile olur(www.autodesk.com). DXF dosyalar küçük çizim ve çıktılar için az sayıda bölümden oluşabilir.

Ayrıntılı DXF dosyalarında şu bölümler yer alır:

HEADER bölümü: Çizim hakkında genel bilgileri içerir. AutoCAD versiyon numarasını ve sistem değişkenlerinden ibarettir. Bazı parametreler değişken adını ve değerlerini ilişkilendirir.

CLASSES bölümü: Uygulamalar için tanımlanan sınıf bilgilerini tutar. Blocks, Entities ve Object bölümleri gibi veritabanında yer alır.

TABLE Bölümü: Takip eden sembol tablosu için tanımlamaları içerir. Table bölümünde birkaç tablo yer alır:

APPID (application identification table)

BLOCK_RECORD (block reference table)

DIMSTYLE (dimension style table)

LAYER (layer table)

LTYPE (linetype table)

STYLE (text style table)

UCS (User Coordinate System table)

VIEW (view table)

VPORT (viewport configuration table)

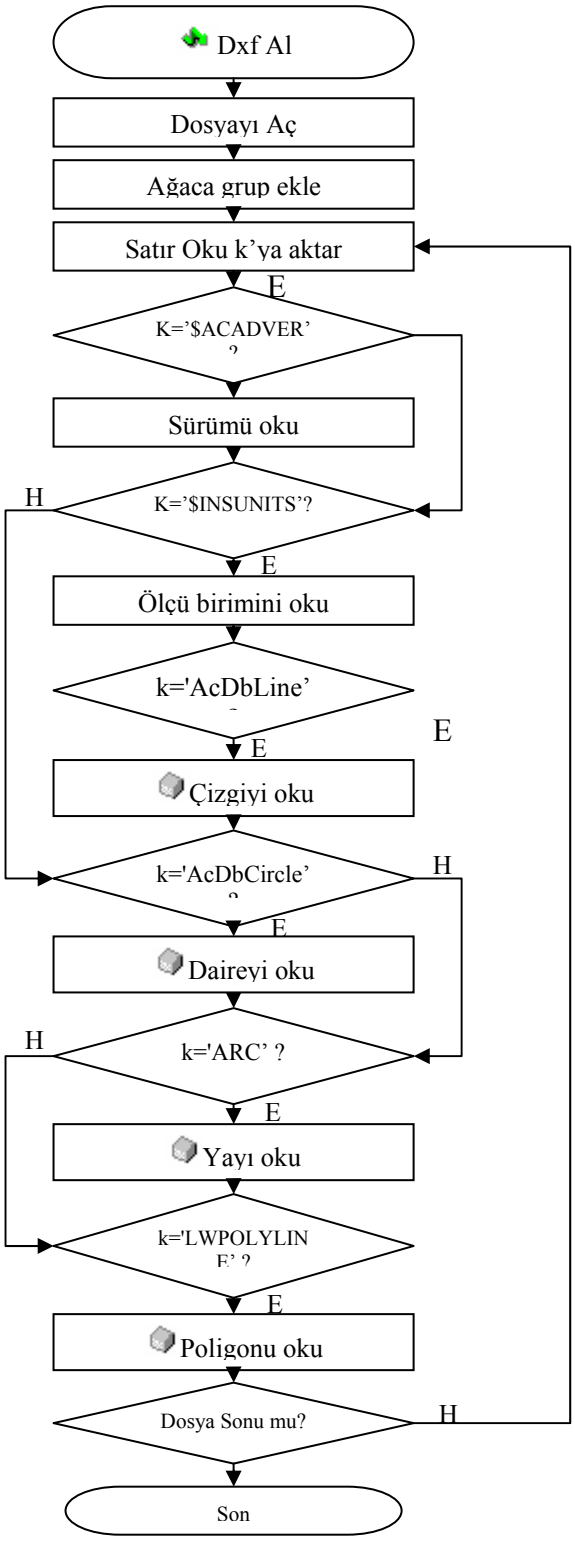
BLOCKS Bölümü: Blok tanımlamalarını içerir ve diğer blokta yer alan çizim nesnelerine referans eder.

ENTITIES Bölümü: Çizimde yer alan grafiksel nesneleri içerir. Blok bölümünde referans edilen içeriklerden oluşur.

OBJECT Bölümü: Çizimde yer alan grafiksel olmayan nesneleri içerir. Entities bölümünde yer almayan bütün kayıtları veya sembol tablosu kayıtlarını veya sembol tablosunu içerir.

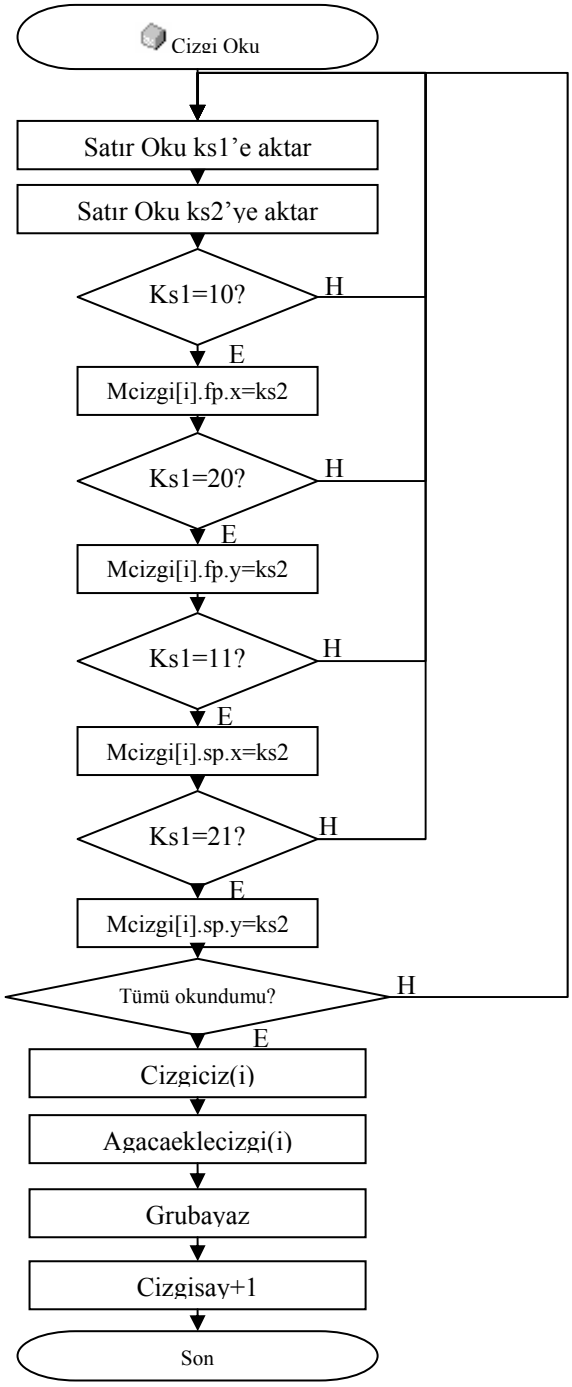
THUMBNAILIMAGE Bölümü: Çizimde yer alan resim verilerini içerir. Bu bölüm isteğe bağlıdır.

Çizim nesneleri entities bölümünde tanımlanmıştır. Bu bölümde çizim nesnelerine ait bilgiler alt alta satırlardan oluşan bir blok içinde yer alır. Bu çalışma için geliştirilen programda dxf dosyaların başlık bölümünde yer alan sürüm ve ölçü bilgileri test edildikten sonra nesneler okunarak programda tanımlanmış nesnelere dönüştürülmektedir. Daha sonra okunan nesneler ekranda gösterilmektedir. Alınan dosya ağaç yapısında bir grup olarak gösterilmektedir. Böylece bir çizim alanına istenilen sayıda dxf dosya alınarak istenildiği gibi yerleştirilebilmektedir. Dxf al komutunun akış çizelgesi şekil 4.10'daki gibidir.

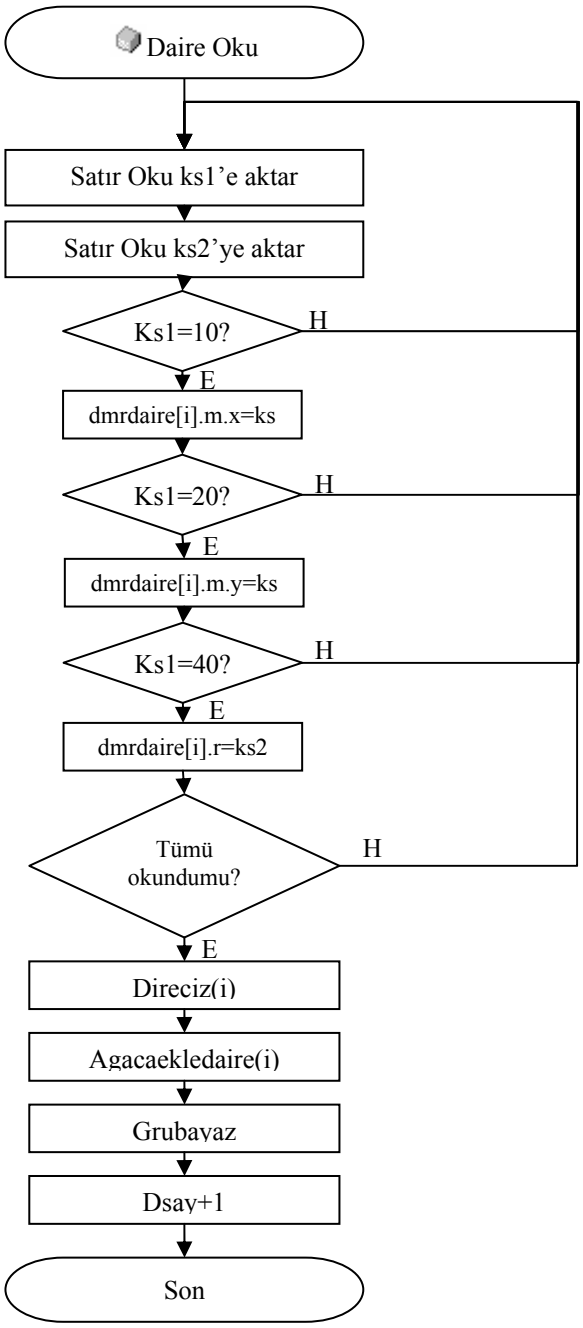


Şekil 4.10 Dosya Al (*.dxf) komutu akış çizelgesi

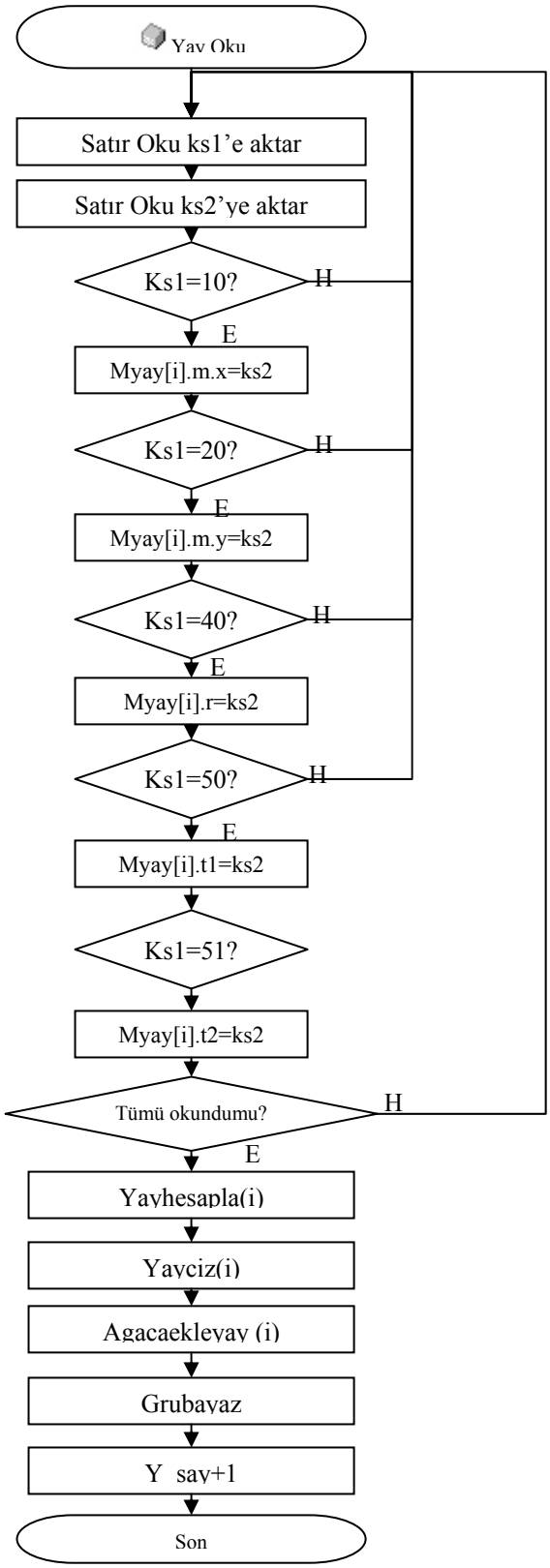
Dxf al komutunun akış çizelgesi verilmiştir. Bu çizelgede gösterilen çizgiyi oku, daireyi oku, yayı oku, poligonu oku komutlarının ayrıntılı akış çizelgeleri Şekil 4.11, Şekil 4.12, Şekil 4.13 ve Şekil 4.14'de gösterilmiştir.



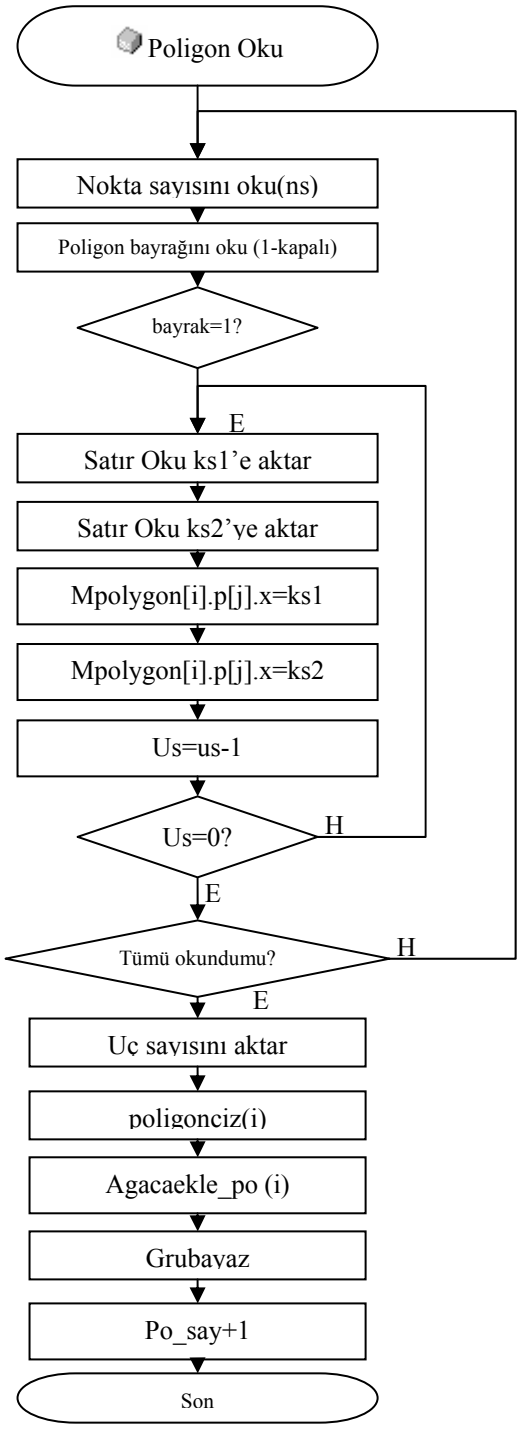
Şekil 4.11 Çizgi oku komutu akış çizelgesi



Şekil 4.12 Daire oku komutu akış çizelgesi



Şekil 4.13 Yay oku komutu akış çizelgesi



Şekil 4.14 Yay oku komutu akış çizelgesi

4.3. Düzen Menüsü

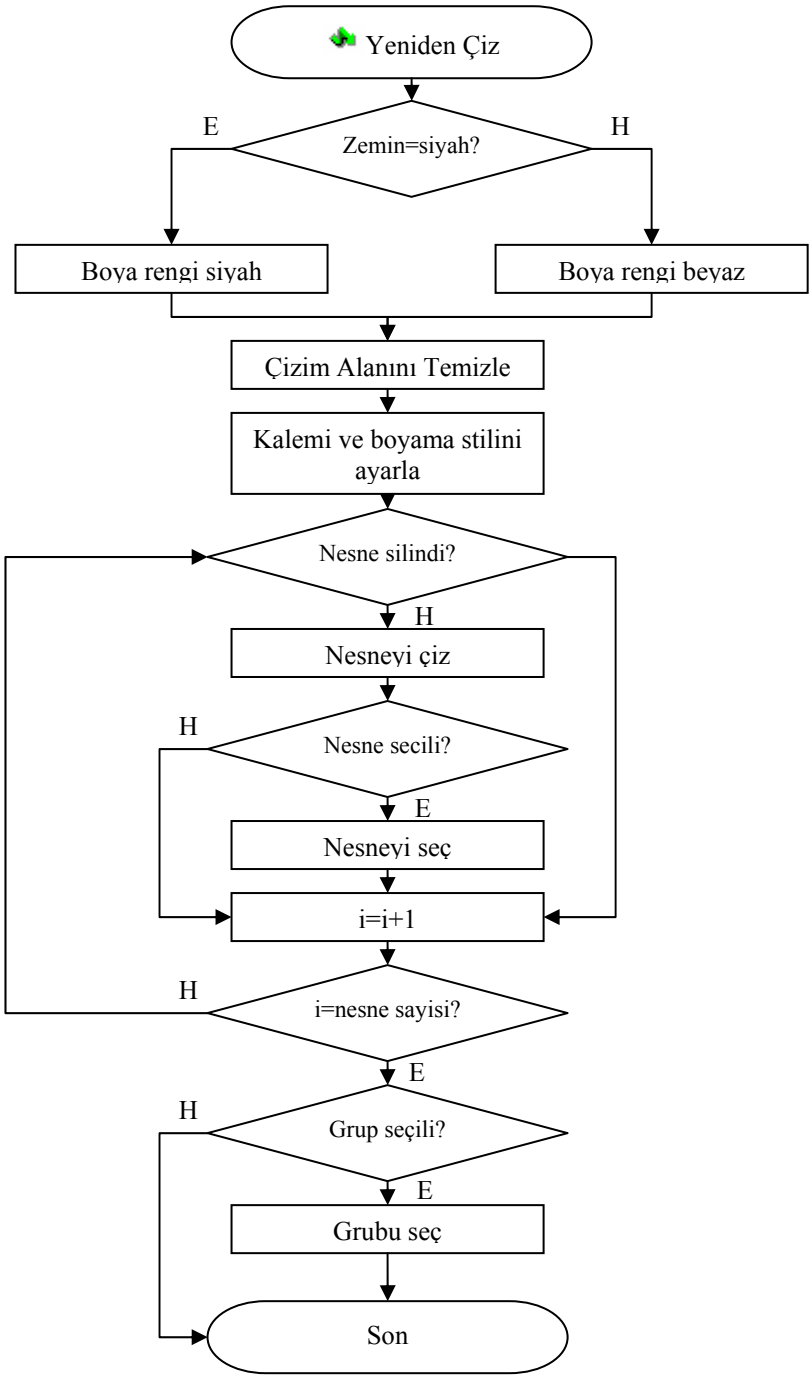
Düzen menüsünde yeniden çiz, tümünü sil ve uygulama tercihleri komutları yer almaktadır. Düzen menüsünün kısa yol tuşu Alt+z'dir. Düzen menüsünün ekran görünümü Şekil 4.15'de gösterilmiştir.



Şekil 4.15 Düzen menüsü

4.3.1. Yeniden çiz komutu

Bu komut seçildiğinde öncelikle zemin rengi test edilerek çizim alanı temizlenmektedir. Zemin siyahsa siyah, beyaz ise beyaz renk ile tüm alan silinir. Çizim için gerekli ayarlar yapılır. Tüm nesneler hafızada tutulan bilgilere göre yeniden çizilerek çizim alanında gösterilir. Örneğin çizgilerin güncellenmesi esnasında birden çizgi sayısına kadar bir döngü kurulur. Gerekli verileri içeren kayıt tipindeki değişkende tutulan bilgilere göre tüm çizgiler yeniden çizilir. Çizgiler için hafızada başlangıç ve bitiş koordinatları saklanmaktadır. Bu işlem tüm nesneler için uygulanır. Böylece çizim esnasında zamanla oluşan bazı piksellerdeki renk değişimi veya artıklar temizlenmiş olur. Bu işlem çizim alanını bütünüyle boyamak suretiyle temizleyerek ekranda oluşan artıkları temizlemektedir. Yeniden çiz komutu çalıştırıldığında ağaç yapısına bir değişiklik yapılmamaktadır. Ağaç yapısının güncellenmesi ağaç altında yer alan ağacı güncelle komutu ile yapılmaktadır. Yeniden çizim işleminde şekiller silinmiş ise yeniden çizilmemektedir. Yeniden çiz komutunun akış çizelgesi Şekil 4.16'daki gibidir.



Şekil 4.16 Yeniden çiz komutu akış çizelgesi

4.3.2. Tümünü sil komutu

Tümünü sil komutu seçildiğinde öncelikle çizim alanı temizlenmektedir. Daha sonra global olarak tanımlanan tüm değişkenler başlangıç durumuna getirilmektedir. Global olarak tanımlanan bu değişkenler ve işlevleri şunlardır;

Sekil: O anda seçili komutun tutulmasını sağlar. String tipinde tanımlanmış bir değişkendir. Aldığı değer fare çizim alanında iken durum çubuğunda gösterilir. Çizim alanında fare tuşuna basıldığında hangi şeklin çizileceği bu değişkene göre belirlenir. Kullanıcı çizim araç çubuğundan veya çiz menüsünden istediği şekli seçerek bu değişkenin değer almasını sağlar.

cizgisay: Çizgilerin indekslenmesini sağlar. Kaç tane çizgi çizildiğini ve çizilen bir çizgiye ait bilgilerin hangi değişkene aktarılacağını belirlemeyi sağlar. Tam sayı tipinde tanımlanmıştır ve ilk değeri sıfırdır. Aşağıdaki değişkenlerde buna benzer biçimde diğer nesneleri indekslemeyi sağlar.

d_say: Dairelerin indekslenmesini sağlar.

y_say: Yayların indekslenmesini sağlar.

p_say: Ucu açık poligonların indekslenmesini sağlar.

po_say: Kapalı poligonların indekslenmesini sağlar.

e_say: Elipslerin indekslenmesini sağlar.

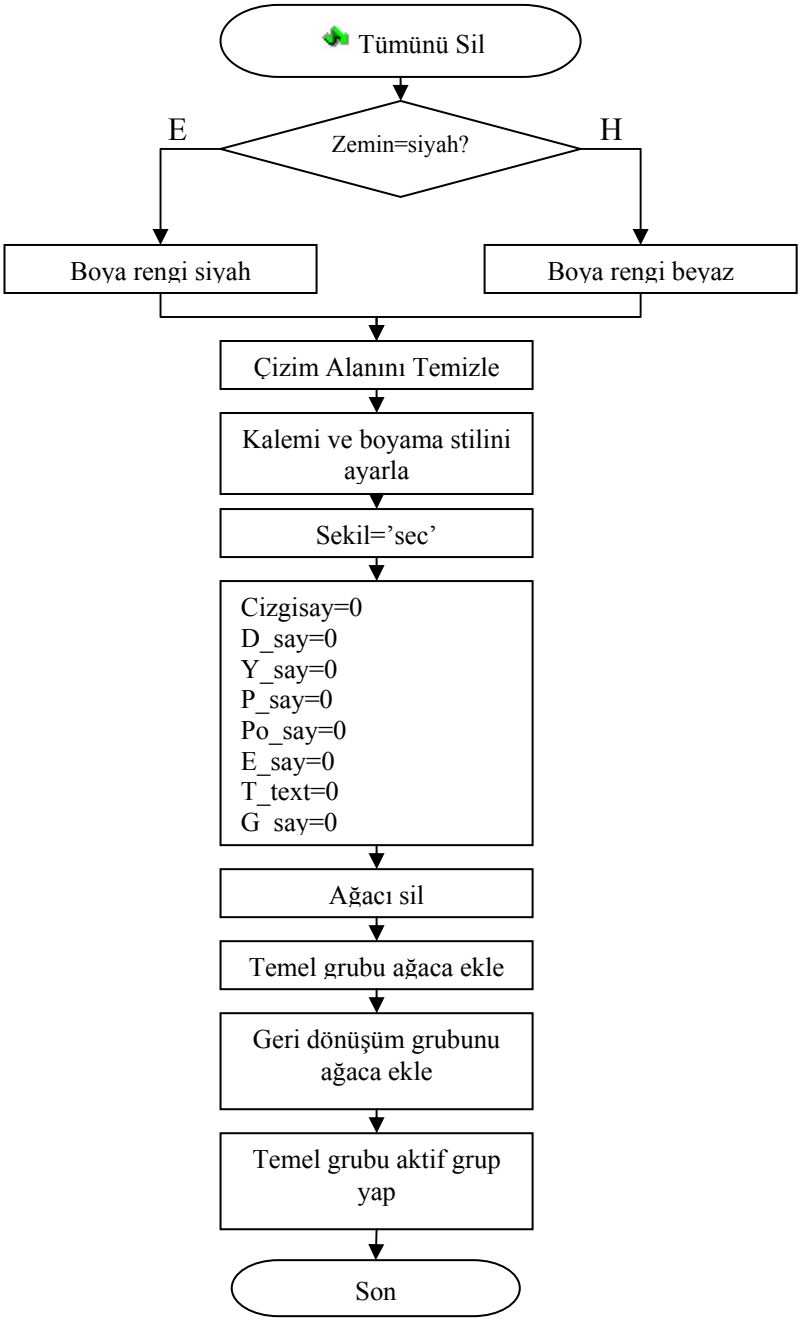
t_text: Yazıların indekslenmesini sağlar.

g_say: Grupların indekslenmesini sağlar.

aktifgrup: Çizilecek nesnelerin ağaç yapısında hangi grupta yer alacağını belirtmek için kullanılan değişkendir. Node tipinde tanımlanmış bir değişkendir.

Kullanıcı ağaçtan bir gruba tıkladığında o grup aktif grup yapılmakta ve çizilecek nesneler o grup altında gösterilmektedir.

Tümünü sil komutunun akış çizelgesi Şekil 4.17'deki gibidir.

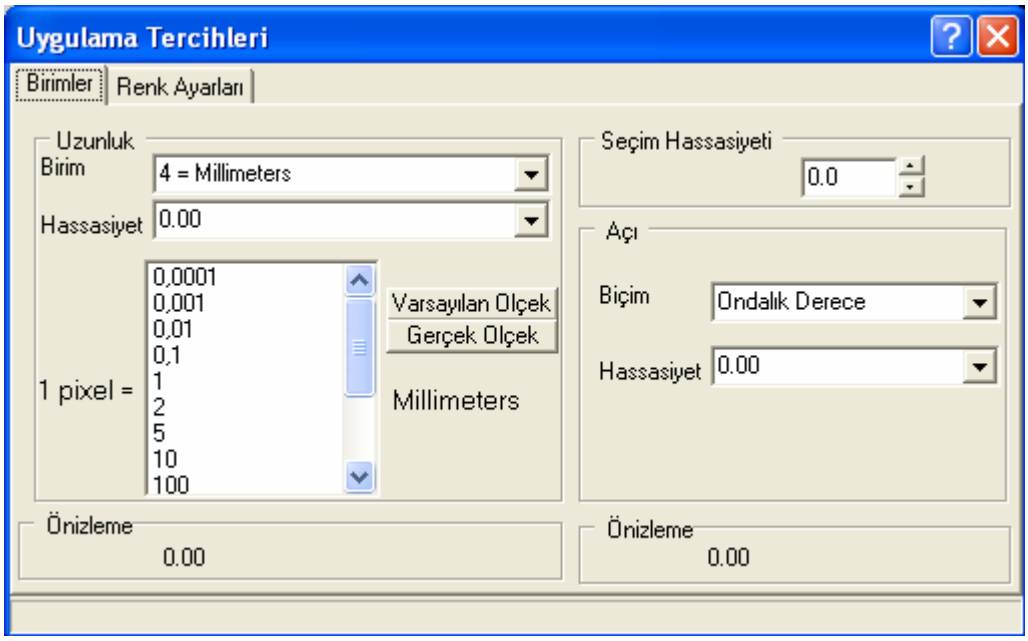


Şekil 4.17 Tümünü sil komutu akış çizelgesi

4.3.3. Uygulama tercihleri komutu

Uygulama tercihleri komutu seçildiğinde yeni bir form açılmaktadır. Bu form sayfa yapısındadır. Şekil 4.18'de birinci sayfa gösterilmiştir. Burada seçim hassasiyeti, uzunluk ölçü birimi, uzunluk hassasiyeti, ölçek, açı birimi ve açı hassasiyeti değişkenlerinin değerleri değiştirilebilmektedir.

Fare çizim alanında hareket ederken farenin konum bilgilerine göre bütün çizilmiş nesneler taranmakta ve durum çubuğunda farenin üzerine geldiği nesne gösterilmektedir. Seçim hassasiyeti farenin konum bilgilerinin çizim nesnelerinin denklemlerinde yerine konurken ne kadar hassas olacağını belirtmeyi sağlar. Ondalık kısım arttırıldıkça konum daha hassas taranmaktadır. Seçim hassasiyetinin değişimi yukarı aşağı butonları aracılığı ile yapılmaktadır.



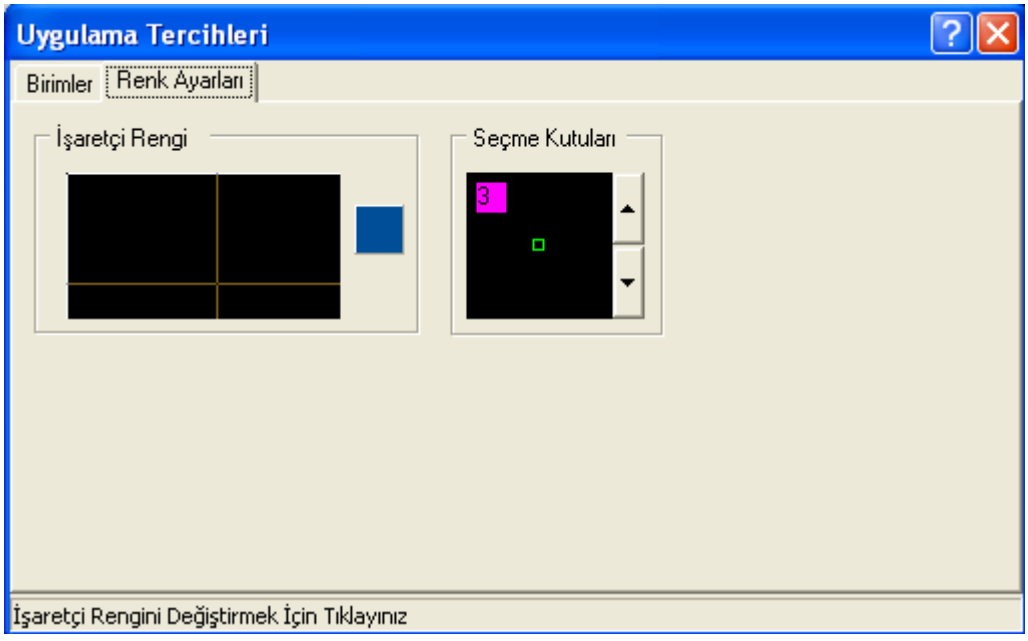
Şekil 4.18 Uygulama tercihleri formu birimler sayfası ekran görünümü

Uzunluk birimi açılan kutudan değiştirilmektedir. Burada dxf dosyasında yer alabilecek bütün uzunluk birimleri tanımlanmıştır. Bunlardan bazıları mm, cm, m, km, feet, mil, yard, micron gibi birimlerdir. Uzunluk hassasiyeti de pikselden gerçek birimlere yapılacak dönüşümlerde ondalık basamak sayısının seçilmesini

sağlamaktadır. Bunun altında yer alan ölçek ayarlamasında bir pikselin kaç birim gerçek ölçek olarak gösterileceği seçilmektedir. Ne olursa olsun ekranda gösterilen en küçük nokta pikseldir ve tam değerlerden oluşur. Gerçek birimlerin ekranda gösterilişinde bu kurala uyulması gerekir.

Açı grup kutusunda çizim esnasında açıların gösterimi ve girilmesi ile ilgili ayarlamalar yapılmaktadır. Buradan açı birimi derece, radyan vb. seçilmektedir. Hassasiyet açılan kutusundan da, açıların gösteriminde kullanılacak ondalık basamak sayısı seçilmektedir.

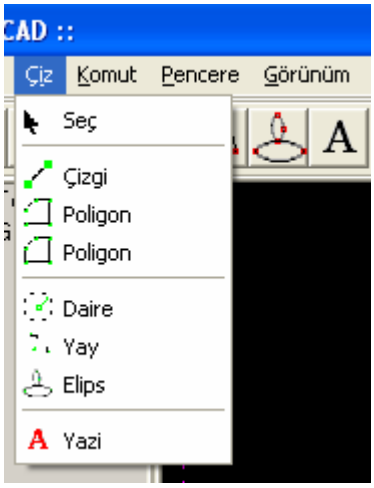
Uygulama tercihleri formunda yer alan diğer bir sayfa da renk ayarları sayfasıdır. Bu sayfanın ekran görünümü Şekil 4.19'da gösterilmiştir. Buradan işaretçi aktif yapıldığında ekranda gösterilecek rengi değişmektedir. Yapılan renk seçiminin siyah ekranda ve beyaz ekranda nasıl görüntüleneceği mavi kutu tıklanarak denenebilir. Soldaki siyah kutu üzerine tıklandığında Windows'ta standart olarak tanımlı renk diyalog kutusu açılmaktadır. Buradan kullanıcı istediği rengi seçerek aynı yerde ön izleme yapılabilmektedir.



Şekil 4.19 Uygulama tercihleri formu renk ayarları sayfası ekran görünümü

Seçme kutuları grup kutusunda da seçme kutularının boyutu yukarı aşağı düğmelerine tıklanarak değiştirilebilmektedir. Aynı zamanda kutu ön izlemesi yapılan yere tıklanarak işaretçi kutusunun renginin değişimini sağlayan renk diyalog kutusu açılmaktadır.

4.4. Çiz Menüsü



Şekil 4.20 Çiz menüsü ekran görünümü



Şekil 4.21 Çizim araç çubuğu ekran görünümü

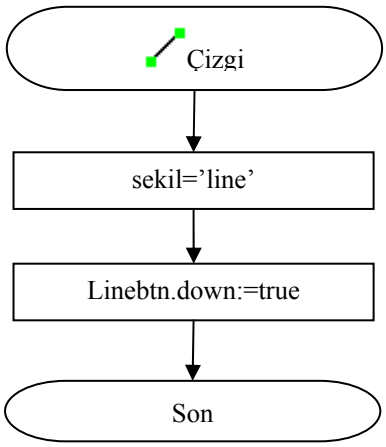
Çiz menüsünde çizim alanında tanımlanmış nesnelerin seçimi yapılmaktadır. Aynı seçimler çizim araç çubuğundan da yapılabilmektedir. Şekil 4.20'de çiz menüsünde yer alan komutlar gösterilmiştir. Şekil 4.21'de de çizim araç çubuğu gösterilmiştir. Çiz menüsünden yapılan işlem ile çizim araç çubuğundan yapılan işlemler eşdeğerdir. Herhangi bir nesne seçildiğinde global olarak tanımlanmış olan şekil değişkenine seçilen komut aktarılmaktadır. Çizim alanında farenin hareketine veya tıklanmasına göre diğer işlemler gerçekleştirilmektedir. Nesnenin seçiminde sadece sekil değişkenine değer aktarılırken takip eden tüm işlemler çizim alanında

tanımlı olan Mouse.down ve Mouse.move olaylarında yazılı olan kodlarla sağlanmaktadır. Örneğin çizim araç çubuğundan veya çiz menüsünden çizgi komutu seçildiğinde sekil değişkenine line verisi aktarılır. Bu esnada kullanıcı çizim alanında farenin tuşuna bastığında Mouse.down olayına yazılı olan kodlar çalıştırılır. Burada sekil değişkeni line verisini içerdiği için basılan koordinatlar nokta tipinde tanımlanmış olan bas değişkenine aktarılır ve mantıksal olarak tanımlanmış olan drawing değişkenine doğru verisi aktarılır. Bundan sonra çalışan olay Mouse.move olacaktır. Burada drawing değişkeni ve şekil değişkeni test edilerek komutun çizgi olduğu ve çizme işleminin başladığı anlaşılır. Buna göre de bu olayın her çalışmasında önceki konum bilgilerinin tutulduğu m değişkenindeki değerlere göre önceki çizgi silinir ve yeni koordinatlara göre yenisi çizilir. Bu esnada koordinat bilgileri ve çizginin uzunluğu ve eğimi gibi bilgiler durum çubuğunda gösterilir. Bu işlemler fare tuşuna ikinci defa basılıncaya kadar bu şekilde devam eder. Tuşa ikinci defa basıldığında şekil değişkeni hala line verisi içerdiğinden buna göre line bloğuna girilir burada drawing test edilmektedir. Drawing doğru değeri aldığı içinde basılan koordinatlar son değişkenine aktarılır. Drawing değişkeni yeniden yanlış yapılır, k değişkenine değer atanır. Blok sonunda da k değişkeni test edilir doğru değer atanmışsa çizgi çizme işlemi tamamlanmış anlamına gelir. Başlangıç ve bitiş koordinatlarının saklandığı baş ve son değişkenlerindeki veriler kayıt tipinde tanımlanmış olan mline değişkenine aktarılır. Ve bir adet çizgi çizilerek verileri hafızada saklanmış olur. Bu işlemler ve adımlar değişik şekillerde diğer nesnenin çizimi esnasında da yinelenir. Çizilecek nesnenin seçiminden sonraki bütün işlemler çizim alanının Mouse.down ve Mouse.move olaylarına yazılan kodlara göre yapılmaktadır.

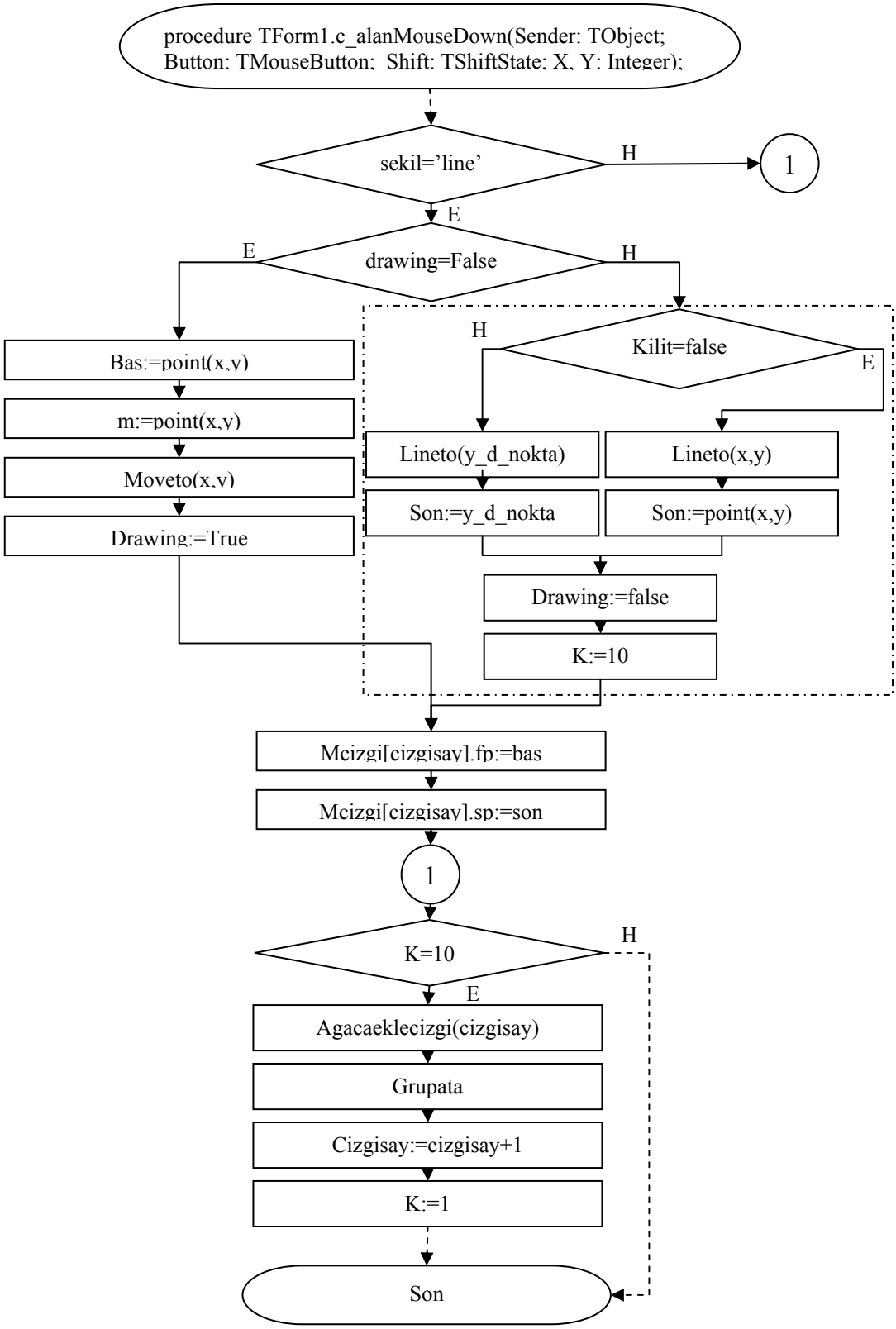
Çizgi, açık poligon ve kapalı poligon çiziminde kullanılan ve çizimi eğime göre yatayda ya da düşeyde kilitleyen kilit işlemi vardır. Kilit F8 kısa yol tuşu ile aktif ya da pasif yapılmaktadır. Çizgiyi ele alırsak kilit aktifse çizim başladıktan sonraki farenin hareketine göre Mouse.move olayında yazılı olan kodlar aracılığıyla eğim hesaplanır ve çizgi yatayda yada düşeyde artar azalır. Çizginin çizimi esnasında olan işlemleri akış şemasında göstermeye çalışalım. Bu işlemin gösterimini komutun seçilmesi, çizim alanında fare tuşuna birinci ve ikinci defa basılması, farenin çizim

alanında hareket ettirilmesi şeklide ayrı akış çizelgeleri ile gösterelim. Çünkü nesnel programlamada çalışılan nesneler ve yapılan işlemlere göre gerçekleşen olaylar vardır. Her olay bir takım hazır verileri de alt programa aktarır. Bu görsel programlamanın hazır sunduğu bir araçtır. Örneğin çizim alanında fare tuşuna basıldığında Mouse.down olayı gerçekleşir. Bu olaya kullanıcı tarafından yazılan kodlar çalıştırılır. Bu olay şu değerleri alt programa aktarır. Olayı gönderen nesnenin adı, farenin basılan tuşu sol, sağ gibi, klavyede basılı olan shift veya ctrl gibi tuşlar ile basılan konumun koordinatları x,y değişkenlerinde olmak üzere alt programa aktarılır.

Çizim menüsünden veya çizim araç çubuğundan çizgi komutu seçildiğinde yapılan işlemler ait akış çizelgesi Şekil 4.22'de gösterilmiştir.



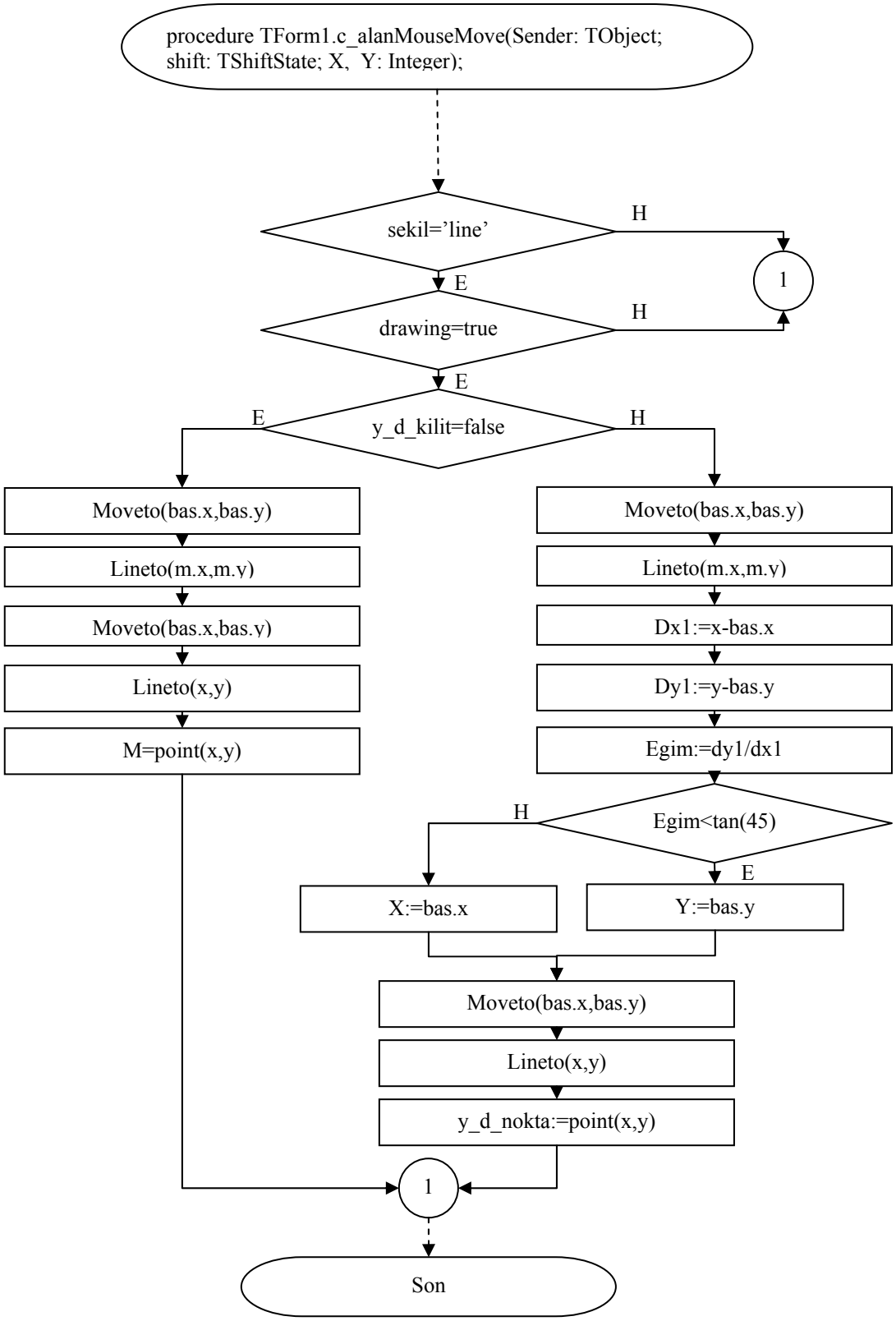
Şekil 4.22 Çiz/Çizgi komutu akış çizelgesi



Şekil 4.23 Çizgi seçildikten sonraki fare tuşuna ilk ve ikinci kez basıldığında oluşan akış çizelgesi

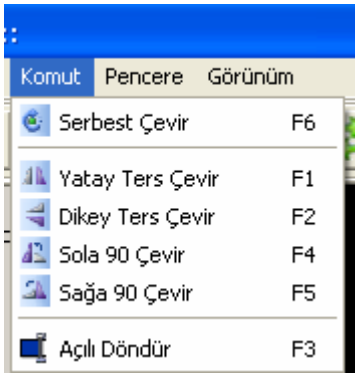
Şekil 4.23'de çiz menüsünden çizgi seçildikten sonra veya çizim araç çubuğundan çizgi komutu seçildikten sonra, çizim alanı üzerinde fare tuşuna basıldığında, Mouse.down olayına ait sadece çizgi işleminin nasıl yapıldığını gösteren akış çizelgesini göstermektedir. Burada çizgi komutu seçildikten sonra fare tuşuna ilk kez basıldığında drawing değişkeninin durumu false olacağı için akış çizelgesinden de görüldüğü işlem sol tarafa dallanacak ve buradaki işlemler gerçekleştirilecektir. Fare tuşuna ilk kez basıldıktan sonra da Mouse.move olayı çalışacaktır. Mouse.move olayında çizgi için yer alan işlemler de Şekil 4.24'de gösterilmiştir. Çizim alanında çizgi çizilirken fare tuşuna ikinci kez basıldığında yapılan işlemler Şekil 4.23'de kutu içerisine alınmıştır. İşlemler hem taşıma hem de basılma olaylarında yatay düşey kilit tuşunun durumuna göre dallanmaktadır. Basılış ve taşınış olaylarında sadece çizgiye ait değil tüm işlemlere ait kodlar yer aldığından akış çizelgelerinde diğer işlemlerin bulunduğu yerler kesikli ok ile gösterilmiştir. Çizme işlemi tamamlandıktan sonra nesne ağaca eklenmektedir. Kayıt tipinde dizi olarak tanımlanmış olan mcizgi değişkeninin indeksi olan cizgisay değişkeni sonraki boş hafıza adresini göstermesi için bir arttırılmıştır.

Çizim alanında farenin taşınışı esnasında çizgi komutu seçilmiş ise ve çizim işlemi başlamış (drawing=true) ise yatay düşey kilit değişkeninin durumuna göre işlemler ikiye dallanmaktadır. Kilit aktif değilse eski çizgi silinip yenisi çizilerek fare tuşuna ikinci kez basılıncaya bu işlemler yapılmaya devam etmektedir. Kilit aktif ise çizginin eğimi hesaplanarak sadece yatayda ya da sadece düşeyde hareket etmesi sağlanmaktadır. Aslında kilit aktifse eğim farenin konumuna göre hesaplanmaktadır. Eğim radyan olarak bulunduğu için 45 derecenin radyanı ile karşılaştırılarak bu değerden küçükse yatay büyükse dikey olarak çizginin çizilmesi sağlanır. Kilit aktif iken farenin konumuna göre gösterilen çizginin ikinci noktası y_d_nokta değişkeninde saklanmaktadır. İkinci kez fare tuşuna basıldığında da kilit aktifse çizginin bitiş noktası y_d_kilit değişkenindeki değer olmaktadır. Kilit aktif değilse çizginin ikinci noktasının koordinatları son değişkenindeki değerler olmaktadır.



Şekil 4.24 Çizgi seçilip fare tuşuna basıldıktan sonra fare hareket ettirildiğindeki akış çizelgesi

4.5. Komut Menüsü

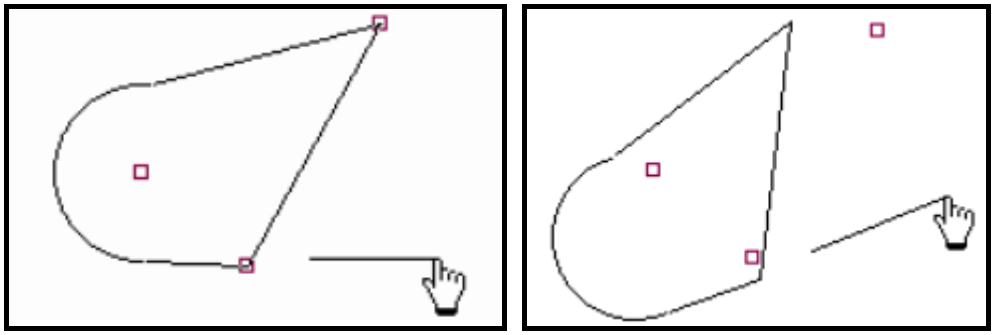


Şekil 4.25 Komut menüsü ekran görünümü

Komut menüsünde çizim nesneleri üzerinde yapılan döndürme işlemleri tanımlanmıştır. Komut menüsünün ekran görünümü Şekil 4.25'deki gibidir. Yerleştirme işlemlerinde kolaylık sağlaması için bu komutlara kısa yol tuşu atanmıştır. Nesneler döndürülürken bir referans noktasına göre konumları değiştirilmektedir. Bu menüde yer alan komutlar ve bu komutların akış şemaları şöyledir;

4.5.1. Serbest çevir komutu

Serbest çevir komutu seçili olan nesneleri veya bir grubu fare ile belirlenen referans noktasında farenin dönüş açısına göre döndürmeyi sağlar. Döndürme işlemi komut menüsünden serbest çevir komutu seçilerek veya araç çubuğundan serbest çevir komutu seçilerek başlatılabilir. Eğer herhangi bir grup veya nesne seçili değil ise ilk olarak kullanıcıdan döndürmek istediği nesneleri seçmesi istenmektedir. Seçim işleminde kullanıcı shift tuşuna basılı tutarak birden çok nesneyi seçebilir. Şekil 4.26'da çizilmiş olan bir şeklin döndürülüşü gösterilmiştir. Şekil bir yay ve ucu açık bir poligondan oluşmaktadır. Nesneler seçildikten sonra döndür komutu seçilir bir referans noktası seçilir ve dönme işlemini görselleştirilmesi için bir çizgi ile temsil edilir.

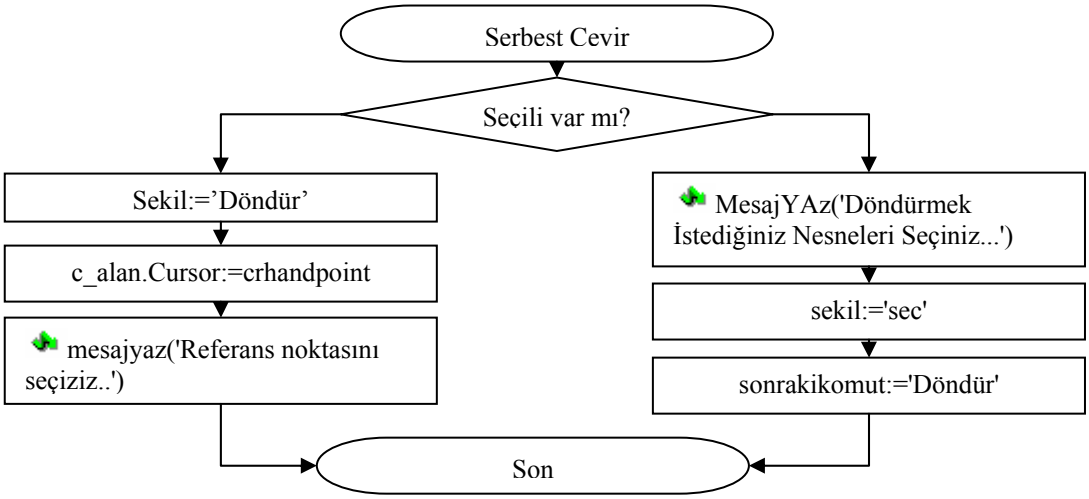


a. Açı sıfır derece iken b. Otuz derece döndürüldükten sonra

Şekil 4.26 Serbest çevir komutu ekran görünümleri

Şekil 4.26.a.'da serbest çevir komutu seçildikten sonra fare ile çizim alanında referans noktası seçilmiş ve fare sıfır derecede hareket ettirilmiştir. Şekil 4.26.b.'de fare yukarı doğru hareket ettirilmiştir ve çizgide oluşan açı kadar seçilen nesneler döndürülmüştür.

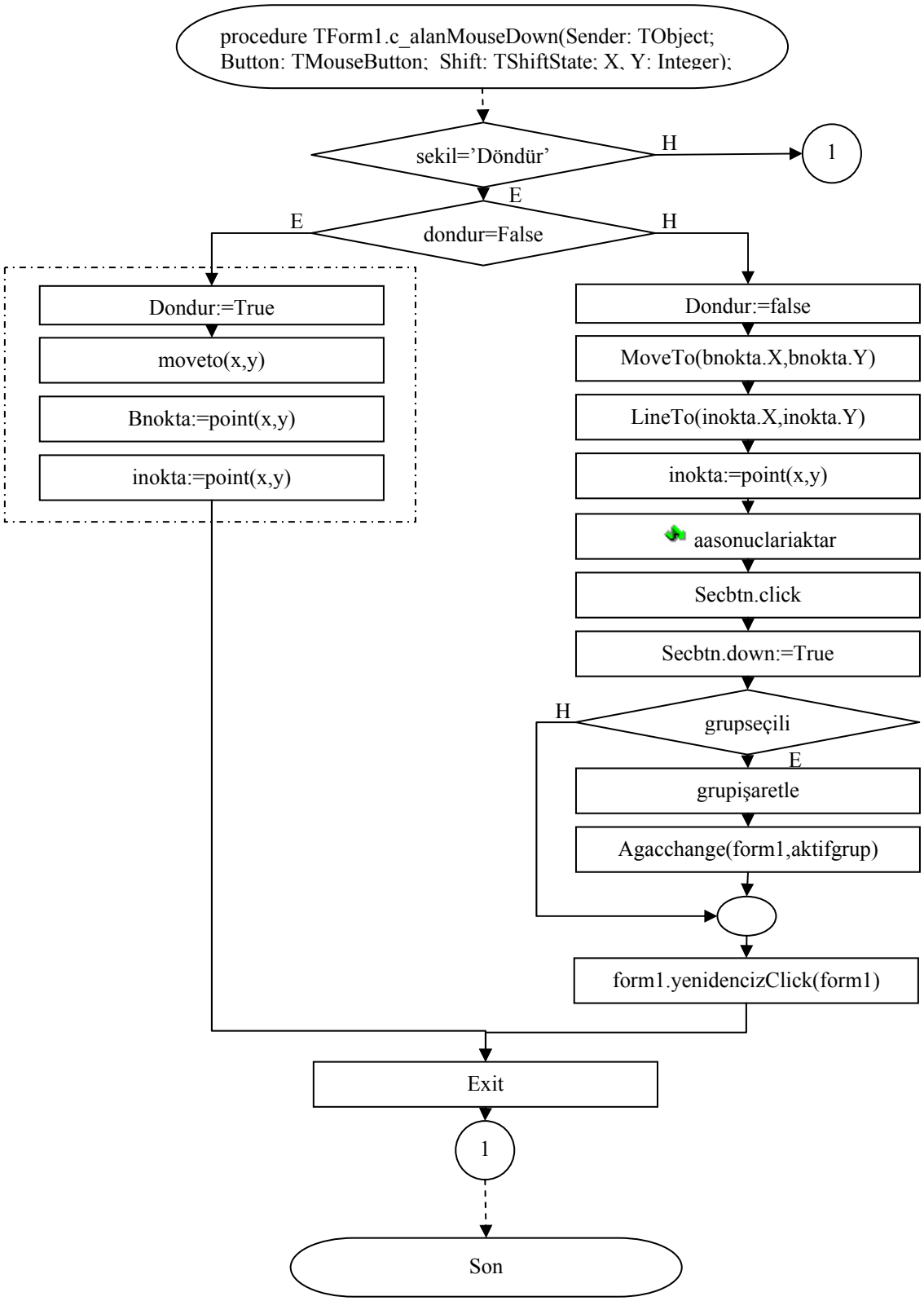
Serbest çevir komutu komut menüsünden seçildiğinde araç çubuğunda yer alan serbest çevir düğmesine basılmasını sağlayan kodlar çalışmaktadır. Yani her iki durumda da aynı işlemler çalışmaktadır. Serbest çevir komutu seçildiğinde yapılan işlemler ait akış şeması Şekil 4.27'deki gibidir.



Şekil 4.27 Serbest çevir komutu akış çizelgesi

Serbest çevir komutu seçildiğinde ilk olarak seçili nesne olup olmadığı test edilmektedir. Seçili nesne varsa program akış şeması sol tarafa dallanmaktadır. Seçili nesne yoksa önce sekil değişkenine ‘sec’ komutu aktarılmaktadır. Ve kullanıcı çizim alanında fare tuşu ile nesne veya nesneleri seçtikten sonra döndürme işlemine geçiş sağlanmaktadır.

Serbest çevirme işlemi çizim alanında görsel olarak yapıldığı için döndürme işlemi çizim alanında fare butonuna basılarak hareket ettirilmesi sonucunda gerçekleşmektedir. Bu amaçla sekil değişkenine ‘Döndür’ komutu aktarılmıştır. Bundan sonraki işlemler çizim alanında Mouse.down ve Mouse.move olaylarına yazılmış olan kodlar tarafından yapılmaktadır. Komut döndür olduğunda Mouse.down olayında gerçekleşen işlemler Şekil 4.28’deki gibidir.

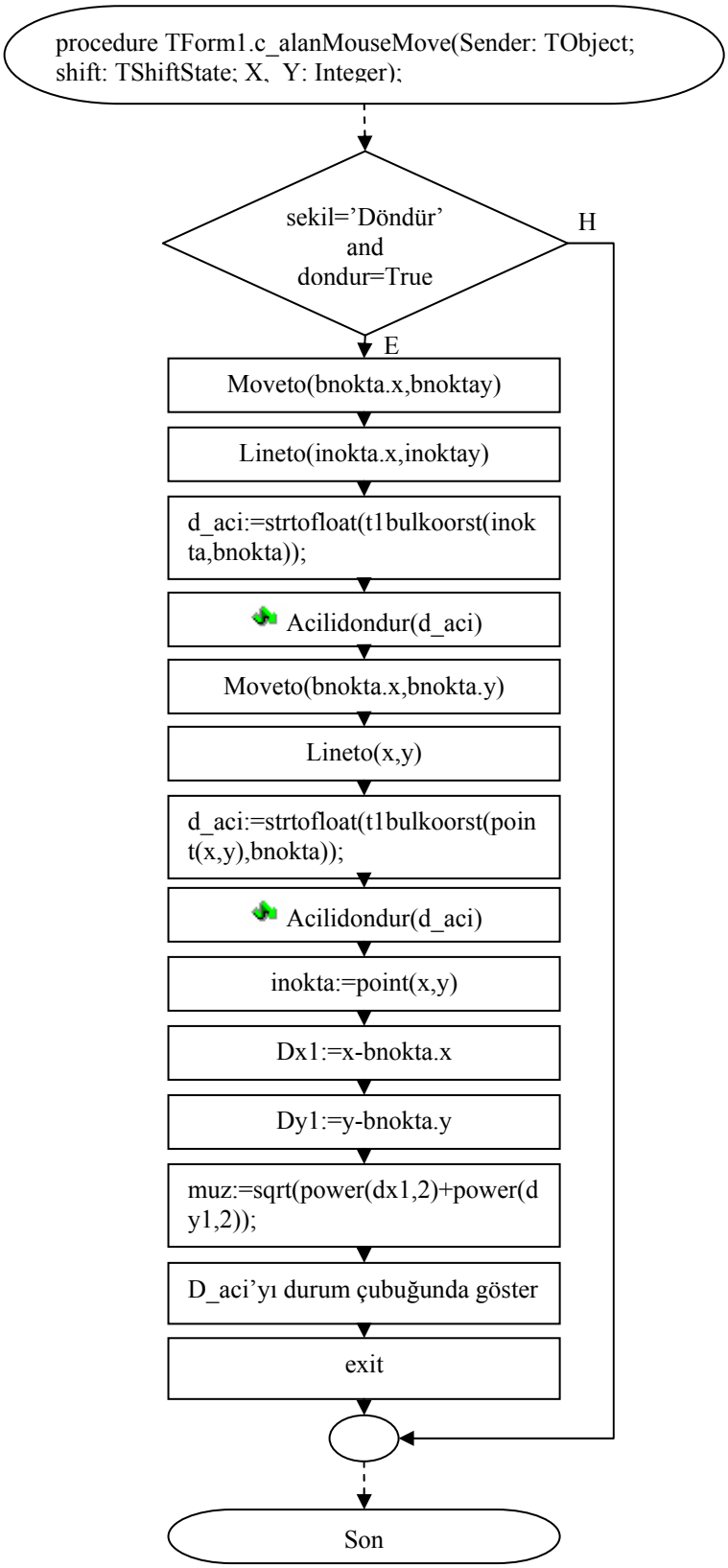


Şekil 4.28 Serbest çevir komutu seçildikten fare tuşuna basıldığındaki akış çizelgesi

Şekil 4.28'de serbest çevir komutu seçildikten sonra farenin çizim alanında tıklanması ile oluşan akış şeması gösterilmiştir. Kutu içerisine alınan kısım ilk tıklama ile çalışan kısımdır. Bu kısımda mantıksal olarak tanımlanmış olan dondur değişkenine doğru değeri aktarılmaktadır. Bu değer fare çizim alanında hareket ettirildiğinde döndürme işleminin başladığını göstermek için kullanılmaktadır. Sonraki komutalarda da döndürme çizgisinin gösterimine başlanmış ve koordinat bilgileri diğer değişkenlere aktarılmıştır.

Döndürme işlemi yapılırken ikinci defa fare tuşuna basılması döndürme işleminin bitirilmesini sağlamaktadır. Burada öncelikle döndürme işleminin bittiğinin anlaşılması için dondur değişkenine yanlış değeri aktarılmaktadır. Sonra döndürme çizgisi silinmektedir. Nesnelerin görsel olarak yapılan döndürme işlemi sonucunda oluşan yeni değerleri aasonuclariaktar alt programı kullanılarak güncellenmektedir. Aasonucalariaktar alt programı dönme işlemi sonucunda elde edilen yeni değerleri döndürülen nesnelere aktarmaktadır. Döndürme işlemi tamamlandıktan sonra seç butonunun seçilmesi sağlanmaktadır. Seçili bir grup olup olmadığı test edilerek varsa grup işaretlerinin yeniden çizilmesi sağlanmaktadır. Son olarak yeniden çiz komutu çalıştırılarak bütün nesnelerin yeniden çizilmesi sağlanmaktadır.

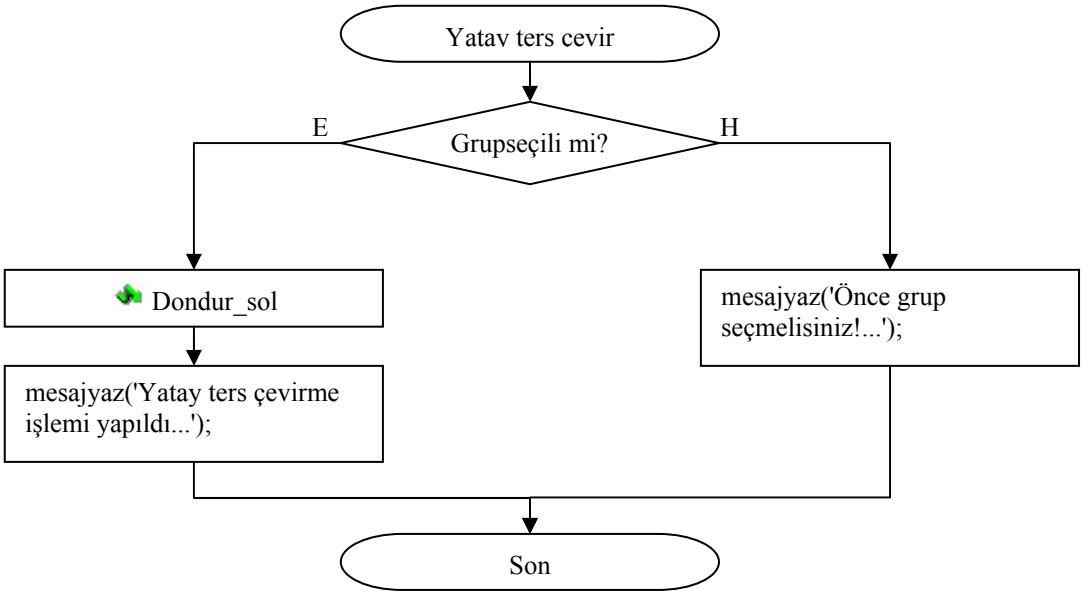
Şekil 4.29'da döndür komutu seçilip çizim alanında fare tuşuna basıldıktan sonra fare çizim alanında hareket ederken yapılan işlemlere ait akış şeması gösterilmiştir. Burada döndürme çizgileri ekranda gösterilmesi sağlanmaktadır ve dönme işleminin tarama yöntemiyle yapılması sağlanmaktadır. Bu amaçla öncelikle eski değerlere göre gösterilen şekiller acilidondur komutunun birinci çalışmasında silinir ve yeni x,y değerlerine göre hesaplanan açıya göre acilidondur komutu yeniden çalıştırılarak oluşan yeni açıya göre nesneler ekranda gösterilmektedir. Ve açı durum çubuğunda gösterilmektedir.



Şekil 4.29 Serbest çevir komutu seçildikten fare çizim alanında hareket ettirildiğindeki akış çizelgesi

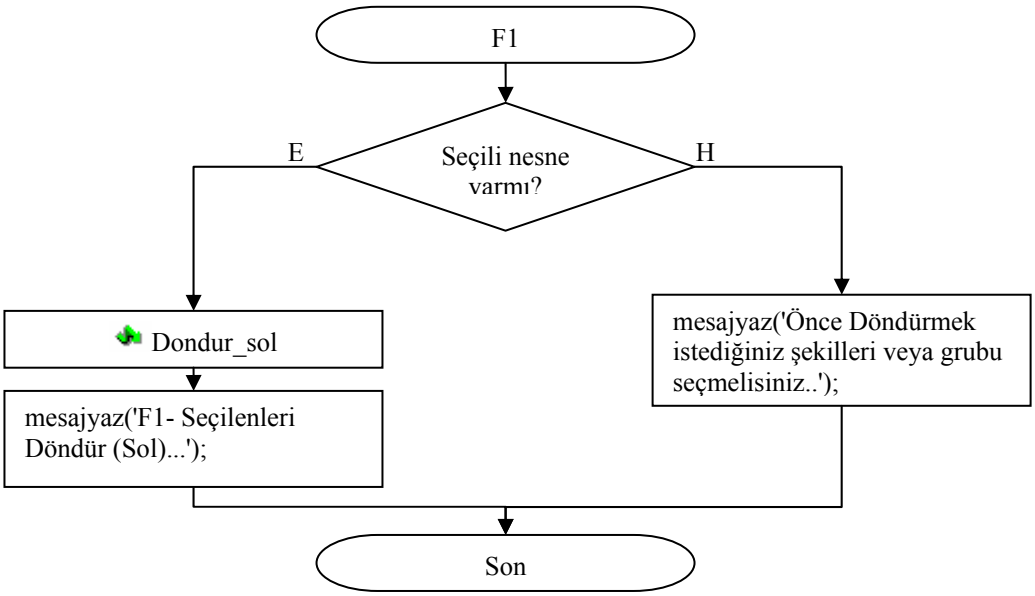
4.5.2. Yatay ters çevir komutu

Yatay ters çevir komutu komut menüsünden seçilerek ve F1 kısa yol ile tuşu ile seçilebilir. Bu komut, komut menüsünden seçildiğinde seçili bir grup varsa işlem yapmaktadır. F1 kısa yol tuşu ile işlem yapıldığında ise seçili nesneler üzerinde işlem yapmaktadır grup seçili olması şart değildir. Bu komut, komut menüsünden seçildiğinde seçili grup olup olmadığı test edilerek var ise dondur_sol alt programının çalıştırılması sağlanır. Bu komut, komut menüsünden seçildiğinde yapılan işlemlere ait akış çizelgesi Şekil 4.30'da gösterilmiştir.



Şekil 4.30 Yatay ters çevir komutu akış çizelgesi

Şekilden de görüldüğü gibi döndürme işlemini yapılmasını sağlayan dondur_sol alt programıdır. Yatay ters çevir komutu F1 kısa yol tuşu kullanılarak seçilirse yapılan işlemlere ait akış çizelgesi Şekil 4.31'deki gibidir. F1 ile döndürme işlemi yapılırken seçili nesne olup olmadığı test edilmektedir. Seçili nesne veya nesneler var ise bu nesneler döndürülmektedir.



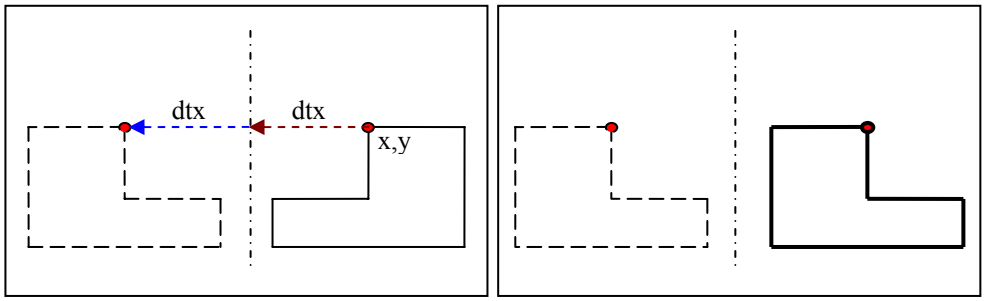
Şekil 4.31 F1 komutu ile yatay ters çevir işlemi akış çizelgesi

Öncelikle secilivarmi alt programı çalıştırılmaktadır. Bu alt program çizilmiş olan hafizadaki bütün nesneleri tarayarak seçili olan olup olmadığını kontrol eder ve geriye mantıksal bir değer döndürür. Seçili var ise doğru yok ise yanlış değerini geri döndüren bir fonksiyondur.

Birçok akış çizelgelerinde yer alan mesajyaz komutu programın alt tarafında yer alan komut kutusu ekranında kullanıcı ile etkileşimi sağlamak için yapılan işlem ile ilgili kullanıcıya bilgi vermektedir. Komut menüsünden veya F1 tuşu ile yapılan yatay ters çevirme işleminde dondur_sol alt programı çalıştırılmaktadır.

Döndür sol alt programı seçilen nesneleri nesnelerin solunda bulunduğu düşünülen dikey bir çizgiye göre aynalamaktadır. Aynalama işleminden sonra seçili nesneler tekrar eski konumuna taşınmaktadır. Aynalama işleminin benzetimi Şekil 4.32'deki gibidir. Şekil 4.32.a.'da bir poligonun referans çizgisine göre aynalanması ve yeni koordinatların nasıl hesaplandığı bir x,y noktası için gösterilmiştir. Bütün noktalar benzer şekilde bu çizgiye göre yeniden hesaplanır ve ilgili nesnelerin koordinatları ve gerekli diğer değerleri hesaplanır. Şekil 4.32.b.'de aynalama işlemi yapıldıktan sonra oluşan şeklin yeniden eski koordinatlarına taşınması gösterilmiştir.

Yani aynalanan şekil tekrar eski yerine taşınmaktadır. Taşıma işlemi aynalama işleminin sonunda yapılmaktadır. Aynalanan nesne bir grup ise grubu çerçeveleyen sol kenar çizgisine göre aynalama yapılmakta ve tekrar gruptasikoor(x,y)[1] alt programı aracılığı ile eski konumuna grup taşınarak tekrar seçili duruma getirilmektedir. Eğer seçilen nesne veya nesneler aynalanmışsa(grup değilse) döndür sol alt programının başında çalıştırılan seckutubul[2] alt programı seçili nesneleri kapsayan kutuyu bularak bu kutunun merkezinin koordinatlarını seckutum değişkenine aktarmaktadır. Dolayısıyla seçili nesneler aynalandığında nesne merkezdeki bir dikey çizgiye göre aynalandığı için nesnenin yeri değişmemektedir. Bu durumda tekrar seçili nesnelerin taşınmasına gerek kalmamaktadır. Yani şekil 4.32'de gösterilen aynalama çizgisi nesnenin solunda değil merkezinde olacaktır.



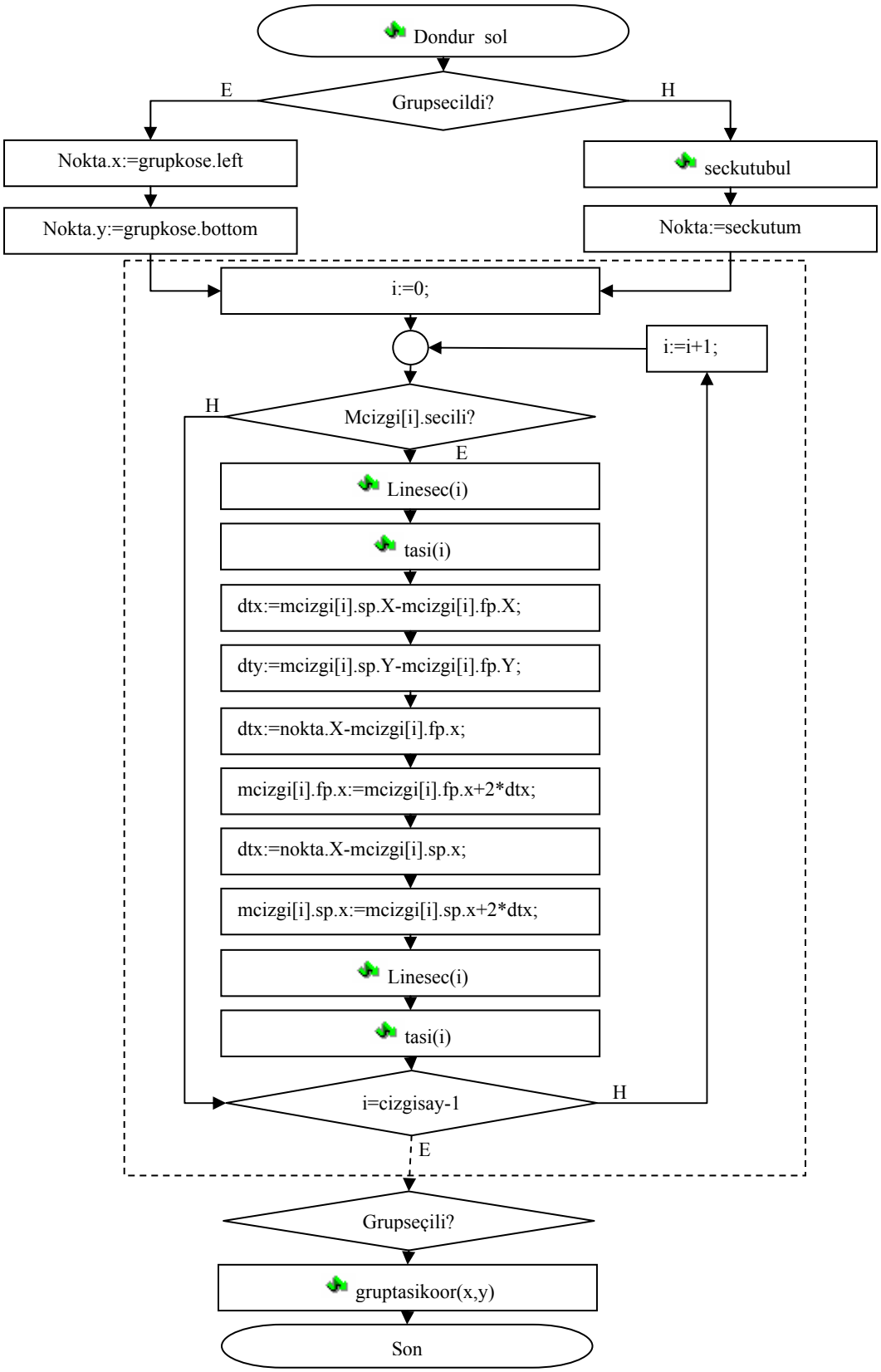
a. Birinci işlem b. İkinci işlem

Şekil 4.32 Yatay ters çevir komutu işlevi

Dondur sol alt programının mantığı ile ilgili verilen açıklamalardan sonra bu alt programın akış çizelgesinden bir kısmı Şekil 4.33'deki gibidir. Bu şekilde sadece çizgi için aynalamanın nasıl olduğu akış çizelgesinde gösterilmiştir. Diğer nesneler için yapılan işlemler de gösterilen içlemlerin aynısıdır. Ancak her nesne için tutulan koordinatlar farklıdır ve bu koordinatlar yeniden hesaplanmaktadır. Örneğin daire için sadece merkez koordinatlarının değiştirilmesi yeterlidir. Poligon için tüm noktalarının çizgideki hesaplanması gerekir. Yay için merkez, başlangıç ve bitiş koordinatları yeniden hesaplanır. Yayı oluşturan açılar da yeniden hesaplanmaktadır.

[1] Seçili grubu ve içindeki nesneleri parametre olarak gönderilen x,y değerleri kadar taşınmasını sağlar.
[2] Seçili nesneleri kapsayan dörtgeni bul

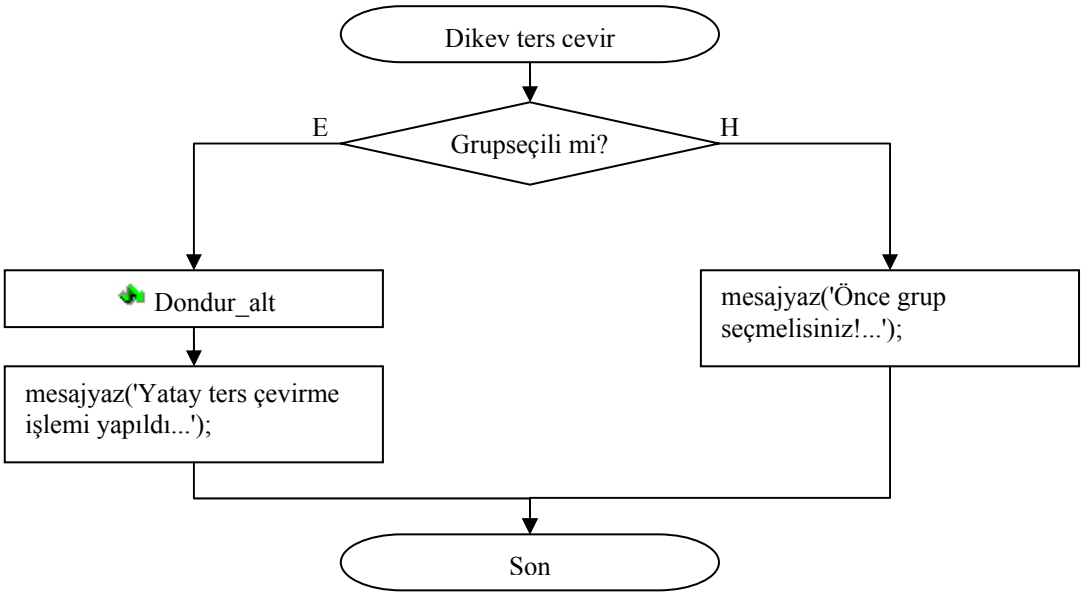
Yazı için sadece konumu belirten sol üst köşesi ve açısı yeniden hesaplanmaktadır. Bu işlemlerden sonrada yatay çevrilen bir grup ise gruptasikoor(x,y) alt programı kullanılarak ilk konumuna taşınmaktadır.



Şekil 4.33 Döndür sol alt programı akış çizelgesi

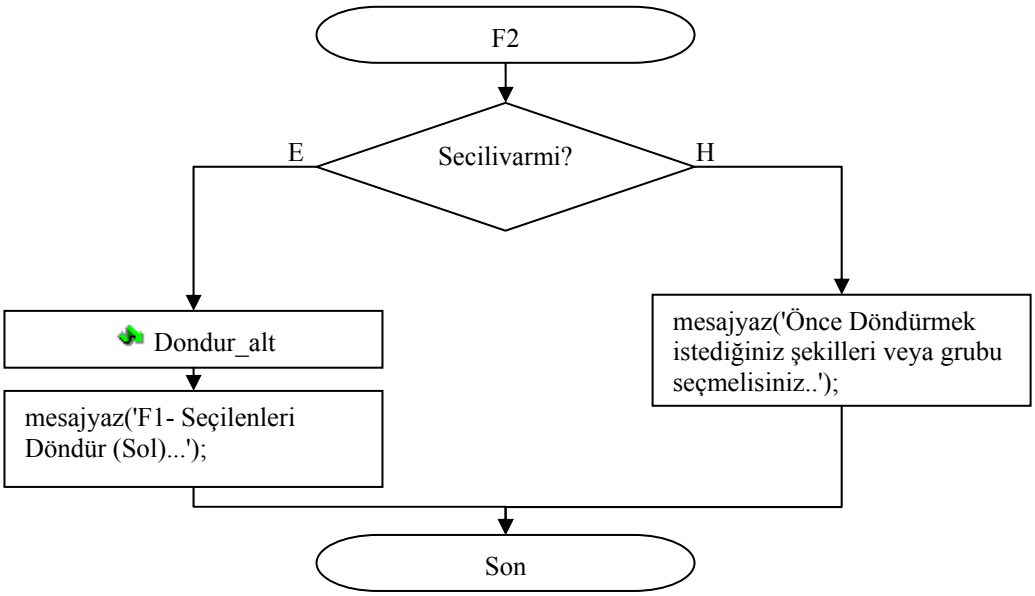
4.5.3. Dikey ters çevir komutu

Komut menüsünden seçilerek veya F2 kısa yol tuşları bu komutun çalışması sağlanabilir. Komut menüsünden seçildiğinde seçili olan bir grup var ise işlem yapmaktadır ve seçili gruba ait tüm nesnelerin yatay bir çizgiye göre aynalanmasını sağlamaktadır. Bu yatay çizgi grup sınırlarını gösteren çizginin alt çizgisidir. F2 kısa yol tuşu kullanılarak dikey ters çevir komutu çalıştığında ise seçili bir grup olması şart değildir seçili bir nesne olup olmadığı test edilir ve seçili nesneler merkezlerinden geçtiği düşünülen bir çizgiye göre aynalanır. Şekil 4.34'de komut menüsünden dikey ters çevir komutu seçildiğinde yapılan işlemlere ait akış çizelgesi gösterilmiştir.



Şekil 4.34 Komut/Dikey ters çevir komutu akış çizelgesi

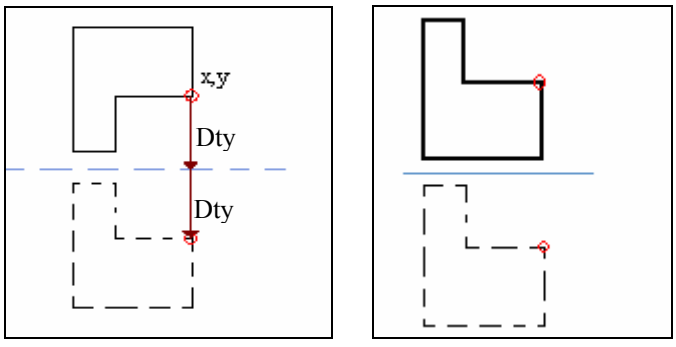
Şekilden de görüldüğü gibi yatay ters çevirme işleminin yapılmasını sağlayan alt program Dondur_alt alt programıdır. Dikey ters çevir işlemi F2 kısa yol tuşu ile yapıldığında gerçekleşen işlemlere ait akış çizelgesi Şekil 4.35'de gösterilmiştir.



Şekil 4.35 F2 komutu ile dikey ters çevir işlemi akış çizelgesi

Dikey ters çevir komutu F2 kısa yol tuşu ile yapıldığında da dondur_alt komutu çalıştırılmaktadır. Tek fark burada seçili bir grup olması yerine bir nesne veya nesnelerin olması yeterlidir. Dondur_alt komutu bir alt programdır.

Dikey ters çevirme işleminin nasıl yapıldığı Şekil 4.36'daki gibidir.

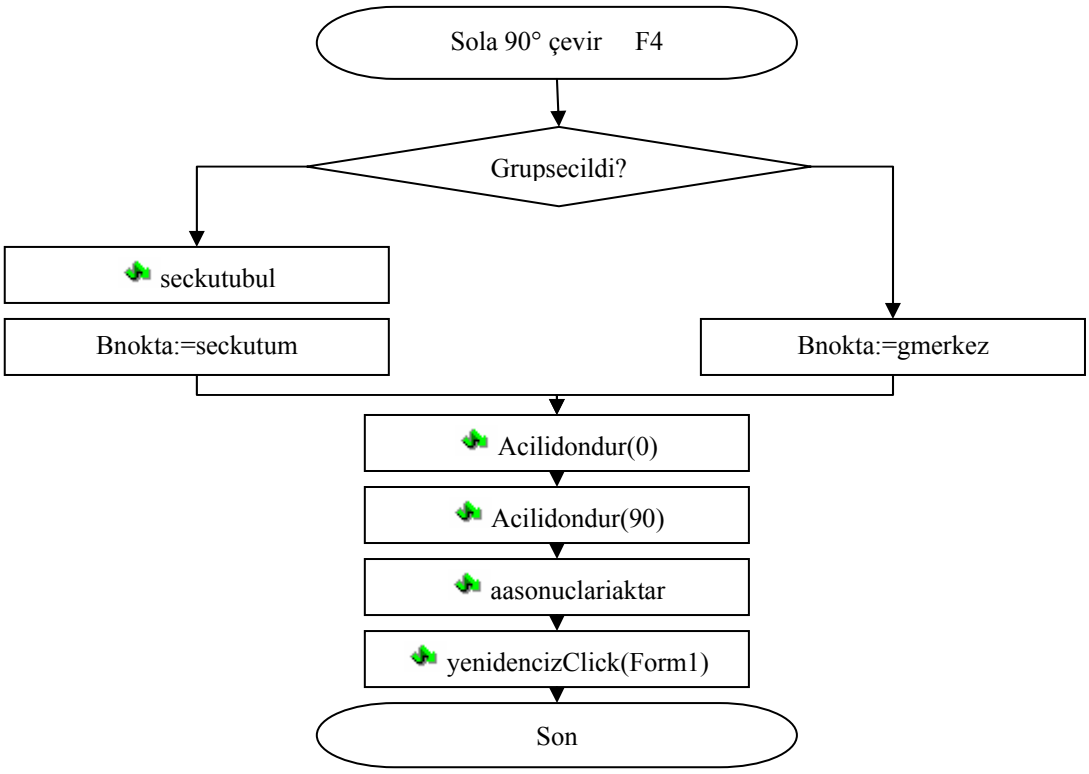


a. Birinci işlem b. İkinci işlem

Şekil 4.36 Dikey ters çevir komutu işlevi

Dikey ters çevir komutu akış çizelgesi de yatay ters çevir komutu akış çizelgesine benzer yapıdadır. Burada bir grup döndürüldüğünde grup alt çizgisi referans olarak alınmaktadır ve nesneler alt çizgiye göre aynalanır. Koordinatların nasıl değiştiği Şekil 4.36.a.'da görülmektedir. Bir x,y koordinatının altta bulunan çizgiye olan dikey uzaklığı bulunmuş bu dty ile gösterilmiştir. aynalama için dty iki ile çarpılarak y'ye eklenmiştir. Yani yeni koordinat x,y+2*dty olmuştur. Bütün koordinat noktaları için aynı işlem yapılarak nesnelerin ayna görüntüsü alınmış olmaktadır. Ayanalam işleminden sonra Şekil 4.36.b.'de kesikli çizgiler ile gösterilen şekil elde edilir ve nesneler tekrar gruptasikoor (x,y) alt programı ile eski konumuna getirilir.

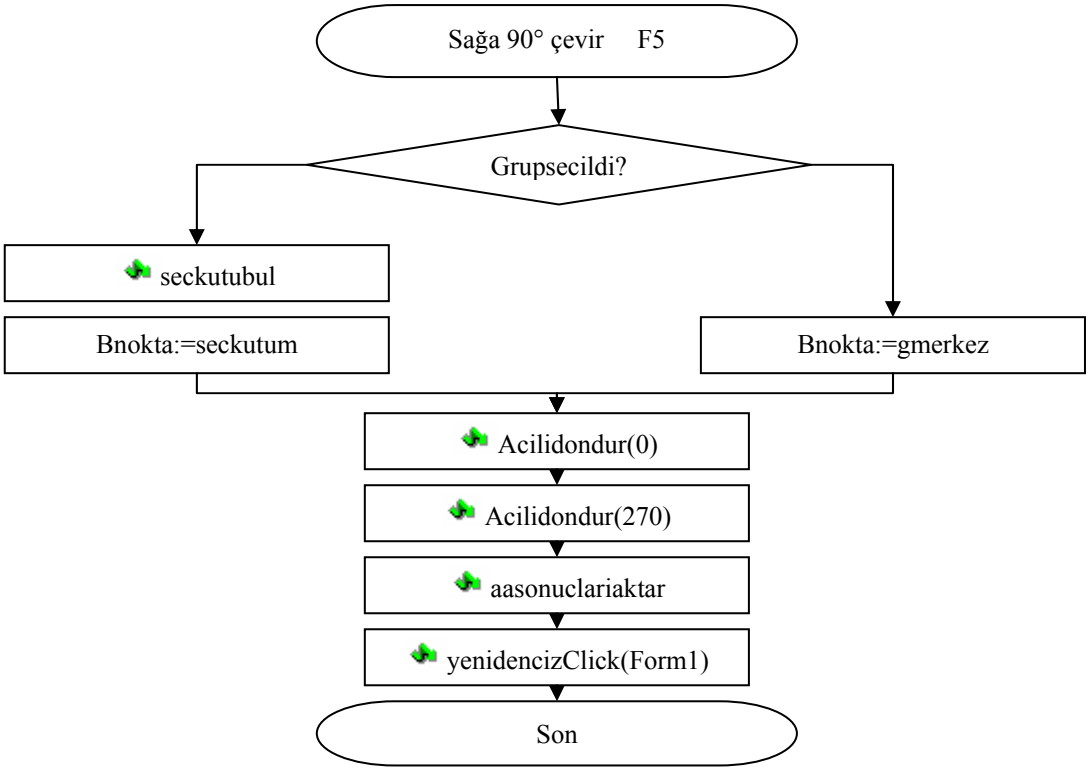
4.5.4. Sola 90° çevir komutu



Şekil 4.37 Sola 90° çevir komutu akış çizelgesi

Sola 90° çevir komutunun akış çizelgesi Şekil 4.37'de gösterilmiştir. Bu komut çalıştığında önce seçili grup olup olmadığı test edilir eğer seçili grup var ise referans noktası olarak grubun merkezi alınmaktadır. Seçili grup yok ise seckutubul alt programı çalıştırılarak seçili nesneleri kapsayan kutunun merkezi bulunur ve referans noktası olarak bu kutunun merkezi alınır. Referans noktası belirlendikten sonra acilidondur komutu çalıştırılarak seçili nesnelerin 90° döndürülmesi sağlanır. Acilidondur komutu iki defa çalıştırılmaktadır. Birincide seçili nesneler silinmekte ikincide ise döndürme işlemi sonucundaki yeni şekiller ekranda gösterilmektedir. Döndürme işleminin ekranda gösterilmesinin ardından oluşan yeni değerler aasonuclariaktar komutu ile hafızadaki ilgili nesnelere aktarılmaktadır. Acili döndür komutunun işlevi ve akış şeması açılı döndür konusunda anlatılacaktır.

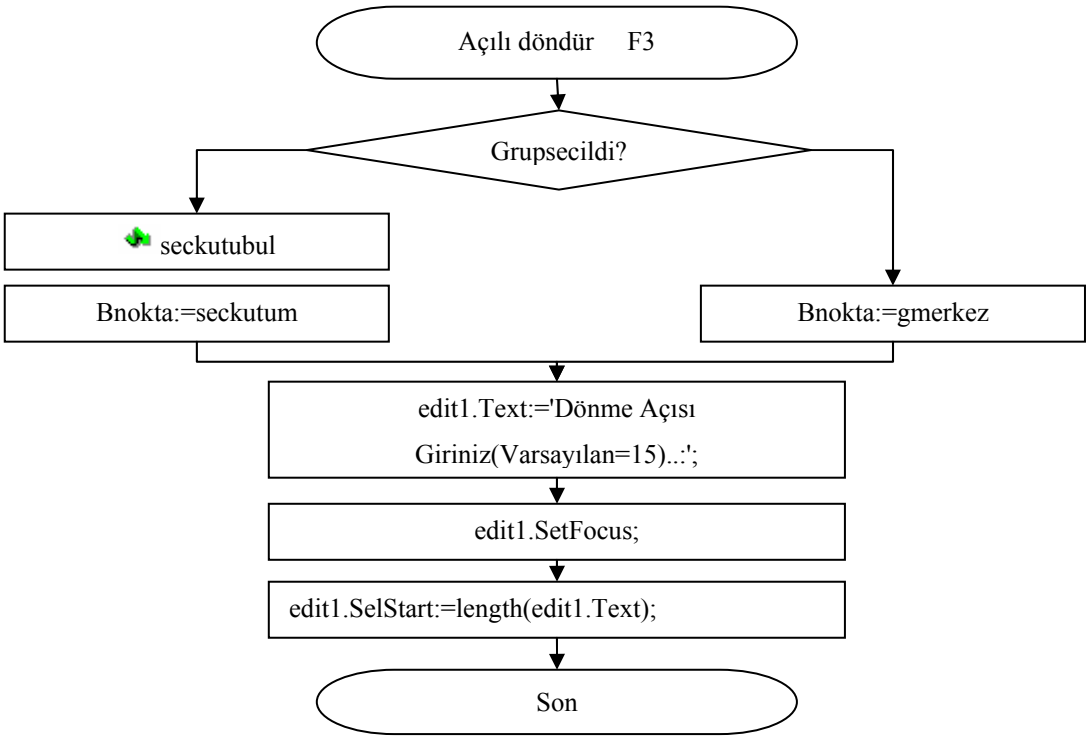
4.5.5. Sağa 90° çevir komutu



Şekil 4.38 Sağa 90° çevir komutu akış çizelgesi

Sağa 90° çevir komutu akış şeması sola çevir komutu ile aynı yapıya sahiptir. Tek fark açılı döndür komutunun ikinci çalışmasında döndürme işlemi 90° yerine 270° olarak yapılmaktadır. Bunun nedeni acilidondur komutunun saat yönünün tersinde döndürme yapmasındandır. Sağa 90° çevir komutu komut menüsünden veya F5 kısa yol tuşları ile çalıştırılsa da aynı işlemler yapılmaktadır. Şekil 4.38'de sağa 90° çevir komutunun akış çizelgesi gösterilmiştir.

4.5.6. Açılı döndür komutu



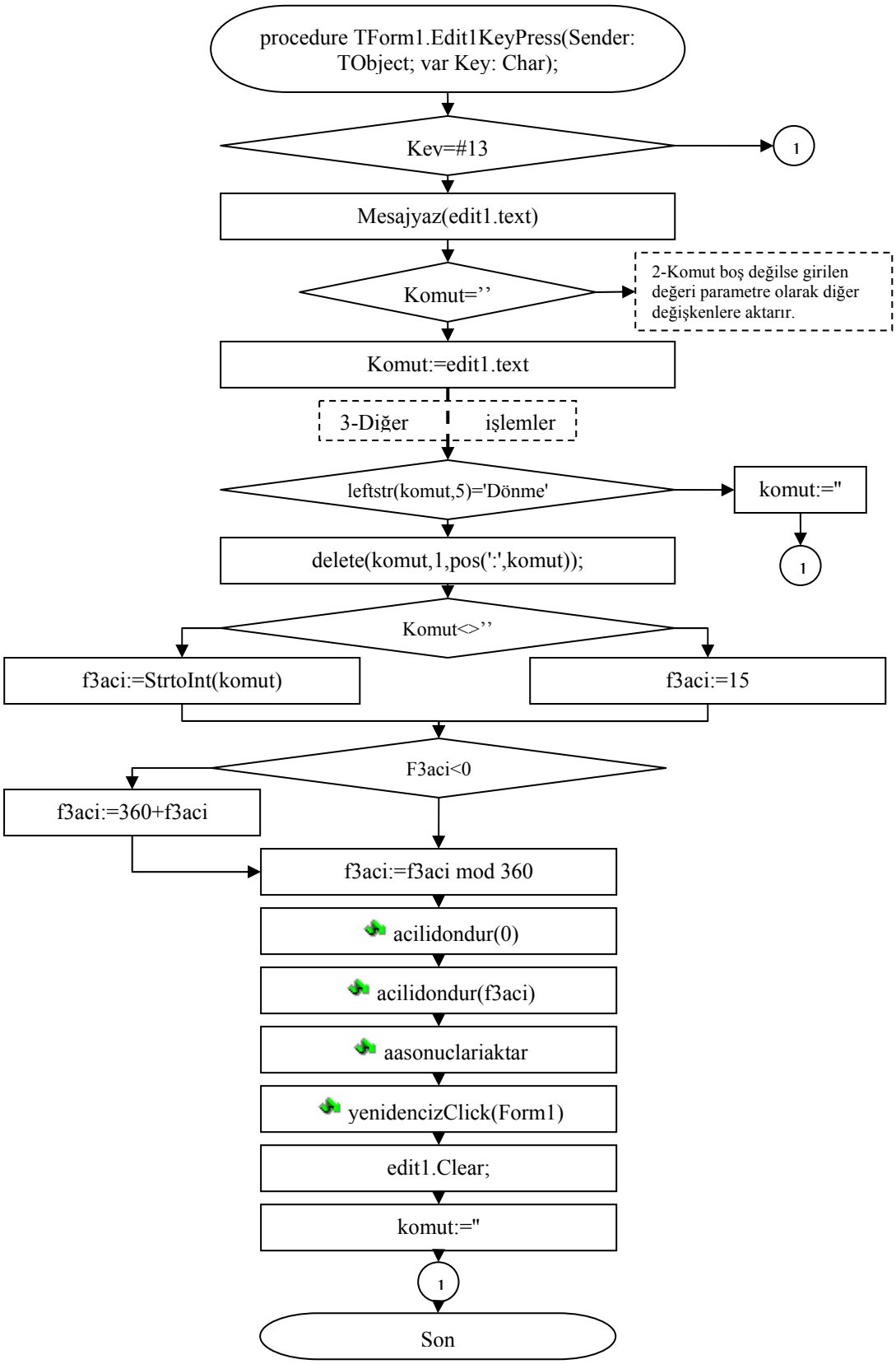
Şekil 4.39 Açılı döndür komutu akış çizelgesi(1)

Açılı döndür komutu seçildiğinde 90° döndürme komutlarında olduğu gibi önce bir referans noktası seçilmektedir. Referans noktası seçiminden sonra kullanıcının veri girebilmesi için program penceresinin altında yer alan komut edit kutusuna mesaj yazılmaktadır. Burada kullanıcıdan döndürmek istediği açıyı girmesi istenmektedir. Kullanıcı bir değer girmeden giriş(enter) tuşuna basarsa seçili nesneleri veya grubu 15° döndürmektedir.

Şekil 4.39'da gösterilen işlemlerin çalışmasından sonra program edit kutusu giriş olayına[1] dallanmaktadır. Edit kutusu giriş olayında kullanıcının klavyeden çizim yapabilmesi ve veri girebilmesi için gerekli kodlar yer almaktadır. Bu olayın açılı döndürme işlemi ile ilgili olarak çalışan kısmına ait akış çizelgesi Şekil 4.40'da gösterilmiştir. Edit kutusu giriş olayında çalışan serbest döndür komutu ilgili olan kısmın çalışması şöyledir;

Bu olayda çalışan alt program edit kutusunda giriş tuşuna basıldığında işlem yapmaktadır. Edit kutusunda giriş tuşuna basıldığında ilk olarak girilecek verinin komut olması nedeniyle komut değişkeni boş ise veri komut değişkenine aktarır. Açılı döndürme işleminde komut ve parametre tek bir satırda yapıldığı için sadece komut değişkeninin kullanılması yeterli olmuştur. (Klavye ile çizim yapılırken komuttan sonra girilen veriler sırayla parametre1, parametre2,..parametre4 değişkenlerine aktarılmaktadır. Bu işlemler akış çizelgesinde iki numaralı kutu ile gösterilmiştir). Edit kutusunda giriş tuşuna basıldığında bu kutudaki yazı komut değişkenine aktarılır. Komutun ne olduğu anlaşılması için bir dizi karşılaştırmadan geçirilmektedir(Bu işlemler 3 numaralı kutuda diğer işlemler olarak gösterilmiştir). Şekil 4.39'da edit kutusuna "'Dönme Açısı Giriniz(Varsayılan=15)..:" mesajı yazdırıldığı için kullanıcı giriş tuşuna bastığında komut değişkeninin ilke beş karakteri "Dönme" kelimesi olmaktadır. Bu nedenlede bu veri test edilerek komutun açılı döndür olduğu anlaşılmakta ve mesajın sonuna kullanıcının yazacağı değer alınarak f3aci değişkenine aktarılmıştır. Bu açı pozitif ise döndürme işlemi sola doğru yapılmaktadır. Negatif ise döndürme işlemi sağa doğru yapılmaktadır. Kullanıcının 360°'den büyük sayılar girmesi durumunda periyot 360 olduğu için gerçek açık girilen açının 360'a bölümünden kalan olarak aktarılmaktadır. Açı da okunduktan ve düzenlendikten sonra açılı döndür alt programı çalıştırılarak parametre olarak girilen açı alt programa aktarılmaktadır. Önce bu alt programa 0 değeri gönderilerek nesneler silinmekte sonra açı kadar döndürülmekte ve sonuçlar da sonuçları aktar alt programı çalıştırılarak nesneleri tutan değişkenlere aktarılmaktadır. Edit kutusu silinip komut değişkeninin içeriği boşaltıldıktan sonra alt program sonlanmaktadır.

[1] procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char); bu alt program edit kutusunda tuşa basıldığında çalışmaktadır.



Şekil 4.40 Açılı döndür komutu akış çizelgesi(2)

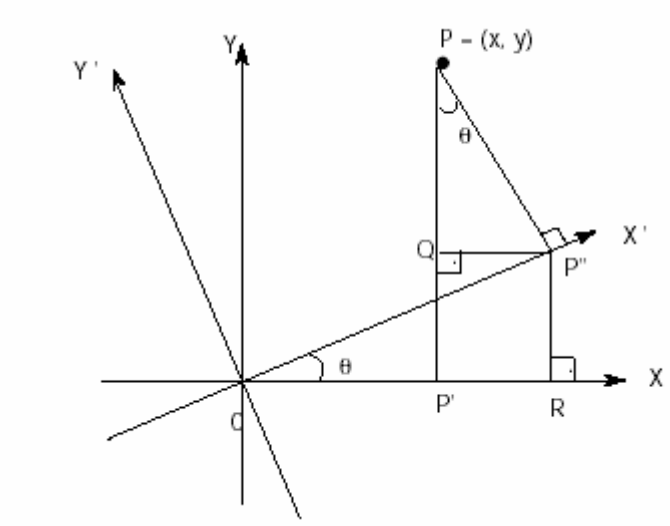
4.5.6.1 Açılı döndür alt programı

Açılı döndür alt programının düzlemde gösterimi Şekil 4.41'deki gibidir. Burada ilk koordinatları (x,y) olan bir P noktasının düzlemin pozitif yönde (saat yönünün tersi) θ açısı döndürülmesi sonucunda oluşacak (x',y') noktalarının koordinatlarının formülü şöyledir;

$$x' = x \cos\theta + y \sin\theta$$

$$y' = -x \sin\theta + y \cos\theta$$

Açılı döndürme işleminde bu formüller kullanılarak seçili olan bütün nesnelerin koordinatları yeniden düzenlenmektedir.



Şekil 4.41 Açılı döndür komutunun düzlemde gösterimi

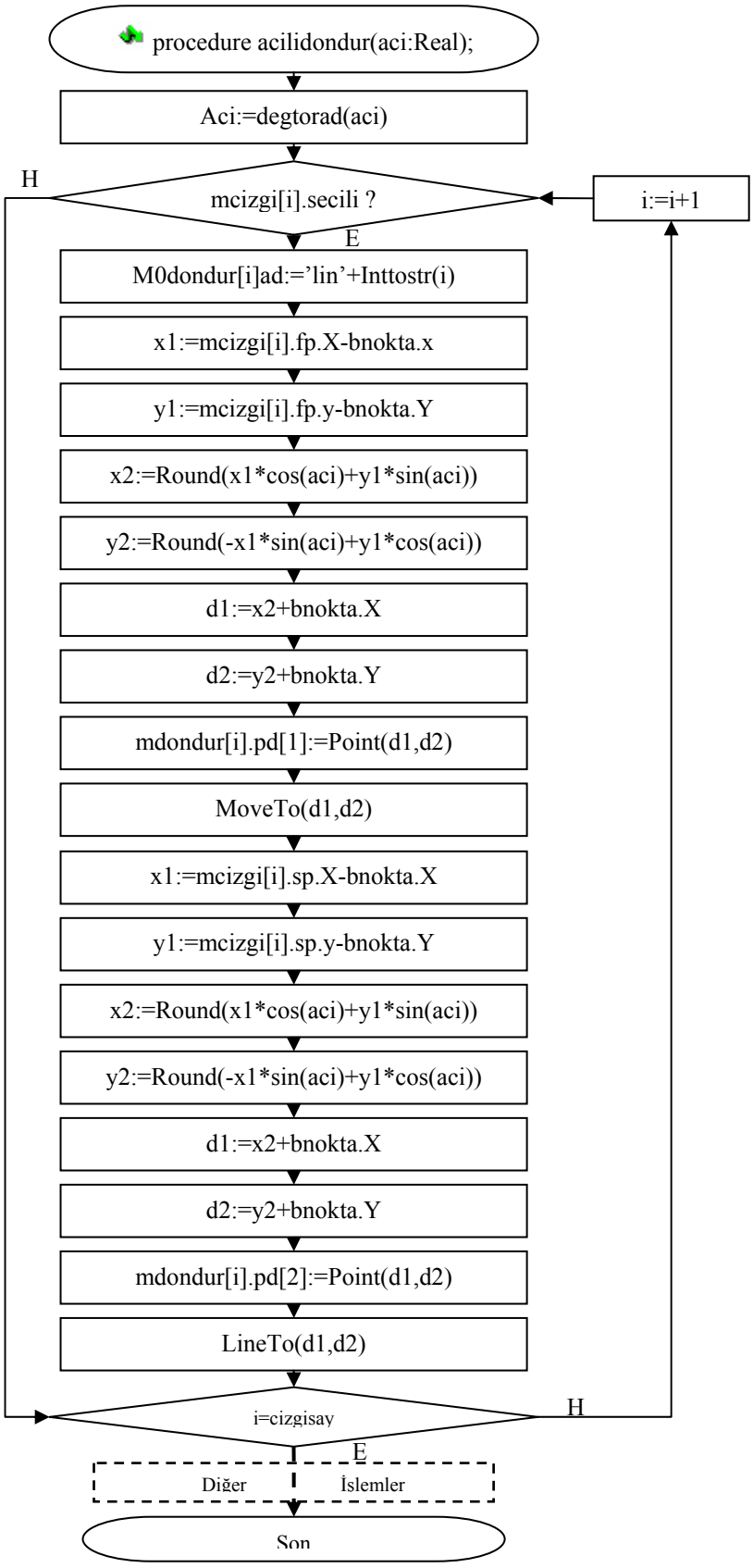
Şekil 4.42'de açılı döndür alt programının çizgiye ait olan kısımları akış çizelgesinde gösterilmiştir. Diğer nesneler içinde aynı işlemler yapılmaktadır. Burada sadece çizgi nesnesinin nasıl döndürüldüğü gösterilecektir. Çizelgede gösterilen akış çizelgesinde yapılan işlemler şunlardır;

Çizginin seçili olup olmadığı test edilir. Eğer seçili ise kayıt tipinde tanımlanmış olan mdondur dizisinin ada alanına komut ve numarası aktarılır. Daha sonra başlangıç koordinatları dönüş açısına göre hesaplanarak mdondur dizisinin pd[1] alanına aktarılır. Aynı işlemler bitiş koordinatları içinde yapılarak pd[2] alanına aktarılır. İ değişkeninin hafızadaki çizgi sayısını gösteren cizgisay değişkenine eşit olup olmadığı test edilir. Eşit değilse i değişkeni bir artırılır. Aynı işlemler çizgi2 içinde tekrarlanır. İ değişkeni cizgisay değişkenine eşit olduğunda seçili tüm çizgiler açıya göre döndürülmüş olur. Ve yeni değerleri mdondur değişkeninde tutulmaktadır. Hafızadaki seçili bütün nesneler için hesaplamalar yapılarak mdondur değişkenine aktarılır.

Mdondur değişkeni aşağıdaki gibi tanımlanmıştır.

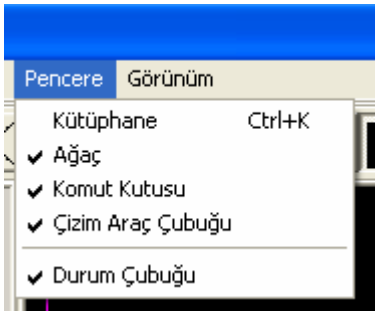
```
Type
mmdondur=record
        ad:String[10];
        pd:Array[1..20] of Tpoint;
        r1,r2,r3:Real;
   end;
Var
mdondur:array[0..1000] of mmdondur;
```

Komut ve numara mdondur değişkeninin ad alanına aktarılmaktadır. Nesneye ait koordinat bilgileri pd alanına aktarılır. Nesne ile ilgili gerçek tipte değerler varsa bunlar da r1,r2,r3 alanlarına aktarılmaktadır. Çizgi için ad alanına ‘lin’+nesneno, pd alanlarına başlangıç ve bitiş koordinatları aktarılmaktadır.



Şekil 4.42 Açılı döndür alt programı alış çizelgesi

4.6. Pencere Menüsü

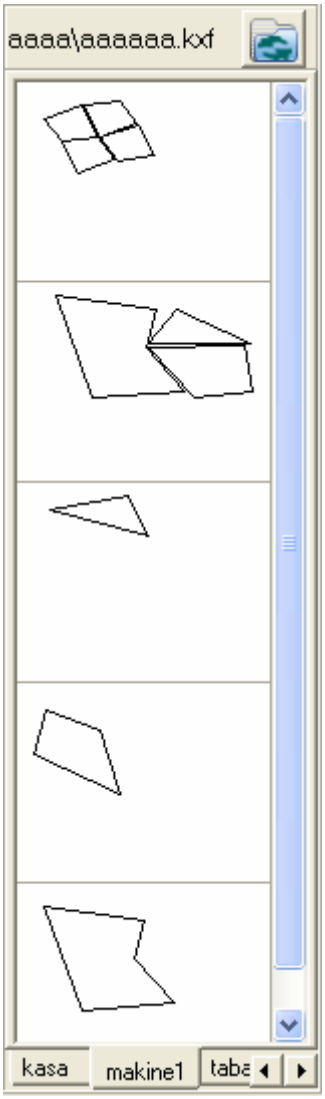


Şekil 4.43 Pencere menüsü ekran görünümü

Pencere menüsünün ekran görünümü Şekil 4.43'deki gibidir. Bu menüden program penceresinde yer alan Şekil 4.2'de gösterilen pencereler gizlenip görünür yapılabilmektedir. Varsayılan olarak kütüphane penceresi gizli diğerleri görünür durumdadır. Kullanıcı bunlardan istediğini gizleyebilir veya görünür hale getirebilir. Bu menüde yer alan komutlar ve akış çizelgeleri şöyledir.

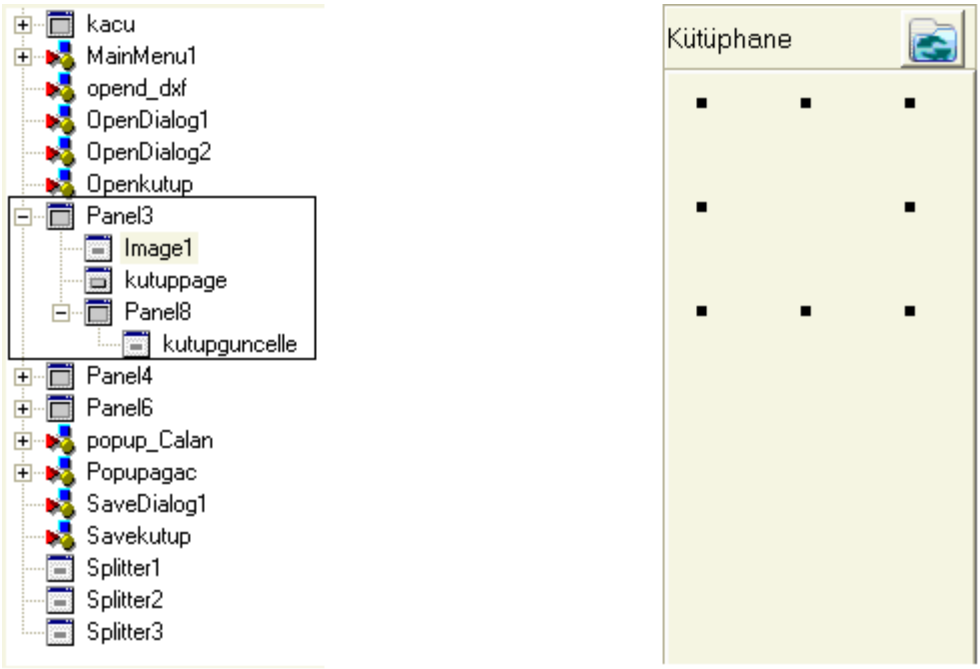
4.6.1. Kütüphane

Pencere menüsü ekran görünümü Şekil 4.44'de gösterilmiştir. Kütüphane penceresi panel nesnesi üzerine yerleştirilmiştir. Kütüphane penceresinde programın bulunduğu kütüphane klasöründe yer alan klasörler bir sayfa olarak ve bu klasörlerde yer alan dosyalar da barındırdığı çizimler ön izleme olarak ekranda gösterilmektedir. Kütüphane penceresinin çalışması 3.6. Kütüphane yapısı konusunda ayrıntılı olarak anlatılmıştı.



Şekil 4.44 Pencere menüsü ekran görünümü

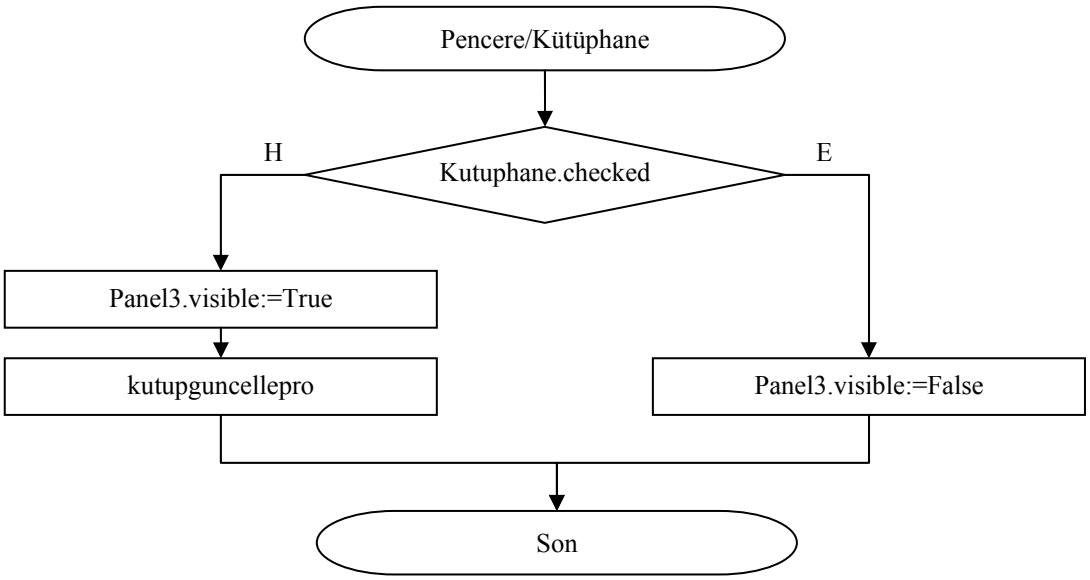
Form üzerinde hali hazırda kütüphane için oluşturulmuş ve forma eklenmiş olan nesneler Şekil 4.45.a.'da ağaç yapısında gösterilmiştir. Gerekli nesneler panel3 nesnesinin içine eklenmiştir. Bu nesnelerin tasarım anındaki ekran görünümü de Şekil 4.45.b.'de gösterilmiştir.



a.Kütüphane penceresinde yer alan nesneler b.Kütüphane penceresi ekran görünümü

Şekil 4.45 Kütüphane penceresinde tasarım anında yer alan nesneler ve ekran görünümü

Pencere menüsünden kütüphane seçildiğinde çalışan kodlara ait akış çizelgesi Şekil 4.46'da gösterilmiştir. Burada pencere menüsünden kütüphane seçeneğinin işaretli olup olmadığı test edilmektedir. İşaretli ise kütüphane penceresini bulunduran panel3 nesnesi gizlenmekte ve olay sonlandırılmaktadır. İşaretli değilse kütüphane penceresini bulunduran panel3 görünür hale getirilmekte ve kutupguncellepro alt programı çalıştırılarak penceredeki nesnelerin güncellenmesi sağlanmaktadır.



Şekil 4.46 Kütüphane komutu akış çizelgesi

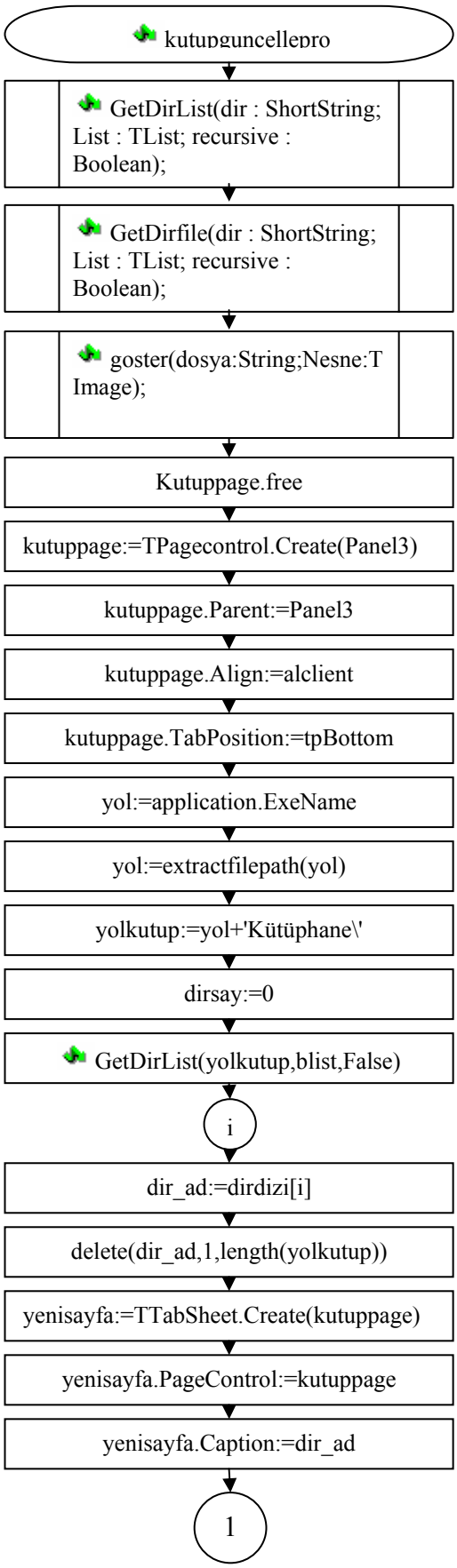
Kütüphane penceresinin oluşturulması ve kütüphaneye kaydedilen nesnelerin ekranda gösterilmesi işlemini yapan alt program kutupguncellepro alt programıdır. Bu alt program kendi içinde üç tane alt program içermektedir. Bunlar

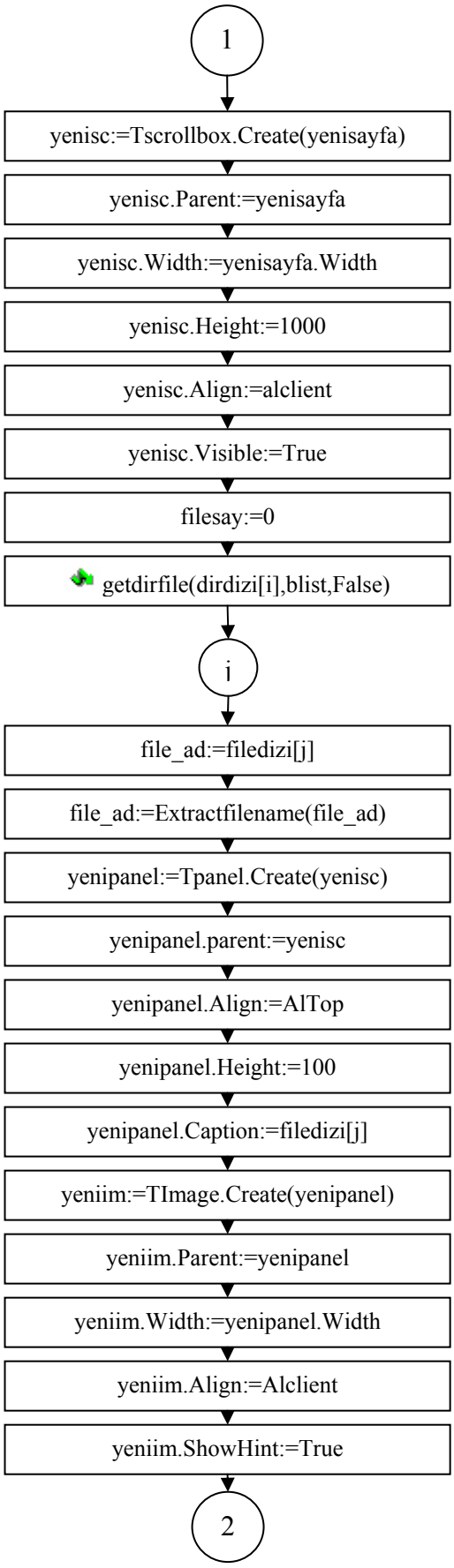
Getdirlist: Parametre olarak gönderilen klasördeki alt klasörleri tarar. Bulunan klasörlerin isimlerini metin tipinde dizi olarak tanımlanmış olan dirdizi değişkenine aktarır. Bulduğu klasörlerin sayısını da tam sayı tipindeki dirsay değişkenine aktarır.

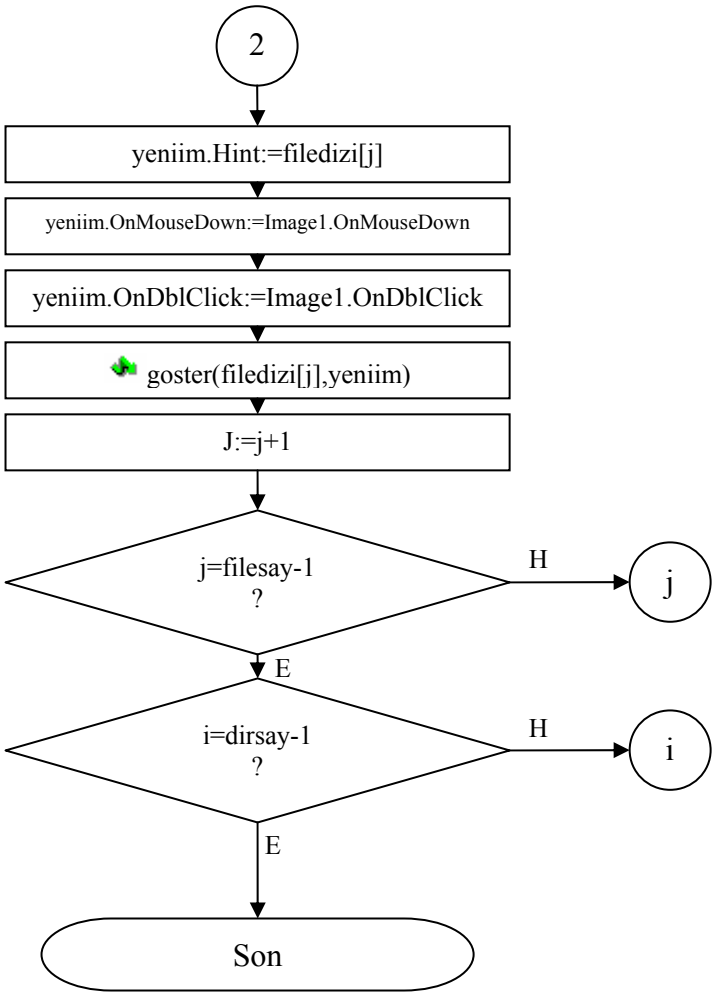
Getdirfile: Parametre olarak gönderilen klasördeki dosyaları bulur. Bulduğu dosyaların isimlerini metin tipinde dizi olarak tanımlanmış olan filedizi değişkenine aktarır. Dosya sayısını da tam sayı tipindeki filesay değişkenine aktarır.

Goster: Parametre olarak gönderilen dosyadaki çizimleri parametre olarak gönderilen nesne içerisinde çizerek ekranda görüntülenmesini sağlar. Bu alt program dosya menüsünde anlatılan aç komutu gibi çalışır.

Kütüphaneyi güncelle alt programının akış çizelgesi Şekil 4.47'deki gibidir.


kutupguncellepro
GetDirList(dir : ShortString; List : TList; recursive : Boolean);
GetDirfile(dir : ShortString; List : TList; recursive : Boolean);
goster(dosya:String;Nesne:TImage);
Kutuppage.free
kutuppage:=TPagecontrol.Create(Panel3)
kutuppage.Parent:=Panel3
kutuppage.Align:=alclient
kutuppage.TabPosition:=tpBottom
yol:=application.ExeName
yol:=extractfilepath(yol)
yolkutup:=yol+'Kütüphane\'
dirsay:=0
GetDirList(yolkutup,blist,False)
i
dir_ad:=dirdizi[i]
delete(dir_ad,1,length(yolkutup))
yenisayfa:=TTabSheet.Create(kutuppage)
yenisayfa.PageControl:=kutuppage
yenisayfa.Caption:=dir_ad
1


1
yenisc:=Tscrollbox.Create(yenisayfa)
yenisc.Parent:=yenisayfa
yenisc.Width:=yenisayfa.Width
yenisc.Height:=1000
yenisc.Align:=alclient
yenisc.Visible:=True
filesay:=0
getdirfile(dirdizi[i],blist,False)
i
file_ad:=filedizi[j]
file_ad:=Extractfilename(file_ad)
yenipanel:=Tpanel.Create(yenisc)
yenipanel.parent:=yenisc
yenipanel.Align:=AlTop
yenipanel.Height:=100
yenipanel.Caption:=filedizi[j]
yeniim:=TImage.Create(yenipanel)
yeniim.Parent:=yenipanel
yeniim.Width:=yenipanel.Width
yeniim.Align:=Alclient
yeniim.ShowHint:=True
2



Şekil 4.47 Kütüphaneyi güncelle komutu akış çizelgesi

Kutuppage isimli Tpagecontrol nesnesi kütüphaneyi güncelle alt programının başında silinerek yeniden oluşturulmaktadır. Oluşturulan nesnenin parent komutu ile panel3'ün içinde olması sağlanmıştır. Oluşturulan Tpagecontrol nesnesinin align komutu ile kullanılabilir tüm alanı kaplaması sağlanmıştır. Tabpozition komutu ile sayfa isimlerinin nesnenin altında görüntülenmesi sağlanmıştır. Getdirlist alt programı ile kütüphane klasöründe yer alan klasörler okunarak dirdizi değişkenine sayısı da dirsay değişkenine aktarılmıştır. Dirsay-1 kadar döngü kurularak bu döngü içinde aşağıdaki işlemlerin yapılması sağlanmıştır.

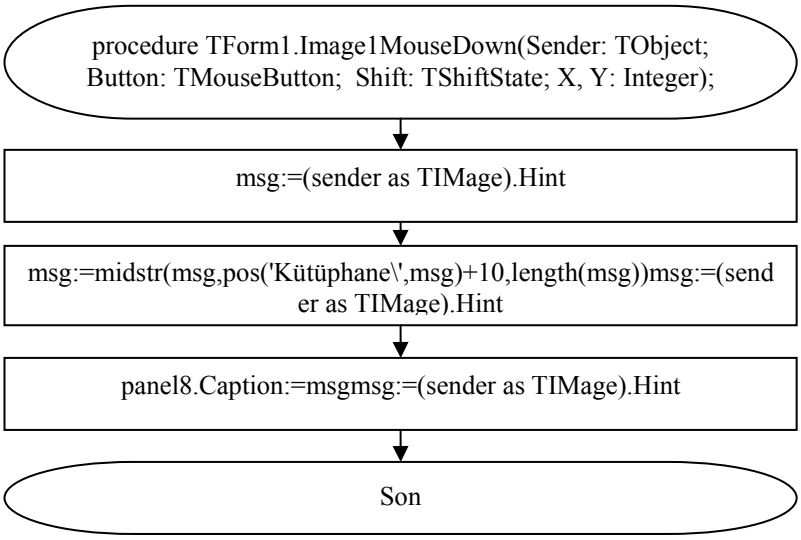
Ttabsheet nesnesine yeni bir sayfa eklenmiş, sayfanın başlığında klasörün ismi gösterilmiştir. Bu sayfanın içine Tscroolbox nesnesi eklenmiştir. Align komutu

ile kaydırma kutusunun yeni sayfanın tüm alanını kaplaması sağlanmış ve yüksekliği 1000 piksel olarak verilmiştir. Daha sonra getfilelist alt programı ile bulunulan klasördeki tüm dosyalar filedizi değişkenine sayısı da filesay değişkenine aktarılmıştır. Burada tekrar filesay-1 kadar bir döngü kurulmuştur. Bu döngü içinde de şu işlemler yer almaktadır;

Filedizi değişkenindeki dosya adlarından ilki alınır yol silinerek sadece dosya adının alınması sağlanır(Extractfilename(file_ad) komutu ile). Dosya adı file_ad değişkenine aktarılır. Daha önce oluşturulan kaydırma kutusu içine yeni bir panel oluşturulur. Panelin kaydırma kutusunun üst kısmında 100 piksel yüksekliğinde gösterilmesi sağlanır. Panelin başlığına dosya yolu ve adı aktarılır. Daha sonra bu panelin içine bir Timage nesnesi oluşturulur. Oluşturulan imaj nesnesinin panelin tamamını kaplaması sağlanır. İmaj nesnesinin showhint özelliği true yapılarak hint mesajına dosya adı aktarılır. Oluşturulan imaj nesnesinin fare tuşuna basılması ve çift tıklama olaylarının tasarım anında oluşturuluş olan image1 nesnesinin ilgili olaylarını kullanması sağlanır. Zaten tasarım anında image1 nesnesi bu amaçla oluşturulmuştur. Daha sonra Goster alt programı çalıştırılarak bulunulan çizim dosyasının oluşturulan yeni imaj nesnesi içinde gösterilmesi sağlanır.

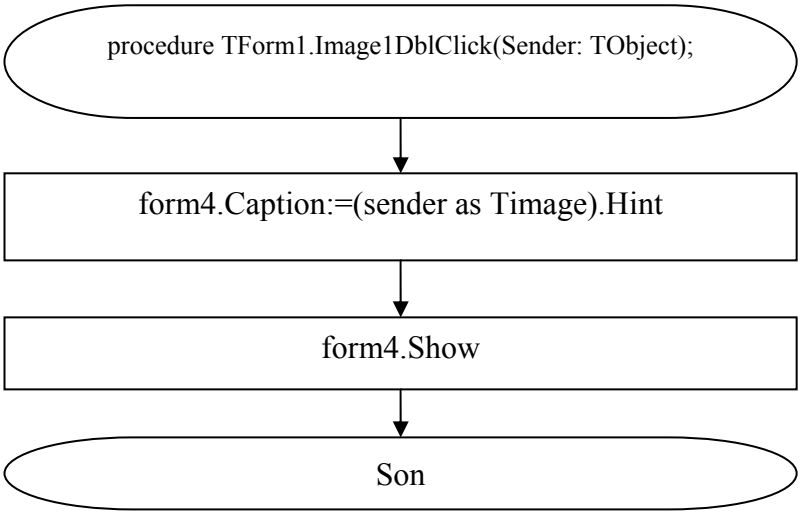
Kullanıcı programı kullanırken sayfalarda gösterilen bir çizim üzerine geldiğinde ilgili çizim dosyasının yolu ve adı işaretçinin altında açıklama olarak gösterilir. Kullanıcı nesne üzerine tıkladığında Mouse.down olayındaki kodlar çalıştırılır. Bu kodlar tüm gösterilen dosyalar için aynıdır ve image1 için yazılmış kodlardır. Bu kodlara ait akış çizelgesi Şekil 4.48'deki gibidir.

Bu akış çizelgesinde olayı çağıran nesne bir imaj ise bunun ipucu mesaj değişkenine aktarılmaktadır. Bu mesajda çizim dosyasının yolu ve adı vardır. Bu metin uzun olduğu için kütüphane'den sonraki kısmı alınarak panel8'de başlığı yapılmaktadır. Böylece kullanıcı seçtiği dosyanın klasörünü ve adını kütüphane penceresinin başında kütüphaneyi güncelle butonunun yanında görebilmektedir.



Şekil 4.48 İmaj nesnelerinde fare tuşuna basıldığına oluşan akış çizelgesi

Kullanıcı sayfalarda gösterilen bir çizim üzerinde çift tıklama yaptığında image1 nesnesi için yazılmış olan Image1DblClick olayındaki kodlar çalışır. Bu kodlar tüm gösterilen dosyalar için aynıdır. Bu olayda çalışan kodlara ait akış çizelgesi Şekil 4.49'da gösterilmiştir.

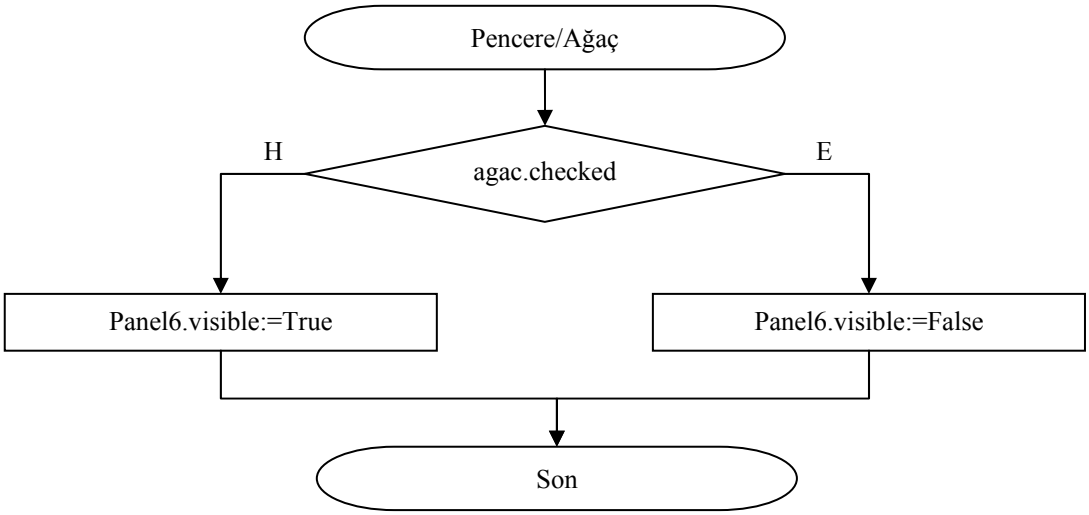


Şekil 4.49 İmaj nesnelerinde fare tuşuna iki kere basıldığına oluşan akış çizelgesi

Burada form4 devreye girmektedir. Form4'ün başlığı çift tıklanan dosyanın adı ve yolu yapıldıktan sonra form4'ün gösterilmesi sağlanmaktadır. Bu formda seçilen dosyadaki çizim ekranda gösterilir ve ana formdaki çizim alanına istenilen sayıda aktarılması sağlanır. Bu formun çalışması ve özellikleri konu 4.8'de ayrıntılı olarak anlatılacaktır.

4.6.2. Ağaç

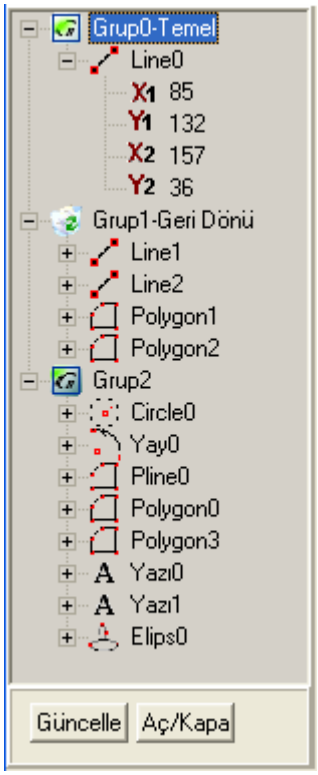
Pencere menüsünden Ağaç seçildiğinde çalışan kodlara ait akış şeması Şekil 4.50'deki gibidir.



Şekil 4.50 Pencere/Ağaç komutu akış çizelgesi

Bu komut seçildiğinde eğer onaylanmış durumda ise ağacın gösterildiği panel6 gizli duruma getirilmektedir. Onaylanmamış durumda ise panel6 görünür yapılarak ağacın görüntülenmesi sağlanmaktadır. Onaylama işlemi nesnenin özelliğinden dolayı otomatik olarak yapılmaktadır. Bu nedenle onay işaretinin konulması ya da kaldırılması kod yazmaya gerek yoktur.

Ağaç penceresinin ekran görünümü Şekil 4.51'deki gibidir.



Şekil 4.51 Ağaç penceresi ekran görünümü

Ağaç penceresinde şunlar gösterilir;

- ✓ Ağaç penceresi çizilen nesnelerin ait olduğu gruba göre gösterilmesini sağlar.
- ✓ Çizilen nesnelere ait bilgileri gösterir.(koordinat, uzunluk, yarı çap vb.)
- ✓ Kütüphaneden ekleme yapıldığında veya dxf dosyası alındığında grup adının yanında "-" işaretinden sonra dosya adı gösterilir.
- ✓ Gruplar nesneler ve özellikleri iki ikonla gösterilir. Seçildiğinde ve normal olmak üzere.

Ağaç penceresinden şu işlemler yapılabilmektedir;

- ✓ Grup ya da nesneler seçilebilir.
- ✓ Nesnelere ait bilgiler değiştirilebilir.
    - o Gösterilen bilgiler şunlardır.
        - ▪ Çizgi için başlangıç ve bitiş koordinatları.
        - ▪ Daire için merkez koordinatları ve yarıçap.
        - ▪ Yay için merkez koordinatları, yarıçapı başlangıç açısı ve yay açısı.
        - ▪ Ucu açık ve kapalı poligon için bütün noktalarının koordinatları.
        - ▪ Elips için bütün köşe koordinatları
        - ▪ Yazı için içerdiği metin, koordinatları ve açısı.
    - o Nesnelere ait bilgiler şu yöntemlerle değiştirilebilir.
        - ▪ Veri üzerine çift tıklanarak.
        - ▪ Veri üzerinde sağ tuşa tıklanarak açılan menüden değiştir komutu seçilerek.
        - ▪ Seçildikten sonra biraz beklenip tekrar seçilmesiyle.
        - ▪ Sadece yazı nesnesi için nesne üzerine çift tıklandığında veya sağ tuş menüsünden değiştir seçildiğinde açılan yeni formdan yazıya ait tüm özellikler değiştirilebilir.

- ✓ Ctrl+shift tuşuna basılı tutularak fare sol tuşu ile çizim nesneleri bir gruptan diğerine veya çöp grubuna taşınabilir.

- ✓ Silinerek çöp grubuna taşınan nesneler buradan başka bir gruba taşınarak tekrar gösterilmesi sağlanabilir.

- ✓ Nesneler çöp grubuna taşınarak silinmesi sağlanabilir.

- ✓ Ağaç üzerinde sağ tuş menüsünden yeni grup eklenebilir.

- ✓ Sağ tuş menüsünden yeni grup eklenebilir.

- ✓ Sağ tuş menüsünden güncelle komutu seçilerek veya ağacın altındaki güncelle butonuna basarak ağaçtaki veri ve değerlerin yeniden yüklenmesi sağlanabilir.

- ✓ Sağ tuş menüsünden veya Ctrl+C, Ctrl+V kısa yol tuşları ile bir grup kopyalanarak yapıştırılabilir.

- ✓ Ağacın altında yer alan Aç/Kapa komutu ile ağaçtaki tüm düğümlerin açılması ve kapanması sağlanabilir.

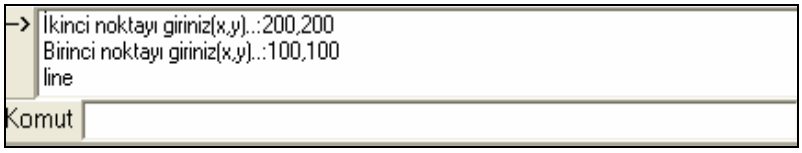
Ağaç üzerinde sağ tuşa tıklandığında açılan menüde yer alan komutlar Şekil 4.52'deki ekran görünümünde gösterilmiştir.



Şekil 4.52 Ağaç sağ tuş menüsü ekran görünümü

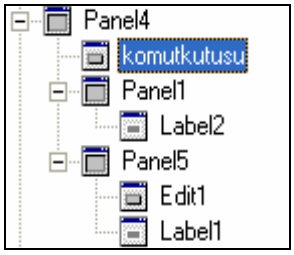
4.6.3. Komut kutusu

Komut kutusu kullanıcı ile etkileşimi sağlamak amacıyla ana form içine yerleştirilmiştir. Komut kutusunun ekran görünümü Şekil 4.53'de gösterilmiştir.



Şekil 4.53 Komut kutusu ekran görünümü

Komut kutusunda yer alan nesneler Şekil 4.54'de gösterilmiştir. Komut kutusu nesneleri panel4 üzerine yerleştirilmiştir. Kullanıcıya gerektiğinde açıklamaların gösterildiği veya edit1'den girilen komut verilerin gösterildiği Tlistbox nesnesi komut kutusu olarak adlandırılmıştır. Gösterilen son verinin üstte olduğunun anlaşılmasını sağlayan ok işareti panel1 üzerindeki label2 nesnesine yerleştirilmiştir. Komut yazısının bulunduğu label1 ve kullanıcının veri girdiği edit1 nesnesi panel5 üzerine yerleştirilmiştir.



Şekil 4.54 Komut kutusunda yer alan nesneler

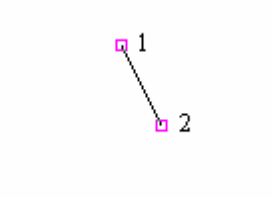
Komut kutusunda mesajlar yukarıdan aşağıya doğru akmaktadır. Yani yazılan en son mesaj komut kutusunda en üstte gösterilmekte diğerleri bir alta kaydırılmaktadır.

Örnek olarak çizgiyi komut kutusu kullanarak çizelim.

Komut: line

Birinci noktayı giriniz(x,y)..:10,10

İkinci noktayı giriniz(x,y)..:30,50



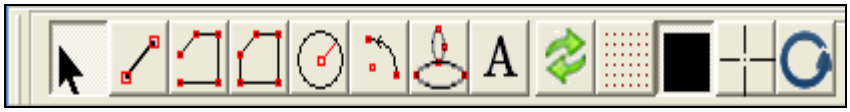
Komut olarak line girilerek giriş tuşuna basılmıştır. Kullanıcıdan birinci ve ikinci noktanın koordinatlarını girmesi istenmiştir. Girilen koordinatlara göre çizgi çizilerek ekranda gösterilmiştir.

4.6.4. Çizim araç çubuğu

Çizim araç çubuğunda yer alan komutlar Şekil 4.55'de gösterilmiştir. Çizim araç çubuğunda yer alan komutlar soldan sağa doğru sırasıyla şöyledir;

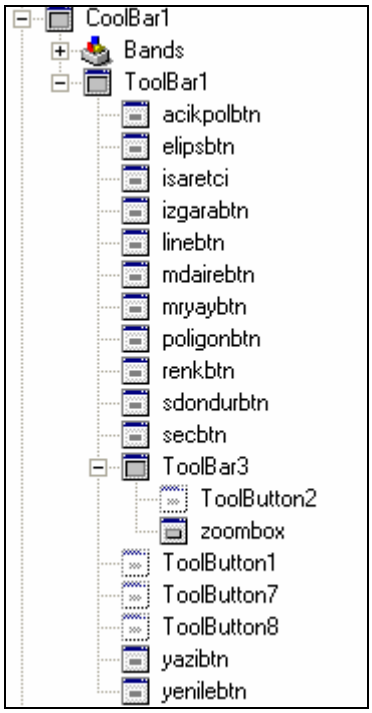
1. Seç
2. Çizgi
3. Ucu açık poligon
4. Ucu kapalı poligon
5. Daire
6. Yay
7. Elips
8. Yazı
9. Yeniden çiz
10. Izgarayı göster/gizle
11. Arka plan rengini siyah/beyaz yap
12. İşaretçiyi göster/gizle

13. Serbest döndür



Şekil 4.55 Çizim araç çubuğu ekran görünümü

Çizim araç çubuğunda komutların gösterilmesi için TspeedButton nesnesi kullanılmıştır. Kullanılan nesneler Şekil 4.56'da gösterilmiştir.

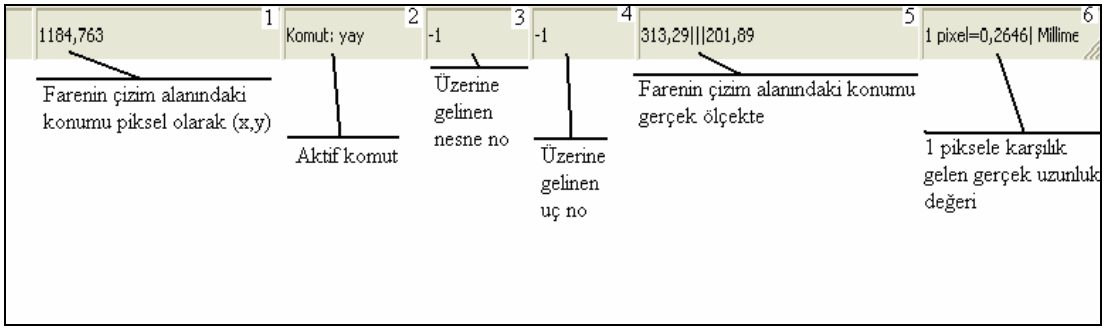


Şekil 4.56 Çizim araç çubuğunda yer alan nesneler

Şekil 4.56'da gösterildiği gibi çizim araç çubuğu Coolbar1 nesnesi üzerine Toolbar nesnesi kullanılarak yerleştirilmiştir.

4.6.5. Durum çubuğu

Durum çubuğu olarak Tstatusbar nesnesi kullanılmıştır. Durum çubuğunun ekran görünümü ve durum çubuğunda gösterilen veriler Şekil 4.57'de gösterilmiştir.



Şekil 4.57 Durum çubuğu ekran görünümü ve gösterilen nesneler

Durum çubuğunun 1. panelinde fare çizim alanında iken konumu piksel olarak gösterilmektedir. Ayrıca boyutlandırma esnasında da ilgili veriler burada piksel olarak gösterilir. Açı gösterildiğinde birim derecedir.

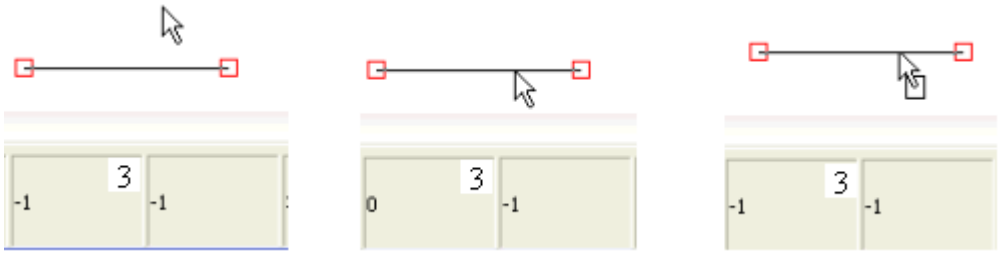
Durum çubuğunun 2. panelinde seçili olan komut gösterilmektedir.

Fare çizim alanında hareket ettirildiğinde her defasında Mouse.move olayı çalışır. Bu olayda diğer işlemlerin yanında fare işaretçisinin bir nesne üzerinde bulunup bulunmadığı da test edilmektedir. Test işlemi için her nesne için *üzerinde mi* alt programı yazılmıştır. Bunlar alt programlardır ancak program içinde kullanımından dolayı komut da denilebilir. Bu komutların tanımları ve çalıştırılma sırasına göre aşağıdaki gibidir;

- ✓ function polygonuzerimi(x,y:integer):integer;
- ✓ function plineuzerimi(x,y:integer):integer;
- ✓ function yayuzerimi(x,y:integer):integer;
- ✓ function daireuzerimi(x,y:integer):integer;
- ✓ function cizgiuzerimi(x,y:integer):integer;

- ✓ Function textuzerimi(x,y:integer):integer;
- ✓ function elipsuzerimi(x,y:integer):integer;

Tanımlarından da görüldüğü gibi bu alt programlara farenin konumu parametre olarak gönderilmekte ve bu alt programlarda farenin bir nesne üzerinde olup olmadığını test ederek üzerinde bulunulan nesnenin numarasını geri döndürmektedir. Her nesne sıfırdan başlayarak numaralandığı için burada gösterilen sayıyı farklılaştırarak her nesne için ayrı bir numara çıkartılması amacıyla üzerinde bulunulan nesne numaralarına bir ofset değeri eklenmiştir.



Şekil 4.58 Fare işaretçisinin nesne üzerine geldiğinde durum çubuğundaki üçüncü panelin ekran görünümü

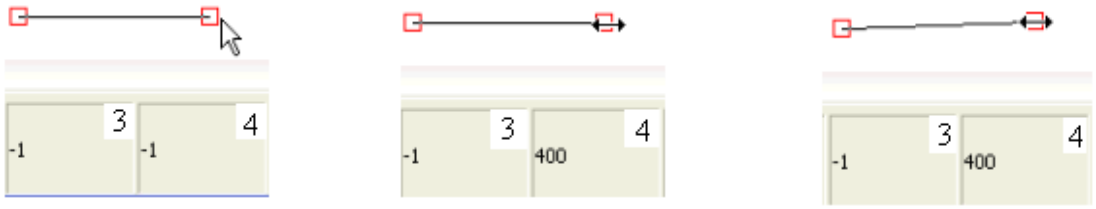
Durum çubuğunda üzerine gelinen nesneye ait numara yukarıda anlatıldığı gibi 3 numaralı alanda gösterilmektedir. Aynı zamanda bu numara ile nesnelerin seçilmesi de sağlanmaktadır. Nesnenin seçilmesi için farenin nesne üzerinde tıklanması gerekir. Farenin her hareketinde zaten taranan konum durum çubuğunda 3 numaralı panelde gösterilmektedir. Bu nedenle de fare tuşuna basıldığında bu alandaki değer okunarak farenin bir nesne üzerinde olup olmadığı test edilerek komut *seç* ise ilgili nesnenin seçilmesi sağlanmaktadır. Eğer fare bir nesne üzerinde değilse bu alanda -1 verisi gösterilmektedir.

Durum çubuğundaki 3. panelin nesne üzerinde değilken, nesne üzerine geldiğinde ve taşınırken aldığı durumlar Şekil 4.58'de gösterilmiştir.

Durum çubuğunda 4 numaralı panelde de fare işaretçisi seçili nesnelerde gösterilen boyutlandırma kutuları üzerine geldiğinde hangi kutuda olduğu

gösterilmektedir. Bu tarama yine Mouse.move olayında yapılmaktadır. Burada tarama işlemi için aşağıdaki alt program(komut) yazılmıştır.

- ✓ function uctami(x,y:integer):integer;



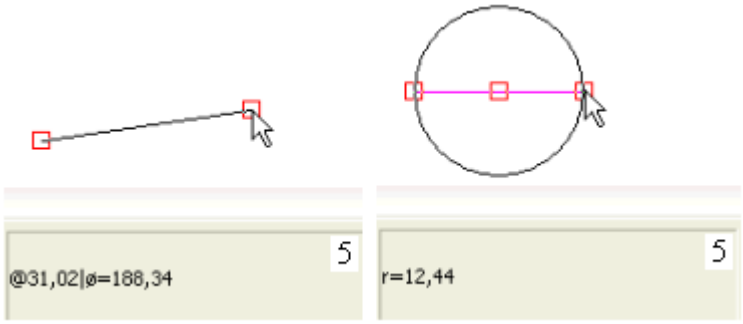
Şekil 4.59 Fare işaretçisinin nesne boyutlandırma kutusu üzerine geldiğinde durum çubuğundaki dördüncü panelin ekran görünümü

Tüm nesneler için ayrı ayrı komut yerine tek bir komut tanımlanarak farenin herhangi bir seçili nesnenin boyutlandırma kutusu üzerinde olup olamadığı test edilmektedir. Yine nesne numaralarının bir birine karıştırılmaması için her nesne ve uç noktası için bir ofset değeri eklenmektedir. Taranarak bu panele yazılan değer kullanılarak çizim alanında fare tuşuna basıldığında bir uç noktada olup olmadığı test edilir ve bir uç nokta üzerinde ise ilgili nesnenin boyutlandırılması sağlanır.

Durum çubuğundaki 4. panelin fare boyutlandırma kutusu üzerinde değilken, boyutlandırma kutusu üzerinde iken ve boyutlandırma esnasında aldığı değerler Şekil 4.59'da gösterilmiştir.

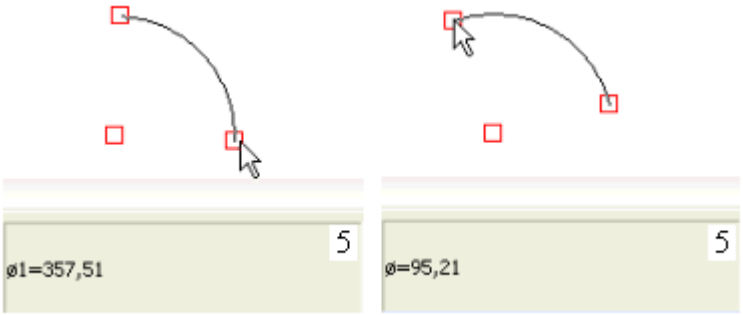
Durum çubuğunda 5 numaralı panelde farenin bulunduğu konum uygulama tercihleri penceresinden seçilen ölçü birimi ve bir piksel uzunluğun kaç birim olarak alınacağı durumuna göre farenin konumu, çizgi boyutlandırılırken uzunluğu ve açısı, daire boyutlandırılırken yarıçapı vb. bilgiler gösterilmektedir.

Şekil 4.60'da çizginin ve dairenin seçme kutularında boyutlandırılması esnasında 5. panelin ekran görünümleri verilmiştir. Çizgi boyutlandırılırken @ işaretinden sonra çizginin uzunluğu θ işaretinden sonrada eğimi derece olarak gösterilmektedir. Daire boyutlandırılırken de r ile yarıçapı gösterilmektedir.



Şekil 4.60 Çizgi ve daire boyutlandırılırken durum çubuğu 5. panelin ekran görünümü

Yayın seçme kutuları kullanılarak boyutlandırılması esnasında 5. panelin ekran görünümleri Şekil 4.61'de gösterilmiştir. İlk şekilde yay başlangıcından bulutlandırılmaktadır. Başlangıç açısı θ1 ile derece cinsinden 5. panelde gösterilmektedir. İkinci şekilde yay diğer ucundan boyutlandırılmaktadır. Burada da yay uzunluğu açı olarak θ ile gösterilmektedir.

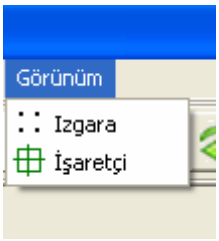


Şekil 4.61 Yay boyutlandırılırken durum çubuğu 5. panelin ekran görünümü

Durum çubuğunun 6. panelinde de 1 pikselin kaç birim olarak alındığı ve birimi gösterilmektedir. Bu değerler uygulama tercihleri penceresinde birimler sayfasından değiştirilebilmektedir.

4.7. Görünüm Menüsü

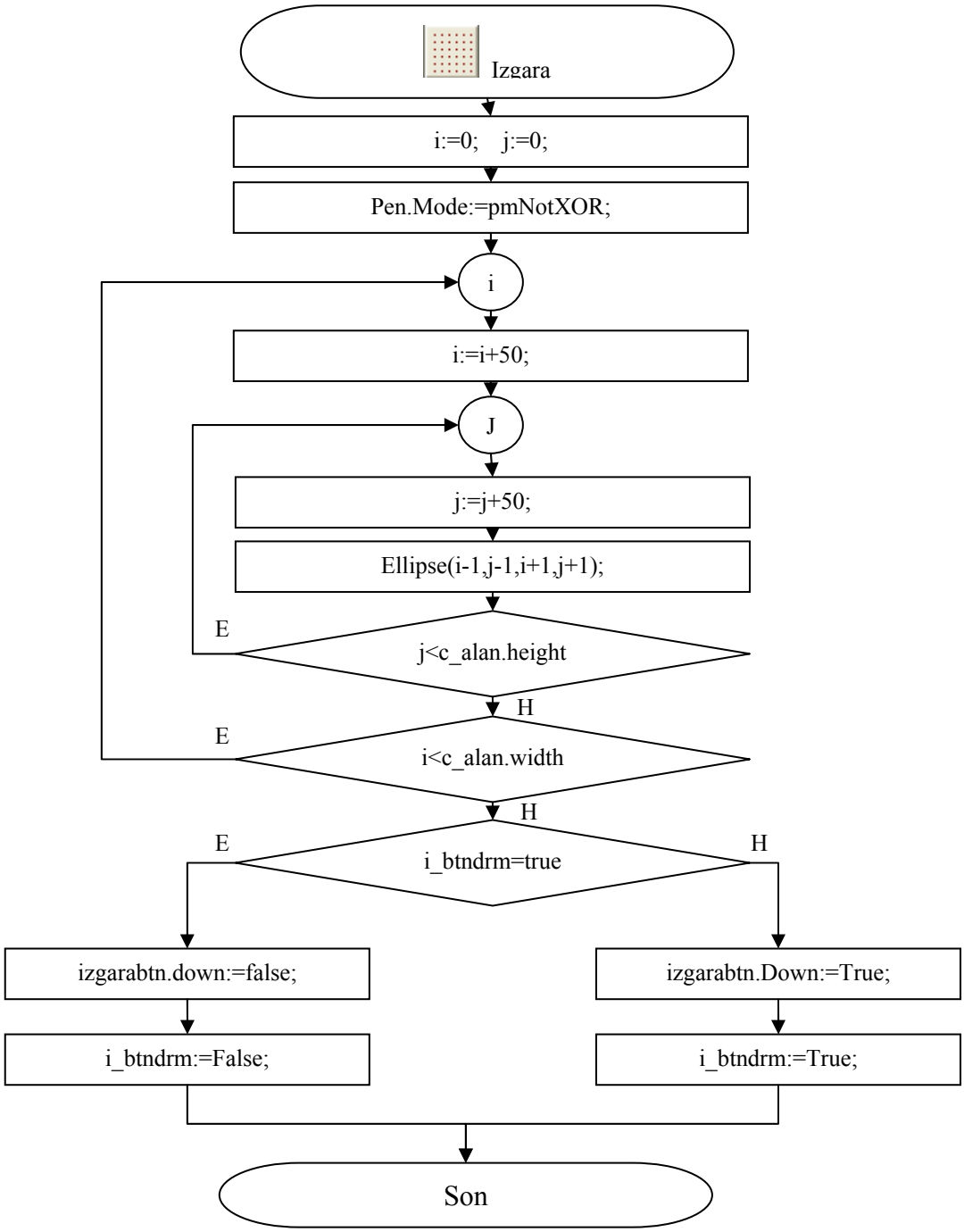
Görünüm menüsünde Izgara ve işaretçi olmak üzere iki seçenek yer almaktadır. Görünüm menünün ekran görünümü Şekil 4.62'de gösterilmiştir.



Şekil 4.62 Görünüm menüsü ekran görünümü

4.7.1. Izgara

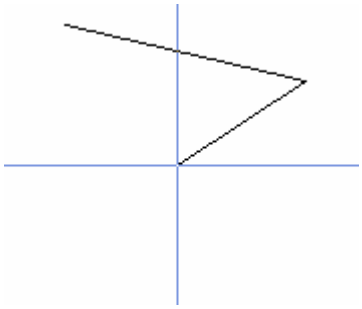
Izgara seçeneği, çizim alnında ızgara oluşturulmasını veya kaldırılmasını sağlar. Izgara araç çubuğundan gösterilip kaldırılabildiği gibi görünüm menüsünden de aynı işlemler yapılabilmektedir. Aslında ızgaranın gösterilmesini sağlayan kodlar araç çubuğundaki ızgara butonunun seçilmesi durumunda çalışmaktadır. Görünüm menüsünden ızgara seçeneği işaretlendiğinde ızgara butonunun tıklama olayını çağırır. Izgara butonu seçildiğinde çalışan kodlara ait akış çizelgesi Şekil 4.63'de gösterilmiştir.



Şekil 4.63 Izgara komutu akış çizelgesi

4.7.2. İşaretçi

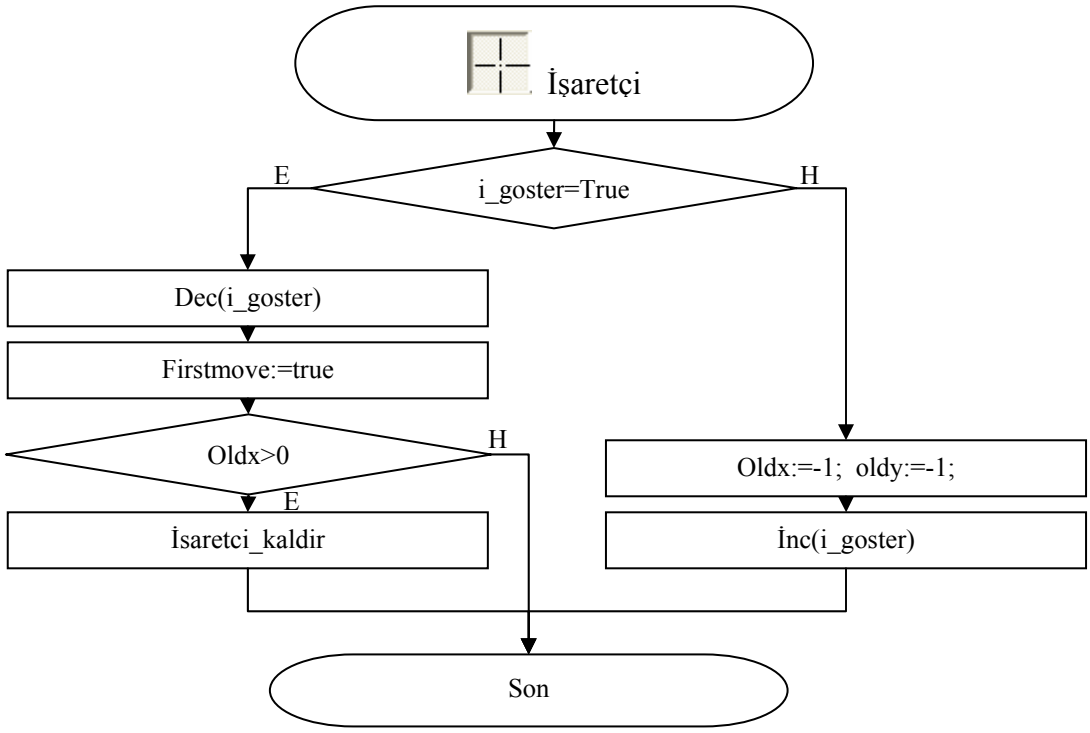
İşaretçi çizim alanında fare işaretçisi ile birlikte hareket eden ve yatay, dikeyde tüm çizim alanını boyunca uzanan ve güncellenen çizgilerden oluşur. Şekil 4.64'de çizim alanından bir bölümünün ekran görünümü alınmıştır. Burada yatay dikeyde saldan sağa ve üstten alta uzanan çizgiler işaretçidir. Bu çizgiler işaretçi aktif yapıldığında çizim alanında Mouse.move olayında sürekli güncellenmektedir.



Şekil 4.64 İşaretçi ekran görünümü

İşaretçi görünüm menüsünden veya araç çubuğundan aktif ya da pasif yapılabilir. İşaretçi butonuna basıldığında veya kodlar aracılığıyla bu olay çalıştırıldığında yapılan işlemlere ait akış çizelgesi Şekil 4.65'deki gibidir.

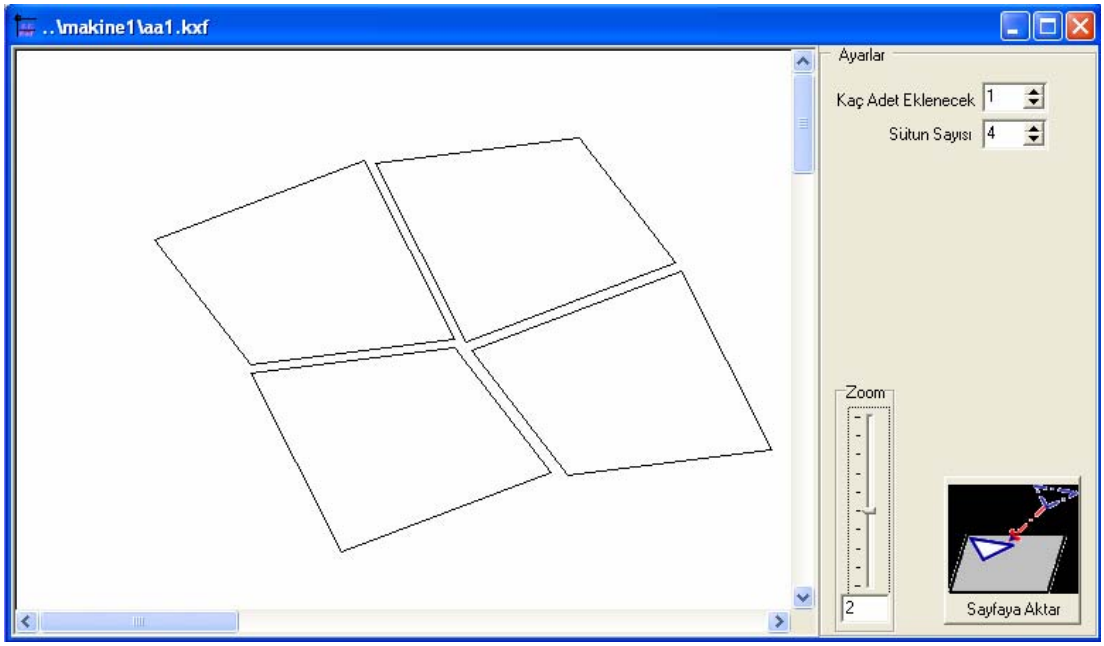
Burada yapılan işlem işaretçinin görüntülenip görüntülenmeyeceğini belirleyen i_goster değişkeninin durumunu ve buton durumlarını değiştirmektedir. Izgaranın asıl gösterimi Mouse.move olayında yapılmaktadır. Bu olayan başında çalışan kodlarla i_goster değişkeni test edilir ve değeri doğru ise Isaretci_goster(x,y) alt programı yada komutu çalıştırılır.



Şekil 4.65 Izgara komutu akış çizelgesi

4.8. Kütüphaneden Ekle Formu

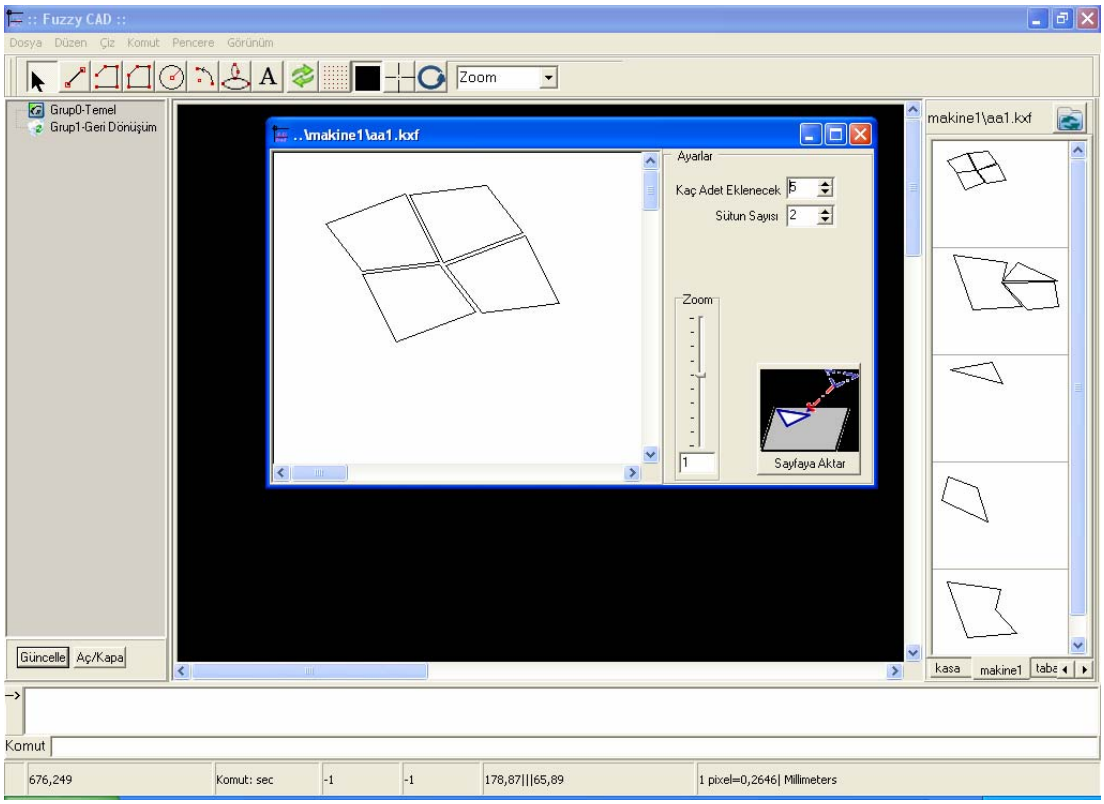
Kütüphaneden ekle formu ekran görünümü Şekil 4.66'da gösterilmiştir. Ana formda kütüphane penceresi aktif yapıldığında bir dosya üzerine çift tıklandığında bu form açılmaktadır. Açılan bu formda seçilen dosya gösterilmektedir. Bu formda dosyanın gösterildiği çizim alanında yakınlaştırma(zoom) ve kaydırma(pan) işlemleri tanımlanmıştır. Seçilen dosya bu form aracılığıyla ana formdaki çizim alanına istenilen sayıda tablo şeklinde yerleştirilmektedir. Bu formdan nesnenin kaç adet ekleneceği ve bir satıra kaç adet yerleştirileceği seçilerek sayfaya aktar butonuna tıklanmasıyla ana çizim alanına eklenmektedir. Ana form alanına eklenen her bir nesne bir grup olarak gösterilir. Böylece aktarma sonrasında nesneler istenildiği gibi düzenleme yapılabilmektedir.



Şekil 4.66 Kütüphaneden ekle formu ekran görünümü

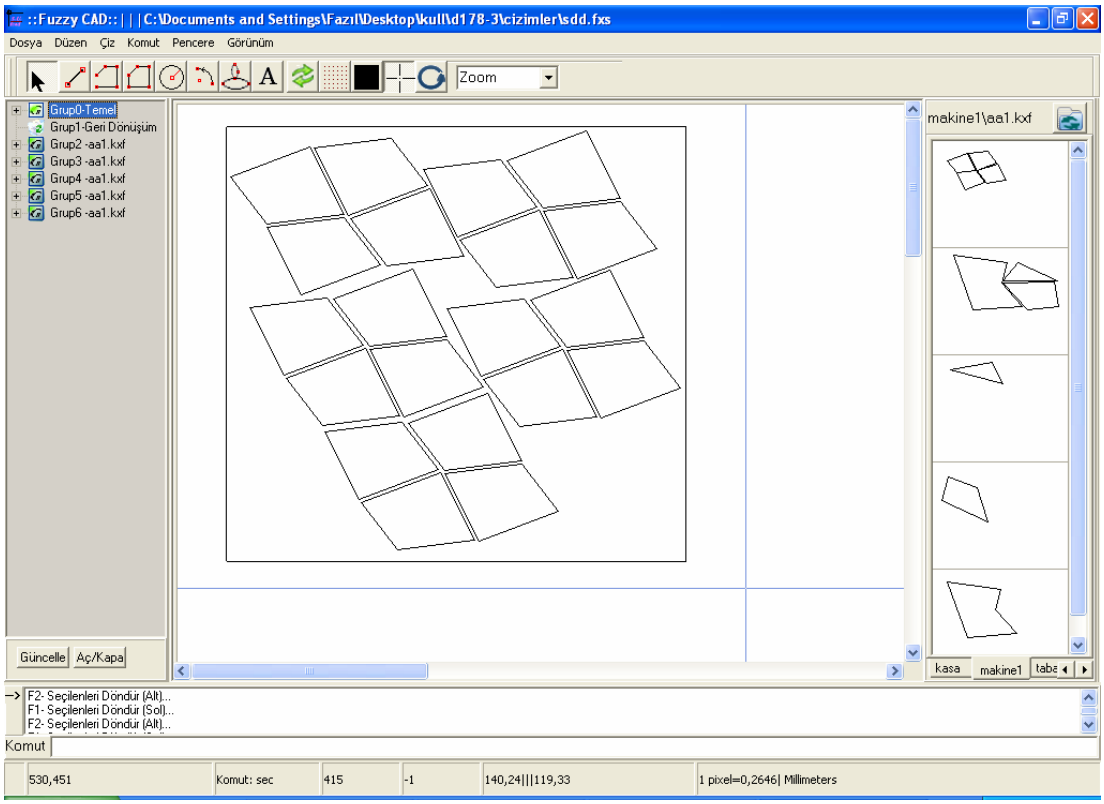
Şekil 4.67'de kütüphane penceresinden daha önce, kütüphanede makine klasörüne kaydedilmiş olan bir dosya üzerinde çift tıklanarak kütüphaneden ekle formunda gösterilmesi sağlanmıştır.

Açılan bu formdan nesnenin 5 adet ekleneceği seçilmiş ve bunların 2 sütun halinde eklenmesi istenmiştir. Daha sonra sayfaya aktar butonuna basılarak nesnenin ana forma aktarılması sağlanmıştır.



Şekil 4.67 Çift tıklanan bir dosyanın kütüphaneden ekle formunda gösterilmesi

Kütüphaneden ekle formunda gösterilen dosya sayfaya aktar komutunun seçilmesiyle ana forma aktarılmaktadır. Dosyanın ana forma istenilen sayıda ve şekilde aktarılması sonucunda oluşan ekran görünümü Şekil 4.68'de gösterilmiştir. Burada dosya çizim alanına 2 sütun halinde 5 adet eklenmiştir ve eklenen her bir dosya ağaç yapısında bir grup olarak gösterilmiştir.



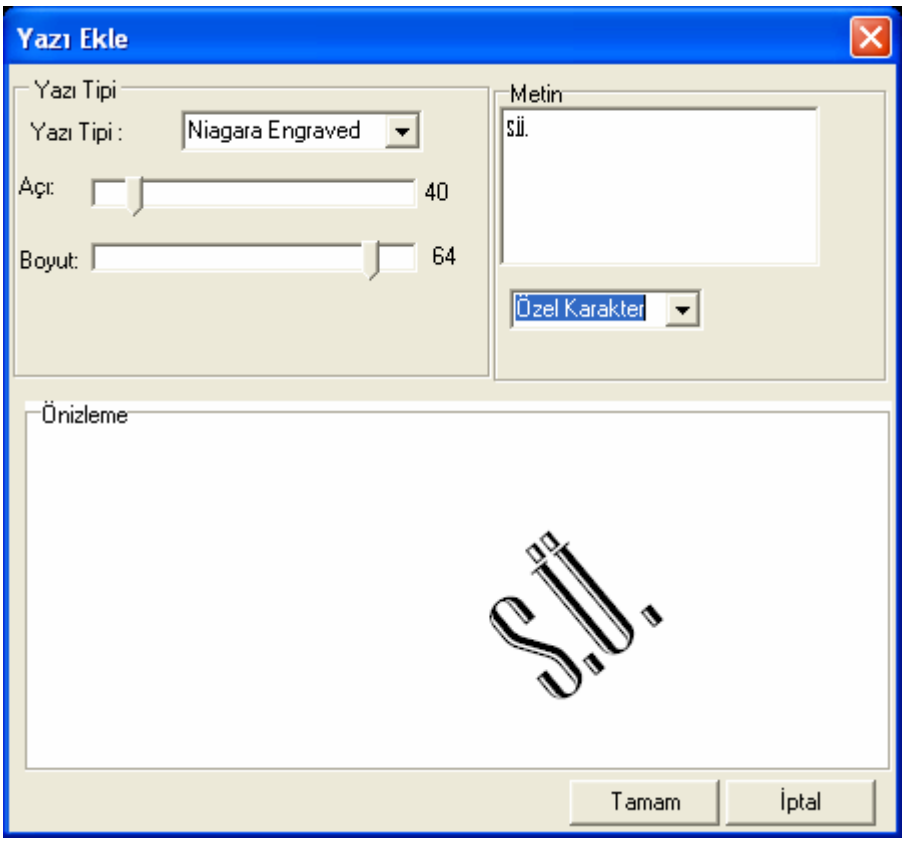
Şekil 4.68 Kütüphaneden ekle formundan sayfaya aktar komutu seçildikten sonraki ana form akran görünümü

4.9. Metin Ekle Formu

Metin ekleme formu çiz menüsünden yazı seçildiğinde, araç çubuğundan yazı butonu seçildiğinde veya ağaç üzerinden bir yazıya çift tıklandığında açılmaktadır. İlk iki yolla açıldığında yeni bir yaz eklemeyi sağlar. Ağaçtan daha önce yazılmış seçilerek açıldığında o yazıyı düzenlemeyi sağlar.

Metin ekleme formu ekran görünümü Şekil 4.69'da gösterilmiştir. Bu formda yazı metin alanına girilmektedir. Örnekte S.Ü. yazısı yazılmıştır. Girilen yazını tipi yazı tipi açılan kutusundan seçilebilmektedir. Açıyı ve yazı boyutunu ayarlamak trackbar nesnesi kullanılmıştır. Değiştirilen açı ve metin boyutu sağında bulunan etikette gösterilmekte aynı zamanda buraya tıklanarak kullanıcı klavyeden de değer girebilmektedir. Metnin girildiği yazı kutusunun altında yazıya özel karakter

eklenebilmesi bir açılan kutu konmuştur ve burada bazı özel karakterler tanımlanmıştır. Kullanıcı buradan yazıya özel karakterler ekleyebilir. Formun altında ön izleme alanı oluşturulmuştur kullanıcı buradan seçtiği ayarlara göre yazının nasıl görüntülendiğine bakabilmektedir. Aynı zaman fare ön izleme alanında hareket ettirildiğinde yazı da fare ile birlikte hareket etmektedir. Ana form da seçilen yazının eklenmesi gibi.

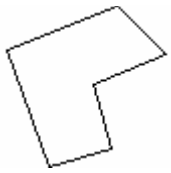


Şekil 4.69 Yazı ekle formu ekran görünümü

Yazı ekle formundan girilen ve özellikleri ayarlanan yazı tamam butonuna basılması ile ana çizim alanında yazı istenilen biçimde fare işaretçisi ile hareket eder kullanıcı yazı eklemek istediği yeri sol fare tuşu ile seçer. Yazıdan sol tuşla istediği kadar ekleyebilir. İşlemi sonlandırmak sağ fare tuşuna basması yeterli olmaktadır.

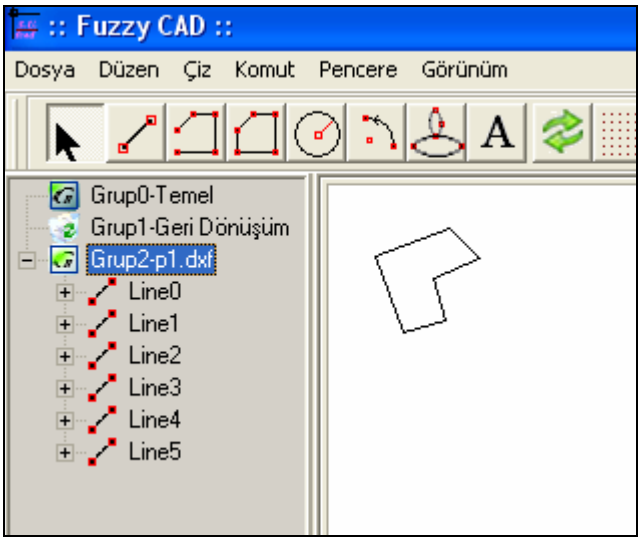
5. UYGULAMA

Şekil 5.1'de bir sac parçası Autocad 2005 programında çizilmiştir. Çizilen bu parça autocad programında Sava As komutu ile açılan pencereden, dosya türü kısmından Autocad 2000 DXF seçilerek p1.dxf adıyla kaydedilmiştir.



Şekil 5.1 Parçanın AutoCAD ekran görünümü

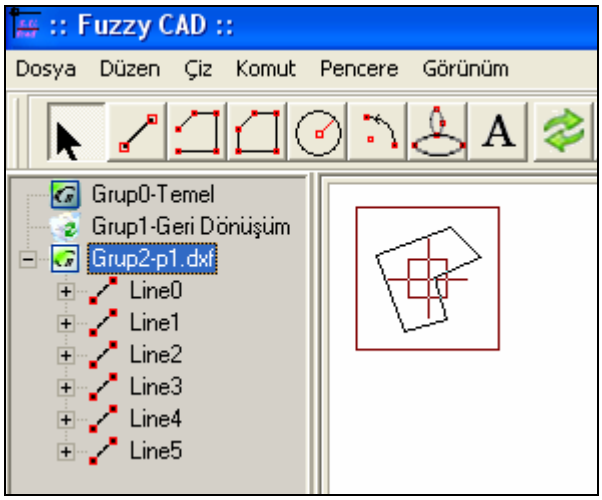
Autocad programında çizilerek dxf olarak kaydedilen çizim geliştirilen uygulama programında dosya menüsünden dxf al komutu seçilerek programa alınmıştır. Parçanın ekran görünümü Şekil 5.2'de gösterilmektedir.



Şekil 5.2 Dxf al komutu ile programa alınan çizimin ekran görünümü

Şekil 5.2'den de görüldüğü gibi dxf al komutu ile alınan çizim grup+dosya adı şeklinde ağaç yapısında temsil edilerek ekranda gösterilmektedir. Çizim autocad ile yapılırken çizgi komutu kullanılarak çizildiği için grup2'de de şeklin çizgilerden

oluştuğu görülmektedir. Alınan çizim üzerinde istenildiği gibi düzenleme yapılabilmektedir. Örneğin bu şekle ekleme ve çıkarmalar yapılabilir, dosya fxs uzantısı ile kaydedilebilir, dosya kxf uzantısı ile kütüphaneye kaydedilebilir vb. Dosyanın gösterildiği grup ağaç yapısından seçildiğinden grup seçili hale getirilerek çizim alanında gösterilir. Grup seçiminin yapılabilmesi için komut seç olmalıdır. Grup seçildiğinde oluşan ekran görünümü Şekil 5.3'de gösterilmiştir.

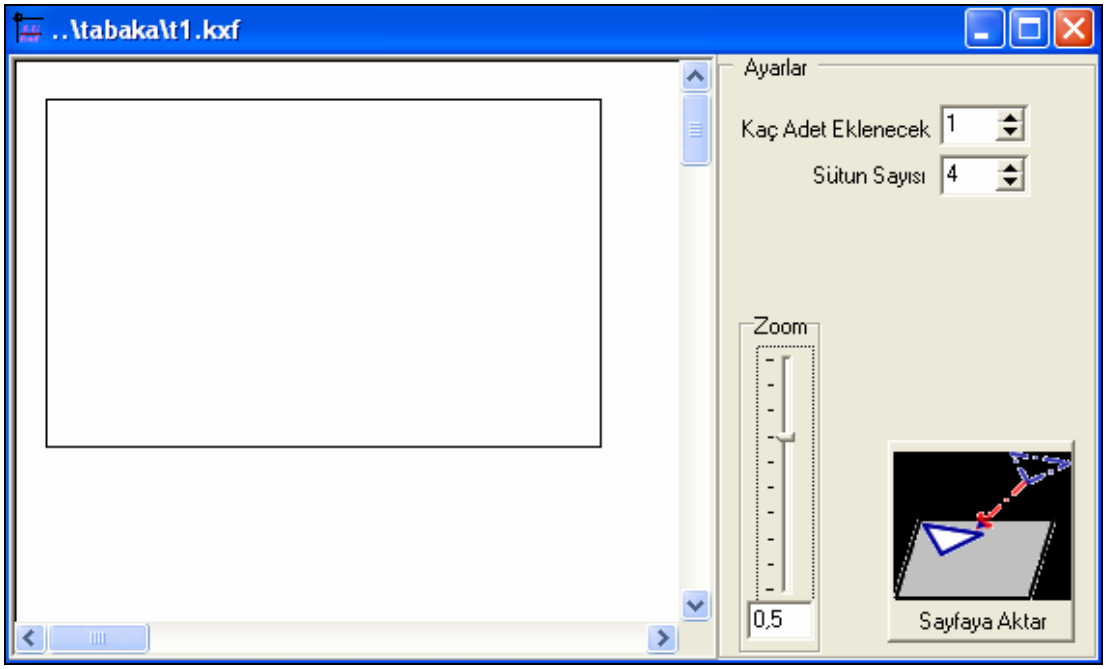


Şekil 5.3 Grup seçildiğinde oluşan ekran görünümü

Şekil 5.3'den de görüldüğü gibi grup seçildiğinde çizimi çerçeveleyen bir kutu çizilmektedir ve ortasında tutma kutusu oluşturulmaktadır. Yani içteki kare üzerine gelindiğinde fare ile grup tutularak taşınabilir. Taşıma esnasında F1, F2, F3, F4, F5 kısa yol tuşları ile seçili nesneler döndürülebilmektedir. Nesneye uygun şekil verildikten sonra çoğullanmak istenirse sağ tuşa basılarak çizili nesnenin bir kopyası ekrana yerleştirilmektedir.

Bu çalışmada amaç sadece dxf dosyaların programa dahil edilmesi değildir. Bir parçanın çizimi geliştirilen programda da yapılabilir veya başka bir programda yapılan ve dxf olarak kaydedilen bir çizimde alınarak düzenlenip kaydedilebilir. Asıl amaç çizilen veya alınan bu parçanın tabaka sac üzerine yerleştirilmesinin sağlanmasıdır. Autocad'de çizilerek programa dahil edilen parçanın bir tabakaya yerleşimini yapalım. Bunun için programı kullanarak daha önce ölçüleri ile çizilen bir tabakayı kütüphaneden programa alalım.

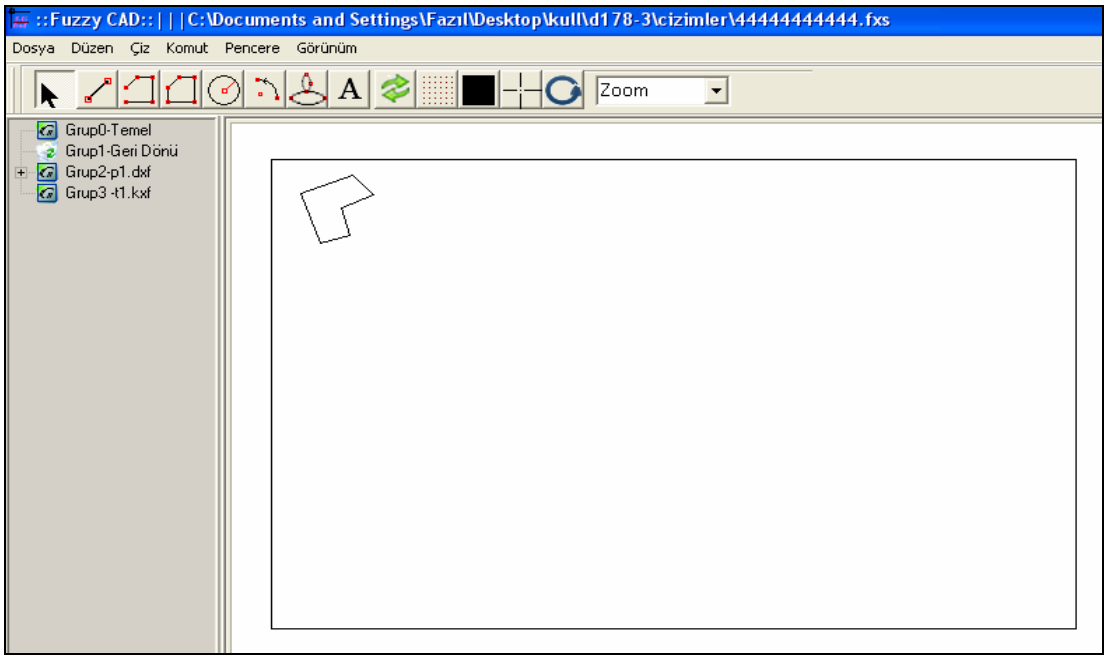
Bu amaçla öncelikle pencere menüsünden kütüphane seçeneğini aktif yapalım ve tabaka klasörüne kaydedilmiş olan t1.kxf dosyası üzerinde çift tıklayarak kütüphaneden ekle formunda açılmasını sağlayalım. Tabaka sacın kütüphaneden ekle formundaki ekran görünümü Şekil 5.4'de gösterilmiştir.



Şekil 5.4 Kütüphaneden ekle formunda tabaka sacın ekran görünümü.

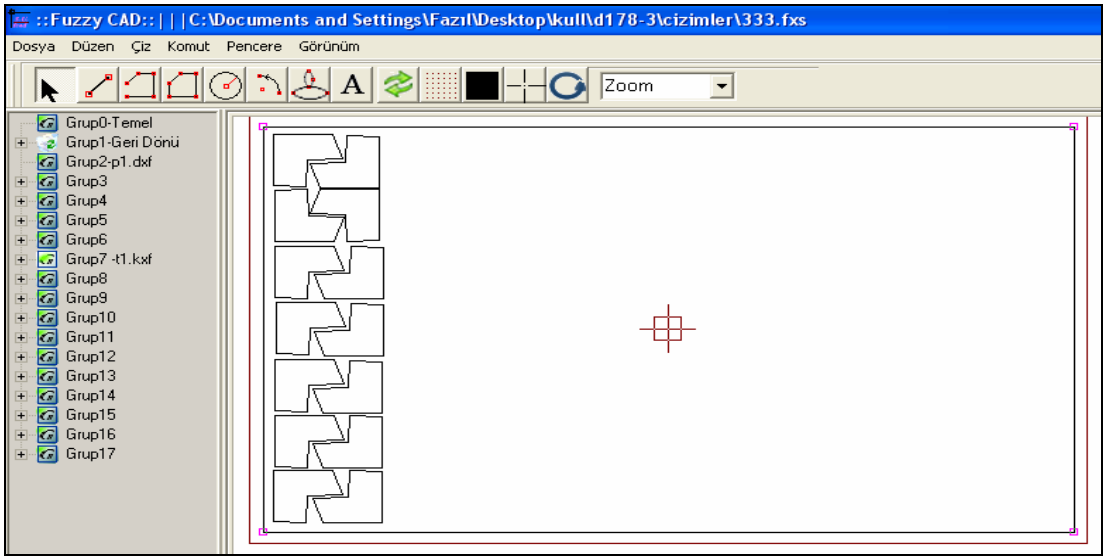
Şekil 5.4'de gösterilen tabaka sac sayfaya aktar komutu ile ana çizim alanına aktaralım.

Şekil 5.5'de daha önce dxf al komutu ile alınan nesne ve kütüphaneden eklenen tabaka saca ait ekran görünümü verilmiştir. Şekilden de görüldüğü gibi kütüphaneden eklenen tabakada çizim alanında gösterilmiş ve ağaca bir grup olarak eklenmiştir.



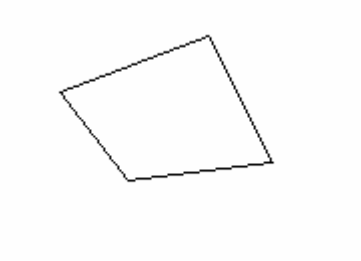
Şekil 5.5 Tabaka sac ve nesnenin ekran görünümü

Daha sonra döndürme işlemleri kullanılarak parçalar Şekil 5.6'daki gibi tabaka sac üzerine yerleştirilmiştir. Burada dxf olarak çizilip programa alınan nesneden 14 adet tabaka üzerine yerleştirilmiştir. Bu tabaka üzerine başka bir dosyadan alınan, kütüphaneden alınan veya çizilen parçalarda çizilerek yerleştirilebilir.



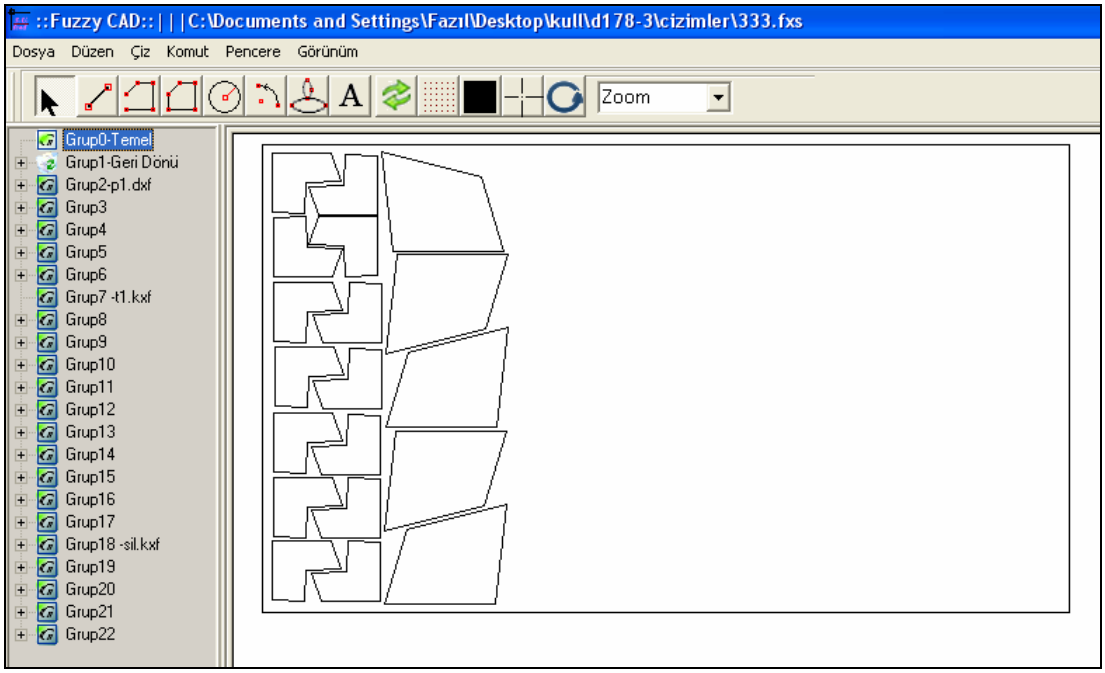
Şekil 5.6 Dxf parçanın yerleştirilişi

Şimdi de aynı tabaka üzerine daha önce çizilerek kütüphaneye kaydedilmiş olan Şekil 5.7'deki nesneyi ekleyelim.



Şekil 5.7 Kütüphaneden eklenen yeni nesne

Şekil 5.7'deki nesnenin tabaka sac üzerine yerleştirilmiş hali Şekil 5.8'de gösterilmiştir.



Şekil 5.8 İkinci yerleşim sonucu ekran görünümü

Burada örnek bir uygulama olarak dxf dosyadan ve kütüphaneden alınan nesneler tabaka bir saç üzerine yerleştirilmiştir.

6. TARTIŞMA

Bu çalışmada sac kesme işlemleri için bir bilgisayar destekli tasarım ve üretim sisteminin geliştirilmesi amaçlanmıştır. Program tasarım, yerleştirme, kesme yollarının çıkarılması ve tezgâha aktarılacak g veya m kodlarına dönüştürülmesidir. Bu bölümlere göre inceleme yaptığımızda;

Tasarım: Bu kısım için ara yüz oluşturulmuştur. Çizim nesneleri tanımlanmış ve çizimlerin diğer cad programlarında olduğu fare ile veya komutlarla çizilebilmesi sağlanmıştır. Tanımlanan nesnelerde (çizgi, daire, yay vb.) diğer cad programlarından eksik bir nesne yoktur. Diğer cad programlarında, bu nesneler üzerinde düzenleme yapılmasını sağlayacak daha fazla komut tanımlanmıştır. Düzenleme komutları gerektiğinde rahatlıkla programa eklenebilir. Diğer cad programlarında olduğu gibi çalışma için geliştirilen programda da farklı programlarda çizilen nesnelerin alınması için *Dxf al* komutu eklenmiştir. Kesilecek parçaların tasarımı yani çizimi için ara yüze birde ağaç yapısı eklenmiştir. Bu ağaçta çizilen nesneler bir gruba dahil edilerek gerekli parametreleri ile gösterilmektedir. Bu ağaç diğer cad programlarında bulunmamaktadır. Ağaç nesneler üzerinde düzenleme, seçme, silme, taşıma vb. işlemlerin yapılmasını sağlamaktadır. Aynı zamanda buradaki gruplar diğer cad programlarında bulunan katman yapısının yerine kullanılabilir. Aynı işi yapması sağlanabilir. Bir parçayı tanımlayan çizim bir grup içinde gösterildiğinden yerleştirme için kolaylık olması amacıyla grubun seçilerek taşıma esnasında döndürülmesi ve sadece farenin sağ tuşuna basılması ile çoğaltılması sağlanmıştır. Diğer cad programları ile kıyaslandığında bir diğer farklılık da kütüphane yapısının oluşturulmuş olmasıdır. Böylece bir makine veya birim için kesilmesi gereken tüm parçalar çizilip kütüphane klasörü içinde bir klasöre kaydedildiğinde; programda kütüphane penceresinde bir sayfa içinde klasörde bulunan bütün dosyalar ön izleme olarak açılıp çizim alanına istenilen sayıda aktarılması sağlanabilmektedir.

Yerleştirme: Yerleştirme kapsamında programda dizi yerleştirme işlemi tanımlanmıştır. Birde kullanıcının grup olarak gösterilen nesneleri çoğaltarak yerleştirmede kolaylık olası açısından sağ tuş ile yerleştirme ve taşıma esnasında F tuşları ile döndürme işlemlerinin yapılması sağlanmıştır. Ancak hedeflenen parçaların çizim alanına istenilen sayıda optimum şekilde yerleştirilmesinin program tarafından yapılmasının sağlanmasıdır. Bunun içinde genetik ve heuristik algoritmaları ile diğer bulanık mantık vb. yöntemleri tek başına veya birleştirilerek kullanılması gerekmektedir. Bu konuda yeterli kaynak taraması yapılmış ve bu algoritmalar incelenmiştir. Ancak programa aktarılması sağlanamamıştır. Bu da diğer tasarım alanı ve programın genel yapısının oluşturulması için planlanandan daha fazla zaman harcanmış olmasıdır. Yerleştirme algoritmalar anlaşıldıktan bu kolaylıkla programa dahil edilebilir. Programın yapısı buna izin vermekte ve gerekli yapıyı sunmaktadır.

Kesme yolları: Çizilen nesneleri oluşturan çizgiler zaten kesme yollarıdır. Bu yollar hafızada ifade edilmiş olduğu için bu yolların gösterimi rahatlıkla yapılabilir. Ancak burada dikkat edilmesi gereken husus bu yolların tezgâhın çalışma yapısıyla uyumlu olacak şekilde gösterilmesidir. Bu işlem kullanıcıya yerleştirme sonucu parçaların tezgâhta nasıl kesileceğini geçiş yollarını doğru olarak göstermelidir. Gerektiğinde buradan yollar ve adımlara müdahale edilebilmeli yapılan düzeltmeler tezgâha aktarım aşamasında aynen uygulanarak aktarılması sağlanmalıdır. Bu kısım içinde kaynak taraması yapılmıştır. Bu işlemi yapan piyasadaki programlar incelenmiştir. Nasıl olacağı ve programa nasıl ekleneceği düşünülmüştür. Ancak zaman yetersizliğinden dolayı uygulaması yapılamamıştır.

Makine kodları: Burada makine kodlarından kasıt CNC tezgâhlarda kullanılan ve dünyada standart olan g ve m kodlarıdır. Programın tam bir tasarım-üretim bütünleşmesini sağlaması için mutlaka yapılan işlemlerin sonucunu doğrudan ilgili tezgâha aktarılabilmesi gerekir. Bu konuda da kaynak taraması ve mevcut programların incelenmesi yapılmıştır. Nasıl yapılacağı belirlenmiştir. Bu kodlar sınırlı sayıda komuttan oluştuğundan aktarmada bir zorluk olmayacaktır. Bu kısımda rahatlıkla program içine eklenebilir.

7. SONUÇ VE ÖNERİLER

Bu çalışmada sac metallerin kesiminde kullanılacak bilgisayar destekli tasarım ve üretim sistemi geliştirilmiştir. Geliştirilen bu sistem sayesinde cad programlarının yapısı, nasıl çalıştıkları, dosya yapıları vb. konularda birçok sorunun cevabı bulunmuştur. Bunun yanında görsel programlama ile alakalı olarak da nesne tanımlama, dosya formatı oluşturma, çalışma anında nesne oluşturma, ölçülendirme vb. birçok konuda bilgi ve deneyim kazanılmıştır. Analitik geometriye dayalı olarak nesneler çizilip üzerlerinde işlem yapıldığı için bu alandaki bilgiler tekrar edilerek, bilgisayara aktarılması sağlanmıştır. Ayrıca yerleştirme konusunda çalışmalar yapılmıştır. Bu konu birçok alana uygulanabilecek karmaşık algoritmaları içeren esas programlama öğeleridir.

Yapılan bu program ve benzerleri sanayide çeşitli firmalar tarafından kullanılmaktadır. Bazı firmalar bu programların incelenmesine izin vermedikleri için ve çoğu da tezgâhla verilen programlar olduğu için çalışması dahi zorluklarla kısıtlı olarak incelenebilmiştir. Ayrıca incelense dahi bu bize programda yer alan pencerelerin ve özelliklerin anlaşılmasını sağlar programın nasıl yapıldığı ve algoritması hakkında bir bilgi vermez. Yani kodlarını görüp incelemek mümkün değildir. Bu nedenle de her şey baştan geliştirilmiş nasıl olması gerektiği tek tek düşünülmüş ve yapılmıştır. Bu da bazen çok basit görünen işlemlerin günlerce uğraşı sonucunda yapılmasına neden olmuştur. Programda delphi nesneleri dışında hazır bir bileşen kullanılmamıştır.

Bu çalışmanın eksik kalan kısımlarının tamamlanması için doktora çalışması olarak devam edilmesi çok uygun olacaktır. Bilindiği gibi doğrudan üretimde kullanılacak uygulamanın hatasız, üreticiler için gerekli kolaylık ve özellikleri içermesi gerekmektedir. Bu konuda da son aşamada piyasa araştırması ve denemeler yapılabilmesi için üzerinde çalışılması iyi olur.

KAYNAKLAR

1. http://www.aliinal.com/Uploads/documents/notlar2003/BBU.doc
   Son Erişim: 10.11.2006

2. XIE S. Q., XU X. September 2006, A STEP-compliant process planning system for sheet metal parts. International Journal of Computer Integrated Manufacturing, Vol. 19, No. 6, , 627 – 638,

3. Sharon J. Kemmerer (ed.), *STEP, the Grand Experience,* NIST Special Publication 939, National Institute of Standards and Technology, Gaithersburg, MD (1999).

4. http://cadd.web.cern.ch/cadd/cad_geant_int/thesis
   /node33.html#SECTION00011200000000000000
   Son Erişim: 12.01.2007

5. Fujita K, Akagi S and Hirokawa N. Hybrid approach for optimal nesting using a genetic algorithm and a local minimization algorithm. Proceedings of ASME Design Automation Conference. Albaquerque. New Mexico. U. S . A. 65: 47744,1993

6. Su Y X, Zhang C .H and Duan 8 Y. An optimal layout methodology for cold forming rectangular parts using genetic algorithms. Robotics and Computer Integrated manufacturing. 17: 429433,2001

7. Beasley J. E., 1985, Algorithms for unconstrained two-dimensional guillotine cutting. Journal of Operational Research Society, 36, 297 - 306.

8. İsmet Çalik, Ali Ünüvar ve Süleyman Yaldız, Dönel parçaların grafik simülasyonu, işlem planlaması ve CNC programlarının türetilmesi, Selçuk Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi cilt:13 Yıl:1998
9. Shpıtalnı M. and Manevıch V., 1996, Optimal orthogonal subdivision of rectangular sheets. Transactions of the ASME, Journal of Manufacturing Science and Engineering, 118, 281-288.

10. Babu R.A. and Babu R.N., 1988, Effective nesting of rectangular parts in multiple rectangular sheets using genetic and heuristic algorithms. int. j. prod. res., 1999, vol. 37, no. 7, 1625 - 1643

11. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gdi/cordspac_3c6d.asp
    Son Erişim: 26.12.2006

12. http://www.autodesk.com/techpubs/autocad/acad2000/dxf/index.htm
    Son Erişim: 18.02.2007

13. http://www.technology.ccsu.edu/personnel/information/it500/CH7_PowerPoint_CAM.ppt
    Son Erişim: 13.01.2007

14. http://www.technology.ccsu.edu/personnel/information/it500/CH7_PowerPoint_CAM.ppt
    Son Erişim: 10.11.2006
15. http://cadd.web.cern.ch/cadd/cad_geant_int/thesis/node31.html
    Son Erişim: 20.02.2007

16. Roger N. Nagel, Walt W. Braithwaite, and Philip R. Kennicott, *Initial Graphics Exchange Specification IGES, Version 1.0,* NBSIR 80-1978, National Bureau of Standards, Washington, DC (1980).