

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ENDÜSTRİYEL TESİSLERDE, DAĞINIK GRUP İLETİŞİM VE
BULANIK ALGORİTMA METOTLARI KULLANILARAK, ACİL
DURUM TOPLANMA ALANLARININ DİJİTAL SAYIMI**

YÜKSEK LİSANS TEZİ

Muhammed Fatih PEKŞEN

Enstitü Anabilim Dalı : YANGIN GÜVENLİĞİ VE YANMA

Tez Danışmanı : Doç. Dr. Yılmaz UYAROĞLU

Haziran 2018

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİYEL TESİSLERDE, DAĞINIK GRUP İLETİŞİM VE
BULANIK ALGORİTMA METOTLARI KULLANILARAK, ACİL
DURUM TOPLANMA ALANLARININ DİJİTAL SAYIMI

YÜKSEK LİSANS TEZİ

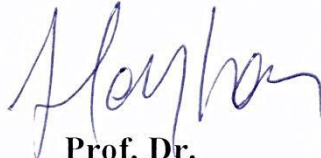
Muhammed Fatih PEKŞEN

Enstitü Anabilim Dalı : YANGIN GÜVENLİĞİ VE YANMA

Bu tez 25.06.2018 tarihinde aşağıdaki jüri tarafından oybirliği / oycokluğu ile kabul edilmiştir.



Doç. Dr.
Yılmaz UYAROĞLU
Jüri Başkanı



Prof. Dr.
Hakan Serhad SOYHAN
Üye



Doç. Dr.
Cenk ÇELİK
Üye

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Muhammed Fatih PEKŞEN

05.06.2018

TEŐEKKÜR

Yüksek lisans eğitiminin boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Doç. Dr. Yılmaz UYAROĞLU'na teşekkürlerimi sunarım.

Ayrıca çalışmam boyunca manevi desteğini eksik etmeyen aileme teşekkür ederim.

İÇİNDEKİLER

TEŞEKKÜR	i
İÇİNDEKİLER	ii
SİMGELER VE KISALTMALAR LİSTESİ	v
ŞEKİLLER LİSTESİ	vi
TABLolar LİSTESİ	viii
ÖZET	ix
SUMMARY	x

BÖLÜM 1.

GİRİŞ	1
-------------	---

BÖLÜM 2.

KAYNAK ARAŞTIRMASI	5
2.1. Dağınık Grup Haberleşme Sistemleri	5
2.1.1. Dağınık grup haberleşme sistemi çeşitleri	8
2.1.1.1. Kapalı grup	8
2.1.1.2. Açık grup	9
2.1.2. Sistem tasarımı.....	9
2.1.2.1. Sistem mimarisi.....	10
2.1.2.1.1. Merkezi yapılı mimari tasarımı.....	12
2.1.2.1.2. Merkezi olmayan mimari tasarımı.....	14
2.1.2.1.3. Hibrit yapılı mimari tasarımı.....	14
2.1.3. Yazılım mimarisi.....	16
2.1.3.1. Engelleyici unsurlar.....	17
2.1.3.2. Sistemin kendi kendine yönetme metodu.....	18

2.1.4. Haberleşme.....	20
2.1.4.1. Uzaktan prosedürlü çağrı.....	20
2.1.4.2. Mesaj bazlı haberleşme.....	21
2.1.4.3. Çok noktalı haberleşme.....	23
2.1.5. Bölgelerin adlandırılması (adreslemesi).....	24
2.1.5.1. İsimler, tanımlayıcılar ve adresler.....	25
2.1.5.2. Hiyerarşik adresleme yöntemi (flat naming).....	26
2.1.5.2.1. Basit çözümler (simple solutions).....	26
2.1.5.2.2. Ev-temelli yaklaşımlar.....	27
2.1.5.3. Alan adı adresleme yöntemi.....	27
2.1.5.4. Hiyerarşik ve merkezi olmayan adresleme yöntemi.....	29
2.1.6. Senkronizasyon problemine çözüm metodu.....	30
2.1.6.1. Sistem saat senkronizasyonu metodu.....	30
2.1.6.2. Lojik saat sistemi metodu.....	31
2.1.6.3. Dışlama metodu.....	32
2.1.7. Toleranslar.....	36
2.1.7.1. İki aşamalı sistem kurulumu.....	41
2.1.8. Kurtarma.....	42
2.1.8.1. Mesajların yedeklenmesi.....	44
2.1.8.2. Yedeklenme bölgesinin seçimi.....	45
2.2. Bulanık Mantık Metodu İle Dijital Sayım Benzetimi.....	45
2.2.1. Bulanık kümeler.....	47
2.2.2. Dilsel değişkenler, terimler.....	49
2.2.3. Üyelik fonksiyonları.....	51
2.2.4. Bulanık kurallar.....	52
2.2.5. Bulanık mantık uygulaması.....	53
2.2.6. Mamdani bulanık modeli.....	54

BÖLÜM 3.

SİSTEM TANITIMI VE YÖNTEM.....	57
3.1. Dijital Sayım Programı Geliştirilmesi.....	57
3.1.1. Sistem tanıtımı.....	57

3.1.2. Uygulama hazırlama aşaması	58
3.1.3. Yazılan uygulamanın işleyişi.....	60
3.1.4. Uygulamanın akış diyagramı.....	65
3.1.4.1. Dijital sayım bölümü.....	65
3.1.4.2. Yeni veri girişi.....	70
3.1.4.3. Veri silme.....	71
3.2. Bulanık Mantık Uygulanarak Sistemin Modellenmesi.....	72
3.3. Bulanık Kuralların Modellenmesi Ve Görüntülenmesi.....	74
BÖLÜM 4.	
ARAŞTIRMA BULGULARI	79
4.1. Elde Edilen Bulgular Ve Genel Yorum.....	79
BÖLÜM 5.	
SONUÇ VE ÖNERİLER	81
KAYNAKLAR	84
ÖZGEÇMİŞ	88

SİMGELER VE KISALTMALAR LİSTESİ

DT	: Data Table
GPS	: Global Positioning System
İHS	: İnternet Hizmet Sağlayıcısı
RFT	: Radyo Frekanslı Tanımlama

ŞEKİLLER LİSTESİ

Şekil 2.1. Kapalı grup sistematiği.....	9
Şekil 2.2. Açık grup sistematiği.....	9
Şekil 2.3. Bir istemci ile sunucu arasındaki genel etkileşim.....	12
Şekil 2.4. İnterneti kenar sunucularından oluşan bir koleksiyon.....	15
Şekil 2.5. Bittorent çalışma prensibi.....	16
Şekil 2.6. Uzak nesne çağrılarını işlemek için engelleyici kullanma.....	18
Şekil 2.7. Geri bildirim kontrol sisteminin çalışma prensibi.....	19
Şekil 2.8. Merkezi algoritma.....	33
Şekil 2.9. Dağınık algoritma.....	35
Şekil 2.10. Simgeli halka algoritması.....	35
Şekil 2.11. İki aşamalı sistem.....	42
Şekil 2.12. İstikrarlı depolama.....	44
Şekil 2.13. Üyelik işlemlerini bağlamak için VE, VEYA, DEĞİL operatörleri.....	52
Şekil 2.14. Bulanık mantık sıcaklık.....	53
Şekil 2.15. Mamdani bulanık mantık (Min ve Max).....	55
Şekil 2.16. Mamdani bulanık mantık (Max).....	55
Şekil 3.1. Sistemin genel görüntüsü.....	58
Şekil 3.2. Tesis/işletme dijital sayım ara yüzü (Boş listeler).....	61
Şekil 3.3. Tesis/işletme dijital sayım ara yüzü (Dolu listeler).....	62
Şekil 3.4. Uyarı ekranı.....	63
Şekil 3.5. Kayıtlı çalışanlar listesi.....	64
Şekil 3.6. Dijital sayım bölümü girişi.....	65
Şekil 3.7. Veri tablosu 1 tarih hücresi kontrolü.....	66
Şekil 3.8. Veri tablosu 1 tarih hücresi yeşil olmayan satırların silinmesi.....	66
Şekil 3.9. Veri tablosu 1 ile veri tablosu 2'nin isim hücrelerinin karşılaştırılması..	67

Şekil 3.10. Sayıma girmeyenler listesinin renklendirilmesi ve veri tablosu 2 ile veri tablosu 3'ün karşılaştırılması.....	68
Şekil 3.11. Listelerin oluşturulması ve sıralı şekilde yazdırılması.....	69
Şekil 3.12. Yeni verilerin sisteme girilmesi işlemi.....	71
Şekil 3.13. Veri silme işlemi.....	71
Şekil 3.14. MATLAB programı terminaller arası uzaklık üyelik fonksiyonu.....	72
Şekil 3.15. MATLAB programı terminale uzaklık üyelik fonksiyonu.....	72
Şekil 3.16. MATLAB programı terminal adet üyelik fonksiyonu.....	73
Şekil 3.17. MATLAB programı Mamdani ve ağırlık merkezi durulaştırma metodu sonuç ekranı 1.....	75
Şekil 3.18. MATLAB programı Mamdani ve ağırlık merkezi durulaştırma metodu sonuç ekranı 2.....	76
Şekil 3.19. MATLAB programı Mamdani ve ağırlık merkezi durulaştırma metodu sonuç ekranı 3.....	77
Şekil 3.20. MATLAB programı Mamdani ve ağırlık merkezi durulaştırma metodu sonuç ekranı 4.....	78

TABLolar LİSTESİ

Tablo 2.1. Hata tipleri ve tanımları.....	39
Tablo 2.2. Terminaller arası uzaklık kümesi dilsel deęişkenleri.....	50
Tablo 2.3. Terminale uzaklık kümesi dilsel deęişkenleri.....	50
Tablo 2.4. Toplam terminal sayısı kümesi dilsel deęişkenleri.....	51
Tablo 2.5. Üyelik fonksiyonlarında VE, VEYA, DEĞİL.....	51
Tablo 3.1. MATLAB bulanık kural tablosu.....	73
Tablo 3.2. Bulanık kurallar.....	74

ÖZET

Anahtar kelimeler: Dağınık grup iletişim metodu, bulanık algoritma metodu, acil durum toplanma bölgesi sayım işlemi, dijital sayım.

Bu çalışmada Türkiye Cumhuriyet Devleti'nin 2012 yılında kabul etmiş olduğu 6331 sayılı kanuna istinaden orta ve büyük ölçekli tesislerde/işletmelerde yapılan acil durum tatbikatlarının toplanma bölgelerindeki klasik sayım işlemi yerine dijital sayım işlemi konusu işlenmiştir. Yöntem olarak dijital sayım işlemi için dağınık grup haberleşme metodu ve bulanık mantık metodu kullanılmıştır.

Dağınık grup haberleşme sistemi kurulurken, radyo frekanslı tanımla sistemlerinden faydalanılmıştır. Virdi marka AC-2100 model cihazlar ile biyometrik parmak izi okuma veya personel giriş kartlarından okutma işlemi yapılarak dijital sayım işlemleri gerçekleştirilmiştir. Dijital sayım işlemlerinin sonuçlarının elde edilmesi için Visual Studio C# programı ile ara yüz yazılmıştır. Dijital sayım sonuçları, bu ara yüz ile ilgili listelere aktarılmıştır. Sonuçlarda tesiste/işletmede çalışan personelin veya ziyaretçilerin hangi acil durum toplanma bölgesinde sayıma girdiği ve sayıma girmeyen çalışanların/ziyaretçilerin listesi gösterilmiştir. Tesis/işletme büyüklüğüne göre ve çalışan personel sayısına göre sahada kaç adet dijital sayım cihazının konumlandırılması gerektiğine, bulanık mantık metodu kullanılarak çözüm aranmıştır. Dijital sayım cihaz adet sayısını belirlemek için ortalama ölçülere sahip bir yetişkinin hareket hızı 1,2 m/s - 1,4 m/s (Schneider ve Kirchberger, 2007) verisi kullanılmıştır. Hesaplamalarda yol değişkeni hız ve zaman değişkenlerinin çarpımına eşittir formülü kullanılmıştır. Bulanık mantık betimlemesi yapılırken çözümlemede Mamdani metodu ve durulaştırmada ağırlık merkezi metodu kullanılarak Matlab programından yararlanılmıştır.

Çıkan sonuçlara göre orta ve büyük ölçekli tesiste/işletmede dijital sayım metodu, klasik sayım metoduna göre daha hızlı sonuç vermiştir. Acil durumlarda, kayıp kişilere ulaşmak için harcanan zaman çok önemlidir. Bu harcanan zamanın önemine istinaden, dijital sayım yöntemi daha kısa sürede sayımı gerçekleştirip eksik kişileri bildirecektir. Dolayısıyla dijital sayım metodunun kullanılması can kayıplarını önlemek adına daha etkili sonuçlar verecektir.

DIGITAL ENUMERATION OF EMERGENCY GATHERING AREAS, USING DISTRIBUTED GROUP COMMUNICATION AND FUZZY ALGORITHM METHODS IN INDUSTRIAL FACILITIES

SUMMARY

Keywords: Distributed group communication method, fuzzy algorithm method, emergency gathering areas enumerations, digital enumeration.

In this study was done according to the law number 6331 which was adopted by the Government of the Republic of Turkey at 2012. The digital enumeration system has been dealt with instead of the classical counting process in the emergency gathering areas in medium and large scale facilities / enterprises. For the digital enumeration method, distributed group communication and fuzzy logic methods are used.

Radio frequency identification systems should be used when a distributed group communication system is established. Digital enumeration operations were performed by reading biometric fingerprints with Viridi brand AC-2100 devices. The application with C # program has been written to obtain the results. The results show the number of employees in the emergency gathering areas and the number of employees who could not count. According to the facility size and the number of employees, how many digital enumeration devices should be positioned at the facility by using fuzzy logic method. The equation path equals to velocity multiplied by time was used. When fuzzy logic simulation is described, Matlab program is used in the solution by using the Mamdani method and the centroid method.

According to the results, the digital enumeration method of medium and large scale facilities has yielded faster than classical enumeration method. In case of emergency, the use of this digital enumeration method will give more effective results in order to avoid loss of life, since the digital enumeration method will carry out the counting in a shorter time and report the missing persons in order to give importance to the time spent to reach the missing persons.

BÖLÜM 1. GİRİŞ

Türkiye Cumhuriyeti Devleti 2012 yılında 6331 sayılı İş Sağlığı ve Güvenliği kanununu kabul etmiştir. 6331 sayılı kanunun en önemli özelliği proaktif yaklaşıma sahip olmasıdır. Kanunda kaza, acil durum olmadan önce alınması gereken tedbirler için çalışmaların yapılması maddelere bağlanmıştır. Bu kanunun on birinci maddesi Acil durum planları, yangınla mücadele ve ilk yardım hakkında, on ikinci maddesi ise Tahliye hakkında yapılması gerekenlerden bahsetmiştir. Ayrıca 2013 yılında 6331 sayılı kanuna istinaden İşyerlerinde Acil Durumlar Hakkında Yönetmelik kabul edilmiştir. Bu yönetmelikte geçen beşinci maddede ise İşverenin yükümlülüklerinden bahsedilmektedir. Yönetmelikte geçen beşinci maddenin “ç” bendinde “Acil durum planlarını hazırlar ve tatbikatların yapılmasını sağlar” konusu geçmektedir.

Türkiye’deki tüm işverenler bu kanuni zorunluluklara uymalıdır. Dolayısıyla çalışanlar, işveren tarafından görevlendirilmiş ve eğitim verme yetkisine sahip kişilerden acil durumlar için eğitim alır. Her yıl planlı olarak düzenlenen tatbikatlar sayesinde, çalışanların acil durumlar esnasında nasıl davranmaları gerektiği hakkında pratik yapmaları sağlanır.

Tatbikatlarda genel işleyiş şu şekildedir. İşveren tarafından tatbikat günü ve saati belirlenir. Tüm çalışanlara, tatbikat yapılacağı hakkında günler öncesinden haber verilir. İşveren gerekli gördüğü durumlarda imza karşılığı tatbikat günü ve saatini haber verdiğini belgelemek isteyebilir. İlgili gün ve saatte alarm verilir. Tüm çalışanlar, kargaşa oluşturmadan, acil durum toplanma bölgesinde/bölgelerinde toplanır. Acil durum toplanma bölgesi saha sorumlusu sayım yapar. Sayım sonuçlarını işverene veya koordinasyon merkezine bildirir. Bildirimlerin sonuçlarına göre tesiste/işletmede kayıp, eksik kişiler varsa, bu kişilerin tespit edilmesi ve en son

görüldükleri bölge/çalıştıkları bölge içerisinde arama yapılması için arama kurtarma ekipleri gönderilir.

Planlı olarak yapılan tatbikatlarda karşılaşılan ilk ve en büyük problem tesiste/işletmede bulunan kişilerin sayımıdır. Planlı tatbikatlar esnasında bile bu sayımlarda kargaşa oluşmakta ve sonuçlar uzun süre boyunca netlik kazanmamaktadır. Özellikle orta (250'den az çalışan) ve büyük ölçekli işletmelerde çalışanların yeterli sürede sayımının yapılması ve sonuçların toplanması önem arz etmektedir. Tesis veya işletmelerde tatbikata katılan kişilerin sayım sonuçları 5 dakika içerisinde bitirilmelidir. Sayımda olmayan ve eksik olarak belirlenen çalışana acil müdahalenin yapılması için ilgili bölgeye hemen arama ve kurtarma ekiplerinin sevk edilmesi gereklidir.

Bu tezde, acil durum toplanma bölgelerinde kurulan dijital sayım cihazları ile oluşturulan sistem sayesinde tesiste/işletmede çalışan tüm personelin sayımının yapılması, eksik kişilerin belirlenmesi ve kişilerin buldukları bölgelerin çıkartılması konusu çalışılmıştır. Çalışmada dijital sayım için biyometrik okuyuculu RFT (VIRDI AC-2100) cihazlar kullanılmıştır. Tesiste/işletmede kurulacak cihazların dağılımı dağınık grup haberleşme sistemi ile yapılması planlanmıştır. Dağınık grup haberleşme yöntemi "kullanıcılarına tek ve tutarlı bir sistem olarak görünen bağımsız bilgisayar veya aygıtlardan oluşan bir topluluk" Tanenbaum, A.S. and Van Steen, M. (2006) kullanılarak, tesiste/işletmede çalışan personele acil durum esnasında, tanımlı olduğu bölgede sayıma katılma zorunluluğu ortadan kaldırılmıştır. Bu yöntem sayesinde sayım süresinin daha kısa sürmesi planlanmaktadır. Dijital sayım cihazları, Ethernet çıkışlarından RJ45 çıkışları ile CAT 5 kablo ile bir hub/router/switch bağlanarak sunucuya bağlanmaktadır. Sunucu üzerinde, C# ile yazılmış olan uygulama çalıştırılarak gelen veriler okunarak algoritma ile oluşturulan gerekli arama ve sıralamalar çalıştırılmaktadır. Sıralama sonuçlarından, çalışanların bulunduğu veri tabanından eksik kişiler aranıp liste halinde yayımlanmaktadır. Bulanık algoritma metodu sayesinde, sahaya kurulacak olan cihazların uzaklıkları ve sayıma katılan çalışanların cihazlara olan uzaklıklarının uygunluk sonuçları hesaplanmıştır.

Tez konusunun bir prototipi; 4 adet RFT cihaz, 1 switch, yeterli sayıda kablo, bir sunucu bilgisayar, C# uygulaması ve MATLAB programı kullanılarak hazırlanmıştır. Yapılan dijital sayımlar sonucunda 55 kişinin sayımı tek cihaz üzerinden, yaklaşık 4 dakikada biteceği hesaplanmıştır. Ayrıca bulanık algoritma metodu kullanılarak cihazların birbirlerine uzaklıkları ve personelin cihazlara ne kadar uzakta olması gerektiği sonucu elde edilmiştir.

Teknolojik gelişmeler her alanda kolaylıklar sağlamakta ve genel güvenliğe her geçen gün daha fazla katkı sunmaktadır. Yangınlar toplumları tehdit eden en önemli felaketlerden biri olarak varlıklarını sürdürmekte ve onunla örgütlü mücadele etmeyi bütün toplumlar kendine görev saymaktadır.

Yangın güvenliği kendi içerisinde birçok aşamayı barındıran bir süreçtir. Yangını önleme, yangından korunma, yangını baskılama veya yangın anında tahliye bu sürecin en önemli aşamalarıdır. Teknolojik gelişmeler yangın güvenliğinin her aşamasında katkı sunmakta ve yangından doğabilecek can ve mal kayıplarını en aza indirmeye çalışmaktadır.

Çalışma insanların tehlike bölgesinden tahliyesi sonrasında, tahliye işleminin tamamen gerçekleşip gerçekleşmediğini tespit etmeye odaklanmaktadır. Bu çerçevede dağınık grup iletişimi devreye girmektedir ve birçok metodundan bahsetmek mümkündür.

Belki teknolojik gelişmelerden önce de tahliye işleminin güvenli bir şekilde yapılıp yapılmadığı kontrol ediliyordu ancak kontroller daha çok manuel metotlarla yapılıyordu ve zaman alıyordu. Araç ve bilişim teknolojisi yeni imkânlar sundu ve daha karmaşık durumlarda dahi tahliye sonrası insan davranışları kontrol edilebilmektedir.

Çalışmada bilgisayar destekli dağınık grup haberleşme sistemlerini kullanarak tahliye sonrası kişi sayımı için ölçüm yapılmıştır. Ölçüm yapmadan önce dağınık grup haberleşme sistemleri hakkında detaylı bilgilere yer verilmiştir. Bulanık mantık

da yine bu çerçeve de etraflıca tanımlanmıştır. Çalışmada bireysel hareket hızı (v) öneme sahiptir ve hız vektörünün miktarı ve yönü ile karakterize edilmektedir. Tahliye yolunun açık olması durumunda hareket hızı normal seviyesine ulaşır. Bu durumda ortalama ölçülere sahip bir yetişkinin hareket hızı 1,2 m/s - 1,4 m/s olarak hesaplanır (Schneider ve Kirchberger, 2007). Çalışmaya konu alan ortalama ölçülere sahip yetişkin insanların çalıştığı bir işletme olduğu için çalışmada hareket hızı 1,3m/s olarak alınmıştır.

BÖLÜM 2. KAYNAK ARAŞTIRMASI

2.1. Dağınk Grup Haberleşme Sistemleri

Bu tezde konu olan metotlardan bir tanesi dağınk grup haberleşme sistemleridir. Bu sistem cihazların birbirinden bağımsız olarak çalışmasını ve ana sunucu ile haberleşme işlemini sağlamaktadır. Sistemin işleyişi sunucu tarafından yürütülmektedir. Sunucu UNIS programı aracılığı ile terminallere mesaj göndermektedir. Mesajı içeriğinde sisteme yeni dâhil olan yeni kişilerin terminallere yüklenmesi sağlanmaktadır. Ayrıca mesaj gönderimi esnasında çalışmayan veya iletişimi kopmuş olan terminal varsa tespit edilmektedir.

Dağınk grup iletişimi sistemsel olarak iki farklı şekilde kurulabilir. Birinci tip olarak kapalı grup iletişim sistemi tasarımıdır. Bu tasarım üyeleri sabittir ve içerisine yeni üye kabul etmez. Grup üyeleri sadece kendi içerisinde haberleşir. Grup üyeleri haricinde kalan tüm diğer işlemler yok sayılır. Bu tip bir yapı kurulmuş olursa, sayıma girecek çalışanların sayım alanları belirli olacaktır. Sayıma katılacak kişiler bu alan haricinde hiçbir alanda sayıma dâhil olamayacaklardır. Dolayısıyla orta ve büyük ölçekli tesis/işletmelerde çalışma alanından farklı bir alanda olan personel, kayıtlı olduğu sayım alanına gitmek için vakit kaybedecektir. Hayati tehlikesi olmasa bile arama ve kurtarma ekiplerine kayıp olarak bildirilecek ve asıl müdahaleye ihtiyacı olan kişilere erişilmesini geciktirecektir. Diğer bir metot olan açık grup iletişim sistemi bu tez çalışmasında esas kabul edilmiştir. Sahada bulunan dijital sayım terminalleri açık grup iletişim yöntemi sayesinde farklı terminale kayıtlı olan personelin sisteme okutulmasını kabul etmektedir. Farklı bir alan terminaline kayıtlı olan personel kendini sisteme okuttuktan sonra asıl kayıtlı olduğu alana gider ve kendini tekrar sisteme okutabilir. Sistem en son okuttuğu veriyi kabul eder ve önceki verileri göz ardı eder. Bu tip yapı sayesinde, çalışan arama kurtarma ekiplerinin listesine çıkmayacaktır. Böylece hatalı aramaların önüne geçilecektir.

Sistem mimarisi üç farklı şekilde tasarlanabilir. Birincisi merkezi yapılmı mimari tasarımıdır. İkincisi merkezi olmayan mimari tasarımıdır. Üçüncüsü ise hibrit yapılmı mimari tasarımıdır. Bu tez çalışmasında birinci tip olan merkezi yapılmı mimari tasarım kabul edilmiştir. Sistemin bir sunucu tarafından yönetilmesi esas alınmıştır. Sunucu acil durum komuta merkezi içerisinde yapılandırılacak ve gelen veriler burada değerlendirilecektir. Değerlendirilen veriler arama ve kurtarma ekiplerine bildirilecek ve en kısa sürede müdahale edilmesi istenecektir.

Merkezi yapılmı mimari tasarımın oluşturulması esnasında yaşanabilecek bazı zorluklar bulunmaktadır. Bunlardan birincisi çok noktalı haberleşmenin zamana bağlı olmasıdır. Sisteme okutulan kişilerin beş dakika içerisinde tamamlanması istenmektedir. Bu sebepten dolayı tesisteki/işletmedeki toplam çalışana göre dijital sayım cihaz sayısı belirlenmeli ve çalışanların yoğun olduğu bölgelerde daha fazla cihaz olacak şekilde konumlandırma yapılmalıdır. Ayrıca acil durum toplanma alanları, çalışanların sürekli ve kolay erişebileceği noktalarda konumlandırılmalıdır.

İkinci zorluk ise terminallerin adreslemesi işlemidir. Bu işlem için rakamlardan oluşan uzun adresleme yerine daha kolay ve anlaşılabilir olan isimlendirme yöntemi seçilmiştir. Buna benzer bir yapı internet sayfalarının adresleme yöntemidir. Adreslemelere göre ana sunucudan terminallere kısa mesaj iletim yöntemi kullanılarak iletişim kurulması sağlanmaktadır. Mesajlaşma tüm terminallere yapılabilir veya tek bir terminale de yapılabilir. Kısa mesajlaşma sayesinde çalışmayan terminal bölgesinin kontrolünün yapılması sağlanabilir.

Üçüncü zorluk olarak terminallerin ana sunucu ile olan saat senkronizasyonudur. Saat senkronizasyonu için her terminalin saatinde farklılık göstermesi sonucunda tesis/işletme çalışanın hangi saatte hangi bölgede kendini sayım sistemine dâhil ettiği problemi ortaya çıkmaktadır. Bu sorunun ortadan kaldırılması için bu çalışmada gelen mesajlar ana sunucudan aldıkları saate göre sisteme kayıt edilmektedir. Dolayısıyla saat senkronizasyon sorunu ortadan kaldırılmaktadır. Fakat tedbir amaçlı olarak her terminalin saat ayarı ortak olarak kullanılan bir saate göre aylık olarak yeniden düzenlenmelidir. Bu kontrol ve düzenleme sayesinde hem

terminalin çalışma durumu hem de yaşanabilecek olan saat senkronizasyon sorununu en aza indirilir.

Dördüncü zorluk verilerin yedeklenmesi ve güvenlikleridir. Bu zorluk aşılmasında sistemin belirli sürelerde bir kopyasının alınması sağlanmıştır. Yedek olarak ayrı bir yedekleme ünitesine aktarımı sağlanmıştır. Aktarımdan sonra dosya şifrelenmiş ve C# ile yazılan uygulama ekranına şifre aktarılmıştır.

Beşinci ve son zorluk olarak tespit edilen durum ise yapısal olarak sahada bulunan terminal sayısı ile ilgilidir. Sahadaki terminal sayısı fazla olacak ise ve ana sunucu ile cihazlar arası mesafeler uzak olacak ise, ana sunucu ile konuşan ara istasyonların oluşturulması gerekebilir. Bu yapının kurulması sayesinde ara istasyon sunucuları hem kendi gruplarının mesajlarını yedeklemiş olacaktır, hem de ana sunucu üzerine aşırı yüklenme sorununu ortadan kaldıracaktır. Ayrıca ara sunuculara hitaben ayrı alt uygulamalar C# ile yazılıp kendi gruplarının sonuçlarının çıkartılması sağlanabilir.

Dağınık grup haberleşme sistemlerine girmeden önce bir kısım kavramlara açıklık getirmek konun daha iyi anlaşılması adına önemlidir. Grup, ortak ilgi alanlarını paylaşan bir kullanıcı topluluğudur. Grup temelli faaliyetler süreklidir ve kendi aralarında iletişim gerekliliği vardır. Grup haberleşmesi ise gruplar arası gerekli olan iletişimin hızlı ve düzenli bir şekilde tesis edilmesidir.

Dağınık haberleşme sistemlerine konu birçok grup türü vardır:

- Açık grup (open group): Bir marketin müşterileri gibi herkes katılabilir, üyelik sınırsızdır.
- Kapalı grup (closed group): Bir sınıfın öğrencileri gibi üyelik sınırlıdır ve yeni üyelik kapalıdır.
- Eşler arası grup (peer-to-peer-group): Bir departman içerisindeki aynı statüye sahip kişilerin oluşturduğu gruptur.
- Hiyerarşik gruplar (hierarchical groups): Bir ya da daha fazla üyenin, diğerlerinden ayrıldığı gruplardır. Bir şirketin yönetim kadrosu ile çalışanları arasındaki grup ilişkisi buna örnek gösterilebilir.

2.1.1. Dağınık grup haberleşme sistemi çeşitleri

Dağınık grup haberleşme sistemleri ile alakalı literatürde birçok tanım vardır. Ancak tanımlar arasında ciddi bir benzerlik bulunmamaktadır. Her tanım sistemi farklı yönlerinden ele almaktadır. Fakat bütün tanımların ortak noktasını barındırması bakımından aşağıdaki tanım kullanılabilir:

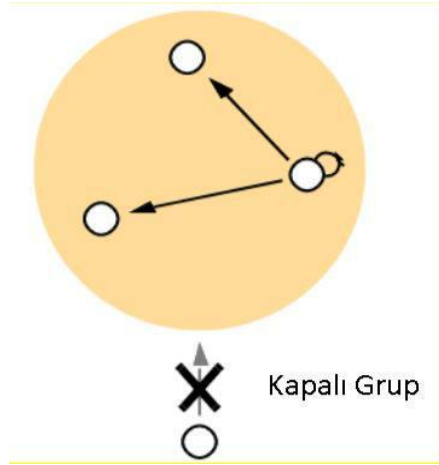
“Dağınık bir sistem, kullanıcılarına tek ve tutarlı bir sistem olarak görünen bağımsız bilgisayar veya aygıtlardan oluşan bir topluluk olarak tanımlanır. (A distributed system is a collection of independent computers or devices that appears to its users as a single coherent system.)” Tanenbaum, A.S. and Van Steen, M. (2006)

Dağınık grup haberleşme sistemlerinde, birden çok süreç, altta yatan ağlarda mesaj alışverişi yaparak birbirleriyle iş birliği yapmaktadır. Dağınık uygulamaları gerçekleştirmek için, birbirleriyle iş birliği yapmaları için bir grup süreci gerekmektedir. Dağınık bir grupta, her bir üye süreci, merkezileştirilmiş koordinasyon olmaksızın diğer süreçlerle birlikte akran ve doğrudan mesajlaşmaktadır. Yani, her süreç bir gruptaki diğer işlemlere mesaj göndermektedir. Her işlem güvenilir ve verimli olmalıdır. Ayrıca, bir grubun ölçeklenebilir olması gerekmektedir.

Yukarıda grup çeşitlilikleri hakkında kısa bilgi verilmiştir. Dağınık grup haberleşme sistemlerini kapalı/açık ya da çakışan/çakışmayan gruplar şeklinde tasnif edilebilir. Bu çalışma açık grubu temsil ettiğinden açık ve kapalı grup terminolojisinin kısaca tarif edilmesi faydalı olacaktır.

2.1.1.1. Kapalı grup

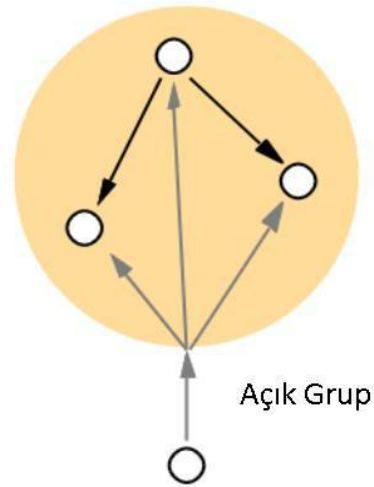
Sadece grubun üyeleri çoklu noktaya gönderim yapabiliyorlarsa kapalı dağınık grup haberleşme sisteminden bahsedilebilir. Şekil 2.1.'de örnek gösterim verilmiştir.



Şekil 2.1. Kapalı grup sistematığı

2.1.1.2. Açık grup

Açık grupta kapalı grubun aksine üye girişi her zaman mümkündür. Çalışmada benzetim yaparken açık grup prensibi üzerinde hareket edilmiştir. Çünkü tahliye yapılan işletmede çalışanların dışında ziyarete gelenlerde göz önüne alınarak bir çalışma yürütülmüştür. Şekil 2.2.'de örnek gösterim verilmiştir.



Şekil 2.2. Açık grup sistematığı

2.1.2. Sistem tasarımı

Bu bölümde merkezi yapılı, merkezi olmayan ve hibrit yapılı mimari tasarımları hakkında bilgi verilmektedir.

Dağınık sistemler genellikle bileşenlerin birden fazla makineye dağılmış karmaşık yazılım parçaları olarak tanımlanmaktadır. Karmaşıklıklarına çözmek için, bu sistemlerin düzgün bir şekilde düzenlenmesi çok önemlidir. Dağınık bir sistemin organizasyonunun nasıl görülebileceği konusunda farklı yollar vardır, ancak bariz olan, yazılım bileşenlerinin toplanmasının mantıksal organizasyonu ile diğer fiziksel gerçekleştirme arasında bir ayırım yapmaktır. Dağınık sistemlerin organizasyonu çoğunlukla sistemi oluşturan yazılım bileşenleri ile ilgilidir. Bu yazılım mimarileri, çeşitli yazılım bileşenlerinin nasıl düzenleneceğini ve nasıl etkileşimde bulunmaları gerektiğini anlatmaktadır. Bu bölümde, ilk olarak, (dağıtılmış) bilgisayar sistemlerine yönelik yaygın olarak uygulanan bazı yaklaşımlara dikkat çekilmiştir. Dağınık bir sistemin gerçekten ortaya çıkması, yazılım bileşenlerini gerçek makinelere yerleştirilmesini gerektirir. Bunu yapmak için çok farklı seçenekler vardır. Bir yazılım mimarisinin son örneği ayrıca sistem mimarisi olarak da anılır.

2.1.2.1. Sistem mimarisi

Sistem mimarisi; dağınık sistemin yapısını, davranışını ve biçimselliğini tanımlayan kavramsal modeldir. Dağınık sistemlerde bir mimari tarz kavramı oldukça önemlidir. Böyle bir tarz, bileşenler açısına, bileşenlerin birbirine bağlanma biçimine, bileşenler arasında veri alışverişi yapılmasına ve son olarak, bu bileşenlerin bir sistemde ortak olarak nasıl yapılandırıldığına bakılarak formüle edilir. Bir bileşen, çevresi içinde değiştirilebilen iyi tanımlanmış gerekli ve sağlanmış ara yüzlere sahip modüler bir birimdir (OMG, 2004b). Aşağıda belirtildiği gibi, dağıtılmış sistemlerin bileşenleri hakkındaki en önemli konu, ara yüzlerine saygı gösterilmesi şartıyla değiştirilebilmesidir. Bağlantı noktası ise, bileşenler arasında genellikle iletişim, koordinasyon veya iş birliğine aracılık eden bir mekanizma olarak tanımlanır (Mehta ve ark., 2000; ve Shaw ve Clements, 1997). Örneğin, bir bağlantı noktası; (uzak) prosedür çağruları, mesaj geçişi veya akan veriler gibi mekanizmalarla oluşturulabilir.

Bileşenleri ve konektörleri kullanarak, çeşitli konfigürasyonlar elde edilebilir. Bu çalışmada mimari sistemler olarak sınıflandırılmaktadır. Günümüze kadar, dağınık sistemler için birçok çeşit ortaya atılmıştır, içlerinde en önemlileri şunlardır:

- Katmanlı mimariler (Layered architectures)
- Nesne-tabanlı mimariler (Object-based architectures)
- Veri merkezli mimariler (Data-centered architectures)
- Olay-tabanlı mimariler (Event-based architectures)

Katmanlı mimarilerde sistem bir dizi katmana veya gruba ayrıştırılır. Bir katman, altındaki katmanlardan hizmet alır ve üzerindeki katmanlara hizmet sunar. Tüm mimari stilleri içerisinde katmanlı stil en çok kullanılan stildir çünkü katmanlar son derece yüksek uyumludur ve bilgi saklamayı artırır. Ayrıca, üzerlerindeki katmanlarla yüksek derecede bağlı olmadıklarından toplam bağlılık seviyesi düşüktür. Katmanlar programı parçalara ayırmayı kolaylaştırır, karmaşıklığı azaltır. Katmanları değiştirmek, onarmak veya tüm katmanı başkasıyla değiştirmek oldukça kolaydır.

Nesne-tabanlı mimaride her nesne bir bileşen olarak tanımladığımız şeye tekabül eder ve bu bileşenler bir uzaktan prosedür çağırma mekanizması ile bağlanır. Şaşırtıcı olmayan bir şekilde, bu yazılım mimarisi yukarıda anlatıldığı gibi istemci-sunucu sistemi mimarisine uymaktadır. Katmanlı ve nesne-tabanlı mimariler hala büyük yazılım sistemleri içinde en önemli stilleri oluşturmaktadır (Bass ve ark., 2003).

Veri merkezli mimariler, işlemlerin ortak (pasif veya aktif) bir veri deposu ile iletişim kurduğu fikri etrafında gelişir. Dağıtık sistemler için bu mimarilerin katmanlı ve nesneye dayalı mimariler kadar önemli olduğu iddia edilebilir. Örneğin, birçok iletişimin uygulamaları içinde bütün iletişimin dosyalar üzerinden gerçekleştiği dağıtık dosya sistemi üzerinden geliştirilmiştir. Aynı şekilde, Web tabanlı dağıtılmış sistemler, büyük ölçüde veri merkezlidir. Yani işlemler, paylaşılan Web tabanlı veri hizmetlerini kullanarak iletişim kurarlar.

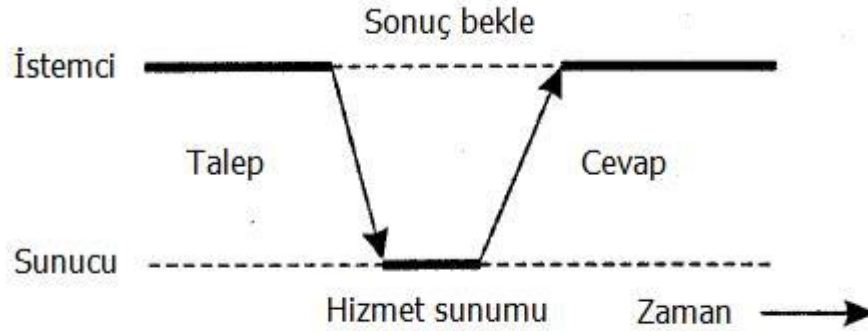
Olay odaklı mimari (EDA), bir sistemin durumundaki önemli değişikliklerin (olaylar olarak bilinir) üretimini, algılanmasını ve bu sisteme tepki göstermesini ve buna tepki verilmesini teşvik eden bir yazılım mimarisidir. Bu, birbirine kısmen bağlı

yazılım bileşenleri ve hizmetleri arasında olayları ileten uygulamalar ve sistemlerin tasarımı ve uygulanması yoluyla gerçekleştirilir (Eugster ve ark., 2003).

Bu tezde, dijital sayımı gerçekleştiren cihazlar için uygulama olarak nesne tabanlı mimarı sistemi kullanılmıştır. İstemci ve sunucu tipi sistemlere en uygun yöntem nesne tabanlı mimari yapıdır.

2.1.2.1.1. Merkezi yapılı mimari tasarım

Araştırmacılar ve uygulayıcılar arasında dağıntık sistemler hakkında fikir birliği bulunmamasına rağmen, hemfikir oldukları bir konu vardır: istemciler hizmet sunucularında hizmet talep etmektedirler. En temel istemci-sunucu modelinde, dağıntık bir sistemdeki işlemler iki (muhtemelen çakışan) gruba ayrılır. Bir sunucu, bir dosya sistemi veya veri tabanı hazırlama gibi belirli bir hizmet sağlayan taraftır. Bir istemci ise bir sunucudan bu konularla ilgili bir istekte bulunan ve sonradan sunucunun vereceği cevabı bekleyen taraftır. Aynı zamanda talep-cevap davranışı olarak süreci olarak da bilinen bu istemci-sunucu etkileşimi, aşağıdaki Şekil 2.3.'de gösterilmiştir.



Şekil 2.3. Bir istemci ile sunucu arasındaki genel etkileşim (Tanenbaum ve Van Steen, 2006)

Bir istemci ve bir sunucu arasındaki iletişim, birçok yerel-alan ağında olduğu gibi alttaki network oldukça güvenilir olduğu zamanlarda basit bağlantısız bir protokol vasıtasıyla uygulanabilir. Bu durumlarda, bir istemci bir hizmet istediğinde, o gerekli olan verilerle birlikte bu hizmet isteğini basit bir şekilde paketleyerek sunucuya bir

mesaj gönderir. Ardından, mesaj sunucusu da gelen isteğe cevap verecek ve cevapları bir mesaj içerisinde istemciye gönderecektir.

Uygulama Katmanı (Application Layering)

İstemci-sunucu modeli uzun yıllardan beri birçok tartışmaya konu olmuştur. Şaşırtıcı olmayan bir şekilde tartışılan ana konulardan biri, bir istemci ile bir sunucu arasında net bir ayrımın nasıl çizilmesi gerektiğidir. Çünkü genellikle net bir ayrım yoktur. Örneğin, dağınık bir veri tabanı için bir sunucu sürekli olarak bir istemci gibi hareket edebilir, çünkü veri tabanı tablolarını uygulamaktan sorumlu olan farklı dosya sunucularına istekleri iletir. Böyle bir durumda, veri tabanı sunucusu aslında işlem sorgularından daha fazlasını yapmaz. Bununla birlikte, birçok istemci-sunucu uygulamasının veri tabanlarına kullanıcı erişimini desteklemeye yönelik olduğu düşünüldüğünde birçok kişi, aşağıdaki üç seviye arasında bir ayrım yapmayı tercih etmiştir.

- Kullanıcı ara yüzü seviyesi (The user-interface level)
- İşlem seviyesi (The processing level)
- Veri seviyesi (The data level)

Kullanıcı ara yüzü seviyesi, görüntü yönetimi gibi kullanıcı ile doğrudan ara yüzle bağlantı kurmak için gerekli olan her şeyi içerir. İşlem seviyesi, tipik olarak uygulamaları içerir. Veri seviyesi ise, üzerinde çalışılan gerçek verileri yönetir.

Hazırlanan bu çalışmada, uygulama katmanı sürekli olarak açık tutulmaktadır. Uygulama katmanı durdurulur veya sonlandırılırsa, tüm sistem etkisiz hale geçecektir. C# ile yazılan uygulama sayesinde, uygulama katmanının çalışır vaziyette olup olmadığının kontrolü yapılmaktadır.

2.1.2.1.2. Merkezi olmayan mimari tasarım

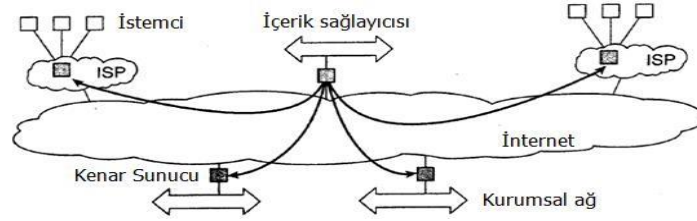
Çok aşamalı istemci-sunucu mimari tasarımları, bölünen uygulamaların bir kullanıcı ara yüzüne, işlem bileşenlerine ve bir veri düzeyine doğrudan bir sonucudur. Farklı katmanlar, uygulamaların mantıksal organizasyonuna doğrudan karşılık gelir. Birçok iş ortamında dağınık işlem, bir istemci-sunucu uygulamasını çok katmanlı bir mimari olarak organize edilmesiyle eşdeğerdir. Bu dağıtım türünü dikey dağılım olarak tanımlanmaktadır. Dikey dağılımın karakteristik özelliği, mantıksal olarak farklı bileşenleri farklı makinelere yerleştirmekteki başarıdır. Bu terim, dağınık ilişkisel veri tabanlarında kullanılan dikey parçalanma kavramıyla ilgilidir. Bu, tabloların sütun şeklinde bölünmesi ve daha sonra birden fazla makineye dağıtılması anlamına gelir (Oszu ve Valduriez, 1999).

2.1.2.1.3. Hibrit yapılmimari tasarım

Hibrit yapılmimari tasarım olarak adlandırılan bu özellikli sistemlerde, istemci-sunucu çözümleri merkezi olmayan tasarımlarla birleştirilir.

Kenar- sunucu sistemleri (Edge-server systems)

Hibrit yapılmimari tasarıma göre düzenlenmiş dağınık sistemlerinin önemli bir sınıfını kenar-sunucu sistemleri oluşturur. Bu sistemler sunucuların iletişim ağının “en kenarında” olduğu internet üzerinde bulunurlar. Bu kenar, bir “İnternet Hizmeti Sağlayıcısı- İHS (Internet Service Provider-ISP)” örneğinde olduğu gibi kurumsal ağlar ve gerçek İnternet arasındaki sınır tarafından oluşturulur. Aynı şekilde, evdeki son kullanıcıların İHS’leri aracılığıyla İnternet’e bağlandığı yerlerde, İHS’nin İnternet’in kenarında yer aldığı düşünülmektedir. Bu, Şekil 2.4.’te gösterildiği gibi genel bir organizasyona yol açar.



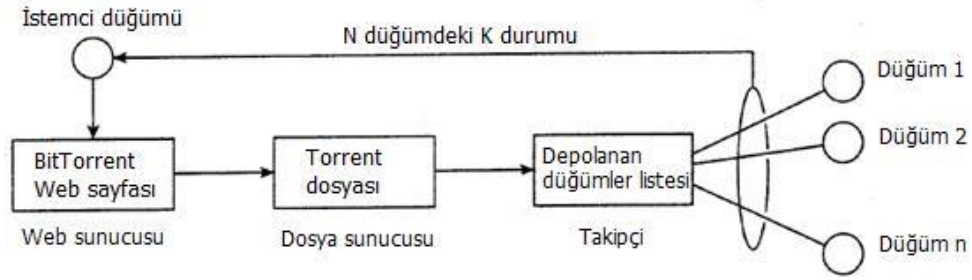
Şekil 2.4. İnterneti kenar sunucularından oluşan bir koleksiyonla görüntüleme (Tanenbaum ve VanSteen, 2006)

Son kullanıcılar veya genel olarak istemciler, bir kenar sunucusu aracılığıyla internete bağlanır. Kenar sunucusunun ana amacı, muhtemelen filtreleme ve kod dönüştürme işlevlerini uyguladıktan sonra içerik sunmaktır. Bunun sebebi, içerik ve uygulama dağıtımını optimize etmek için bir dizi kenar sunucularının kullanılabilmesidir. Temel model, belirli bir kuruluş için, bir kenar sunucusunun tüm içeriğin ortaya çıktığı bir başlangıç sunucusu görevi görmesidir. Bu sunucu, Web sayfalarını ve benzerlerini çoğaltmak için diğer kenar sunucularını kullanabilir (Leff ve diğerleri, 2004; Nayate ve diğ., 2004; ve Rabinovich ve Spatscheck, 2002).

İşbirlikçi dağıtık sistemler (Collaborative Distributed Systems)

Hibrit yapılar, görünür bir şekilde özellikle işbirlikçi dağıtılmış sistemlerde bulunmaktadır. Bu sistemlerde işleme başlarken halledilmesi gereken ana, geleneksel istemci-sunucu düzenin hangi sıklık içinde kaldığıdır. Bir düğüm sisteme katıldıktan sonra, iş birliği için tamamen merkezi olmayan bir plan kullanabilir.

Tanımları biraz daha somutlaştırmak için, önce BitTorrent dosya paylaşım sistemini düşünülebilir (Cohen, 2003). BitTorrent bir bireylerarası dosya indirme sistemidir ve çalışma biçimi Şekil 2.5.'de gösterilmiştir. Buradaki temel fikir, bir son kullanıcı bir dosya ararken, indirilen parçalar bir araya getirilinceye kadar dosyanın tüm parçalarını diğer kullanıcılardan indirmesidir. Asıl tasarım hedefi, iş birliğini sağlamaktır. Çoğu dosya paylaşım sisteminde, katılımcıların önemli bir kısmı sadece dosyaları indirirler; aksi takdirde başka hiçbir şeye katkıda bulunamazlar (Adar ve Huberman, 2000; Saroiu ve ark., 2003; ve Yang ve ark., 2005). Bu amaçla, bir dosya yalnızca indirme istemcisi bir başkasına içerik sağladığında indirilebilir.



Şekil 2.5. Bittorrent çalışma prensibi (Tanenbaum ve Van Steen, 2006)

Bu tezde konusu geçen dijital sayım cihazlarının sunucuya olan uzaklıkları büyük önem arz etmektedir. Uzaklığa bağlı olarak ilave ara sunucuların yapılandırılması ve bu sunucuların da ana sunucu ile konuşması gerekebilir. Bu durumda merkezi yapıli mimari sistem terk edilip, hibrit yapıli mimari sistemin işbirlikçi dağılık sistemler yöntemine geçilmesi, dijital sayım işleminin daha hızlı şekilde tamamlanmasına sebep olacaktır.

2.1.3. Yazılım mimarisi

Yazılım mimarisi, bir sistemin ana yapısını gösterdiği ve her bir paydaşın ihtiyacını dikkate aldığı için yazılım geliştirme sürecinin temel taşlarından biridir. Yazılım mimarisi geliştirilen uygulamaların maliyetlerinin düşürülmesi için kullanılır. Örneğin bir model kullanarak bir uygulama geliştirildiğinde ve ara yüz bölümünde bir değişikliğe gitmeniz gerektiğinde sadece ara yüzde bulunan kodu değiştirmek yeterlidir. Genel yazılım mimarisi basit, katmanlı ve doğrudan çalıştırma olmak üzere üçe ayrılır.

Basit mimari:

Bir bilgisayarın işletim sistemi üzerinde çalışan bir programdır. Örneğin internet bağlantısız bir otomasyon programı basit mimariye bir örnektir.

Karmaşık mimari:

N-katmanlı ve çok katmanlı mimari olarak da isimlendirilen karmaşık mimari makinelerin birden fazla kullanılması ve bunların arasındaki iletişimin sağlanmasıyla oluşur. Ölçeklendirilebilirlik, esneklik ve yönetim kolaylığı en önemli avantajlarından.

Doğrudan çalıştırma:

Doğrudan çalıştırma gömülü sistemler üzerinde çalışan programlardır. Örnek olarak çamaşır makinesine 30 dakika içinde işlemi bitirmesi için kodlama yapılması gerekir.

Bu tezde, basit mimari ile yapılandırılmış ve ara yüze sahip bir uygulama geliştirilmiştir. İnternete ihtiyaç duymayan fakat kendi ağ sistemine sahip olması gereken sunucu-istemci yapısı bu uygulama sayesinde sonuçları kısa sürede ortaya çıkarmıştır.

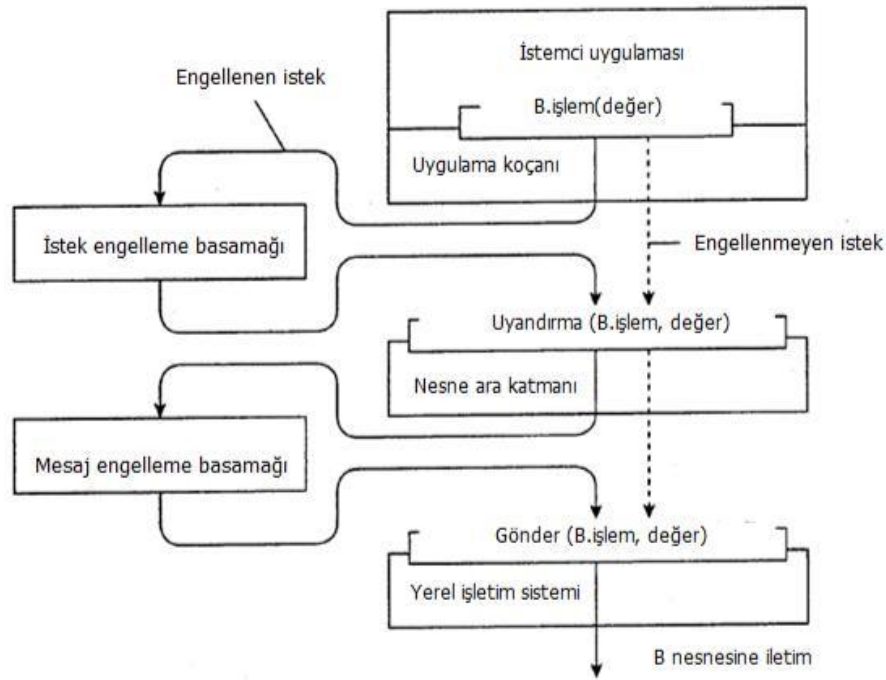
2.1.3.1. Engelleyici unsurlar

Kavramsal olarak, bir engelleyici, olağan kontrol akışını engelleyen ve diğer (uygulamaya özel) kodların gerçekleşmesine yarayan bir yazılım yapısıdır. Engelleyicileri üretken yapmak yüksek oranda uygulama çabası gerektirebilir ve bu şekilde yapmanın kısıtlı uygulanabilirlik ve basitliğe tercih edilip edilmemesi gerektiği tam belli değildir (Schmidt ve ark., 2000). Ayrıca, çoğu durumda sadece sınırlı engelleyici vasıtalarına sahip olmak yazılımın ve dağınık sistemin yönetilmesini bir bütün olarak geliştirebilir.

Daha da somutlaştırmak için, birçok nesne tabanlı dağınık sistemde engelleyicilerin desteklendiği düşünülebilir. Temel fikir basittir: A nesnesi, B nesnesine ait bir yöntemi çağırabilirken, ikincisi A'dan farklı bir makinede bulunur. Böyle bir uzaktan nesne çağırma işlemi, üç adımlı bir yaklaşımda olur. Object A, arabirimle tam olarak aynı olan yerel bir arabirim sunar. A nesnesi, B nesnesi tarafından sunulan ara yüzle

tam olarak aynı olan yerel bir arabirim sunar. A basitçe bu ara yüzde mevcut yöntemi çağırır. 2. A tarafından yapılan çağrı, genel bir nesne çağrısına dönüştürülür. 3. Son olarak, genel nesne çağrısı bir mesaja dönüştürülür ve A'nın yerel işletim sistemi tarafından sunulduğu gibi ulaşım seviyesi ağ ara yüzü ile gönderilir. Şekil 2.6.'da engelleyici unsurlar gösterilmiştir.

Yapılan bu çalışmada engelleyici unsurlara çözüm bulmak için nesne çağrısına yönelik mesajlaşma yöntemi kurgulanmıştır. Dijital sayım için kullanılan her bir cihaz nesneleştirilmiş ve bu cihazlara mesajlaşma yöntemi ile bilgi aktarımı sağlanmıştır. Ayrıca sağlıklı olarak gözlemleme yapılabilmesi için, sahada bulunan dijital sayım cihazlarının çalışıp çalışmadığı UNIS programı tarafından izlenebilmektedir. Bu izlemede isimlendirme yapılarak adresleme yöntemi geliştirilmiştir.



Şekil 2.6. Uzak nesne çağrılarını işlemek için engelleyici kullanma (Tanenbaum ve Van Steen, 2006)

2.1.3.2. Sistemin kendi kendine yönetme metodu

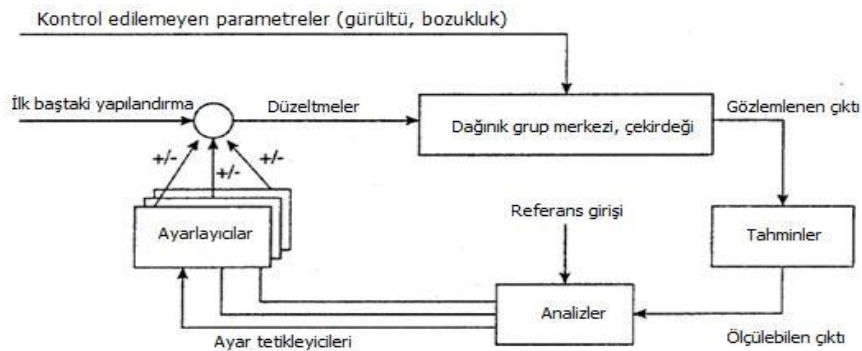
Dağınık sistemler, özellikle de içerdikleri yazılım bileşenleri yönünden değil, yürütme davranışlarını adaptasyon konusunda kendilerini ayarlayabilmelidir. Adaptasyonun otomatik olarak yapılması gerektiğinde, sistem mimarileri ve yazılım

mimarileri arasında güçlü bir etkileşim görülür. Bir yandan, dağıtılmış bir sistemin bileşenlerini, izleme ve ayarlamaların yapılabileceği şekilde organize edilmesi gerekirken, diğer taraftan, uyum sürecini yürütecek süreçlerin nerede yürütüleceğine karar verilmesi gerekmektedir.

Sistemin kendi kendini yönetme metodu, dağınık istemlerin değişimlere otomatik adaptasyon olarak izin veren yüksek seviyeli geri bildirim kontrol sistemleri demektir. Bu olgu aynı zamanda otonom hesaplama (Kephart, 2003) veya öz-yıldız(self-star) (Babaoğlu ve Toueg, 2005) sistemleri olarak da bilinir. Öz-yıldız sistemler otomatik ayarlamaların yakaladıkları farklılıkları ifade eder: kendi kendini yöneten, kendini iyileştiren, kendini düzenleyen, kendini optimize eden vb.

Geri bildirim kontrol modeli (The feedback control model)

Kendi kendini yöneten sistemlerle ilgili birçok farklı görüş vardır. Bu görüşlerin çoğunun ortak varsayımı aynıdır: Uyarlamalar bir veya daha fazla geri besleme kontrol döngüsü vasıtasıyla gerçekleşir. Bu durumda, bu döngüler vasıtasıyla organize edilen sistemler geri bildirim kontrol sistemleri olarak adlandırılır. Geri bildirim kontrolü, uzun zamandan beri çeşitli mühendislik alanlarında uygulanmaktadır ve matematiksel temelleri giderek bilgisayar sistemlerinde de kullanılmaya başlanmıştır (Hellerstein ve ark., 2004; Diao ve ark., 2005). Kendi kendini yöneten sistemler için, mimari konular başlangıçta en zorlayıcı konulardır. Bu organizasyonun arkasındaki temel fikir, Şekil 2.7.'de gösterildiği gibi oldukça basittir.



Şekil 2.7. Geri bildirim kontrol sisteminin çalışma prensibi (Tanenbaum ve Van Steen, 2006)

Bu çalışmada, geri bildirim kontrol modeli esas alınmış olup, dijital sayım için sahada kullanılan cihazlara kendini tanıtan personel, eş zamanlı olarak kontrol edilmekte ve personelin adı veya üyelik numarası cihaz üzerinde gösterilmektedir. Eşleşmeme durumunda ise veya hızlı çekme (fast back) durumunda ise cihaz sesli ve yazılı olarak personele tanınmadığı hakkında bilgi vermektedir. Personel tanınmasa bile sistemin veri tabanına bilgi düşmekte fakat personel ismi ve üyelik numarası boş gözükmemektedir. Hazırlanmış olan C# uygulaması sayesinde bu boş veriler temizlenmekte ve düzeltilmiş personel listesi programı kullanan kişiye gösterilmektedir.

2.1.4. Haberleşme

Süreçler arası iletişim (interprocess) tüm dağıtık sistemlerin merkezinde yer almaktadır. İşlemlerin farklı makineler üzerinde bilgi değiştirebilmesinin yollarını dikkatlice araştırılması. Dağıtık sistemlerde iletişim her zaman temel ağ tarafından sunulan düşük seviyeli iletiyi temel alır. Mesaj iletimi yoluyla iletişimin ifade edilmesi, dağıtılmamış platformlar için hazırlandığı gibi paylaşılan hafızaya dayalı primitifleri kullanmaktan daha zordur. Modem dağıtılmış sistemler, internette olduğu gibi güvenilir olmayan iletişim ağıyla etrafa binler hatta milyonlar işlem yayar. Bilgisayarın temel iletişim olanakları başka bir şeyle değiştirilmediği müddetçe, büyük ölçekli dağıtılmış uygulamalar geliştirmek çok zordur.

Dağıtık sistemlerde iletişim için yaygın olarak kullanılan haberleşme modelleri aşağıda incelenmiştir.

2.1.4.1. Uzaktan prosedürlü çağrı

Birçok dağıtık sistem işlemler arasında açık mesaj alışverişine dayanmaktadır. Ancak, gönder-al prosedürleri, iletişimi hiçbir şekilde gizlememektedir. Oysa dağıtık sistemlerdeki erişim şeffaflığını elde etmek için çok önemlidir. Bu problem uzun zamandır biliniyordu, ancak Birrell ve Nelson'un (1984) iletişimi kullanmanın tamamen farklı bir yolunu ortaya koydukları makalelerine kadar bu konuda çok az

şey yapılmıştır. Bu fikir herhangi bir kişinin düşünmüş olabileceği kadar oldukça basit olsa da anlaşılması çoğu zaman kolay olmamıştır.

Özetle, Birrell ve Nelson'ın yaptığı, diğer makinelerde bulunan çağrı prosedürleri programlara izin vermektir. A makinesindeki bir işlem B makinesindeki bir işlemi çağırdığında, A'daki çağrı işlemi askıya alınır ve çağrılan prosedür B üzerinde gerçekleşir. Bilgi, arayan kişiden aranan kişiye parametreler içinde gönderilebilir ve prosedür sonucuna geri dönebilir. Programlayıcıya hiçbir şekilde görünür şekilde mesaj iletilmez. Bu metot Uzaktan Prosedür Çağrısı (Remote Procedure Call) olarak da bilinir.

Uzaktan Prosedürlü Çağrı kısaca, iletilen mesajların çoğunu gizlemeyi amaçlamaktadır ve istemci-sunucu uygulamaları için idealdir. Birçok dağıtık grup uygulamasında iletişim, istemci-sunucu etkileşimini takip etmemektedir. Bu durum, mesajlar açısından düşünmenin daha uygun olduğunu ortaya çıkarmaktadır. Bununla birlikte, bilgisayar ağlarının düşük seviyeli iletişim yapıları, dağıtım şeffaflığından yoksun olmalarından dolayı pek uygun değildir. Bir alternatif, iletişimin elektronik mail sistemlerinde olduğu gibi devam ettiği yüksek seviyeli bir mesaj kuyruğu modelinin kullanılmasıdır.

2.1.4.2. Mesaj bazlı haberleşme

Uzaktan prosedür çağrıları ve uzak nesne çağrıları dağıtık sistemlerdeki iletişimin gizlenmesine katkıda bulunur. Diğer bir ifadeyle erişimin şeffaflığını artırır. Bazen her iki mekanizma da her zaman uygun olmayabilir. Özellikle, alıcı taraf bir istekte bulunduğu zaman, alternatif iletişim hizmetlerine de ihtiyaç duyulmaktadır. Aynı şekilde, içsel uzaktan prosedürü çağrıların içsel uyumlu doğası bazen başka bir şeyle değiştirilmesini gerektirmektedir. İşte bu durum mesajlaşmadır. Yani mesaj bazlı haberleşmedir. En önemli mesaj bazlı haberleşme yöntemleri şunlardır:

Mesaj odaklı geçici iletişim:

Birçok dağınık sistem ve uygulama, doğrudan aktarım katmanı tarafından sunulan basit mesaj odaklı model üzerine inşa edilmiştir. Ara katman yazılım çözümlerinin bir parçası olarak mesaj tabanlı sistemleri daha iyi anlamak için ilk olarak taşıma seviyesi soketleri üzerinden mesajlaşmaya bakılması gerekmektedir.

Berkeley soketleri:

Bu soketlerin kullanımında taşıyıcı tabakanın ara yüzünün standartlaştırılmasına özel önem verilmiştir. Bu sayede programcılar basit bir temel set aracılığıyla tüm mesajlaşma protokollerini kullanabilirler. Ayrıca standart ara yüzler bir uygulamayı farklı bir makineye yönlendirmeyi kolaylaştırırlar. Bu konuda en önemli ara yüzlerden biri AT & T tarafından geliştirilen ve ilk önce Aktarım Katmanı Ara yüzü olarak adlandırılan X10pen Aktarım Ara yüzüdür (X10pen Transport Interface/XTI). Soketler ve X10pen Aktarım Ara yüzü modelleri ağ programlaması bakımından çok benzerdir, ancak temel kümeleriyle farklılık gösterir.

Kavramsal olarak, bir soket, bir uygulamanın altta yatan ağ üzerinden gönderilecek verileri yazabilecek ve gelen verileri okuyabilecek bir iletişim bitiş noktasıdır. Bir soket, belirli bir taşıma protokolü için yerel işletim sistemi tarafından kullanılan gerçek iletişim bitiş noktası üzerinden bir soyutlama oluşturur.

Mesaj geçiş ara yüzü (The message-passing interface)

Yüksek performanslı çoklu bilgisayarların (multicomputer) gelişmesiyle birlikte, geliştiriciler son derece verimli uygulamaları kolayca yazabilmelerini sağlayacak mesaj odaklı temelleri aramaktadırlar. Bu, temellerin uygun bir soyutlama düzeyinde (uygulama geliştirmeyi kolaylaştırmak için) ve onların uygulanmasının sadece asgari yüke maruz kalması gerektiği anlamına gelmektedir. Bu konuda ve benzer konularda soketler yetersiz kabul edilmiştir.

Donanım ve platformun bağımsız olma ihtiyacı en sonunda mesaj geçişi için bir standardın tanımına yol açmıştır. Basitçe Mesaj Geçiş Ara yüzü olarak adlandırılan bu iletişim sistemi paralel uygulamalar için tasarlanmıştır ve geçici iletişim için uyarlanmıştır. Alttaki yatan ağın doğrudan kullanılmasını sağlar. Ayrıca, işlem çökmeleri veya ağ bölünmelerinin ciddi hatalar olduğunu varsayar ve otomatik kurtarma gerektirmez.

Bu çalışmada mesaj bazlı haberleşme yönteminin mesaj odaklı geçici iletişim modeli esas alınmıştır. Gönderilen mesajlar sunucu tarafından iletilmekte olup, tek taraflı sistem kurgusu yapılmıştır. Sistemin maliyeti uygun tutulabilmesi için bu yapı kurulmuştur. Ayrılacak bütçeye bağlı olarak sahada çalışan dijital sayım cihazlarının mesaj gönderme işlemi yapabilen modelleri seçilebilir. Bu tezde amaç mesajlaşma olmayıp dijital sayım işleminin gerçekleştirilerek, hızlı bir şekilde sonuçların elde edilmesidir.

2.1.4.3. Çok noktalı haberleşme

Çok noktalı haberleşme, yani verinin birden çok alıcıya birden gönderilmesi, dağıtık sistemlerde iletişimini en önemli başlıklarından birisidir. Bu konu uzun yıllar içinde ağ ve ulaşım seviyesi için çok sayıda teklifin uygulandığı ve değerlendirildiği ağ protokollerinin etki alanına ait olmuştur (Janic, 2005; Obraczka, 1998). Bilginin yayılması için iletişim yollarının hazırlanması bütün çözümlerde temel bir sorun olmuştur. Pratikte, bu durum çoğu zaman insan müdahalesini gerekli kılan büyük bir yönetim çabasını gerektirmiştir. Bunun yanında, tekliflerin bir araya gelmediği sürece, internet servis sağlayıcıların (İSS) çoklu noktalı haberleşmeyi desteklemede isteksiz oldukları görülmüştür (Diot ve ark., 2000).

Eşler arası (peer-to-peer) teknolojinin gelişimi ve yönetimdeki yeniliklerle birlikte iletişim yollarını kurmak daha kolay olmuştur. Eşler arası çözümler genellikle uygulama katmanında bulunduğu için birçok farklı uygulama düzeyinde çoklu gönderim teknikleri ortaya çıkmıştır. Bunlardan öne çıkanlar aşağıda belirtilmiştir.

Uygulama-düzeyi çoklu gönderim:

Uygulama düzeyi çoklu gönderimin amacı düğümlerin üstten bindirmeli ağ içine yerleşmesi ve daha sonra üyelerine bilgi yaymasıdır. Burada dikkati çeken husus, ağ yönlendiricilerinin (network routers) grupta yer almamasıdır. Sonuç olarak, bindirmeli ağ içindeki düğümler arasındaki bağlantılar birçok fiziksel bağlantıyla karşılaşabilir. Örneğin, bindirmeli ağ içindeki mesajları yönlendirmek ağ düzeyinde yönlendirmeye göre uygun olmayabilir.

Bindirmeli ağın yapımı çok önemli bir tasarım konusudur. Bu konuda iki temel yaklaşım vardır (El-Sayed, 2005). İlk olarak, düğümler kendilerini doğrudan bir ağaca yerleştirebilirler ki bu durum her düğüm çifti arasında benzersiz bir yol olduğu anlamına gelmektedir. Alternatif bir yaklaşım ise, düğümlerin içinde her bir düğümün birden çok komşuya sahip olacağı bir örgütsel ağa yerleşmesidir. Burada her bir düğüm çiftinde birden fazla yolun bulunmasıdır. Bu ikisi arasındaki ana fark, ikincisinin genellikle daha yüksek sağlamlık sağlamasıdır. Eğer bir bağlantı koparsa (örneğin bir düğüm başarısız olduğu için), tüm kaplama ağını derhal yeniden düzenlemeye gerek kalmadan bilgiyi yayma fırsatı olacaktır.

2.1.5. Bölgelerin adlandırılması (adreslemesi)

Tüm bilgisayar sistemlerinde isimler çok önemli rol oynarlar. İsimler genellikle kaynakları paylaşmak, varlıkları benzersiz bir şekilde tanımlamak, konumlara yönlendirmek vb. amaçlar için kullanılırlar. Bir ismin atıfta kastedildiği varlığa çözülebilmesi isimlendirmede önemli bir konudur. İsim çözümlemesi böylece bir işlemin isimlendirilmiş öğeye erişmesine izin verir. İsimlendirme konusunda dağınık ve dağınık olmayan sistemler arasındaki fark isimlendirme sistemlerinin uygulanma yolunda yatmaktadır.

Dağınık bir sistemde, bir isimlendirme sisteminin kendi kendine uygulanması sıklıkla birden fazla makineye dağıtılarak sağlanır. Bu dağıtımın nasıl yapıldığı, isimlendirme sisteminin verimliliği ve ölçeklenebilirliği açısından anahtar bir role sahiptir. Başlıca isimlendirme çeşitleri aşağıda belirtilmiştir.

2.1.5.1. İsimler, tanımlayıcılar ve adresler

Bir ismin gerçekte ne olduğuna daha yakından bakılması gerekmektedir. Dağınık sistem içinde bir isim, bir varlığı (entity) kastetmek için kullanılan bir bit veya karakter dizisidir. Dağınık bir sistemde bir varlık ana bilgisayarlar, yazıcılar, diskler ve dosyalar, kullanıcılar, posta kutuları, haber grupları, web sayfaları, grafik pencereler, mesajlar, ağ bağlantıları gibi kaynaklardır. Bu varlıklar üzerinde çalışılabilir. Örneğin, bir yazıcı bir belge yazdırma, yazdırılan işleminin durumunu öğrenme vb. işlemleri içeren bir ara yüz sağlar. Daha da ötesi, ağ bağlantısı gibi bir varlık veri gönderme ve alma, hizmet kalitesi parametrelerini ayarlama, durum isteme vb. işlemleri sağlar.

Sadece ismin özel bir durumu olan bir adres ise bir kaynağın erişim noktası anlamına gelmektedir. Bir erişim noktası bir varlık ile sıkı bir şekilde ilişkilendirildiğinden, bir erişim noktasının adresini ilişkili bir varlık için normal isim olarak kullanmak uygun görünebilir. Yine de bu tür bir isimlendirme genellikle çok esnek olmadığı ve dostça olmadığı için çok az yapılır. Örneğin, belirli bir sunucu şimdi daha önce çalıştığından farklı bir ana bilgisayarda çalışsın diye dağınık bir sistemi yeniden düzenlemek nadir değildir. Böylece, sunucunun daha önce çalıştığı eski makine tamamen farklı bir sunucuya yeniden atanabilir.

Adreslere ek olarak, bir varlığı benzersiz olarak tanımlamak için kullanılan isimler gibi özel muameleye tabi olan başka isimler de vardır. Bunlara kısaca tanımlayıcılar denilir. Gerçek bir tanımlayıcı aşağıdaki özelliklere sahip bir isimdir (Wieringa ve de Jonge, 1995):

- Bir tanımlayıcı en fazla bir varlığa işaret eder
- Her varlık en fazla bir tanımlayıcı tarafından belirtilir.
- Bir tanımlayıcı her zaman aynı varlığa atıfta bulunur (yani hiçbir zaman tekrar kullanılmaz).

2.1.5.2. Hiyerarşik adresleme yöntemi (Flat naming)

Yukarıda, tanımlayıcıların varlıkları benzersiz bir şekilde temsil etmek için uygun olduğunu açıklanmıştır. Çoğu durumda, tanımlayıcılar yapılandırılmamış veya hiyerarşik adresleme yöntemi olarak kastedilen basit rastgele bit dizgileridir. Bu şekilde isimlendirmenin önemli bir eksikliği ilişkili olduğu varlığın erişim noktası nasıl bulunacağına dair herhangi bir bilgi içermemesidir. Aşağıda, düz isimlerin nasıl çözüleceği ile bazı bilgiler verilmektedir.

2.1.5.2.1. Basit çözümler (Simple solutions)

İlk önce bir varlığı bulmak için iki basit çözümü düşünülür. Her iki çözüm sadece yerel alan ağlarına uygulanabilir. Yine de, bu ortamda, sadeliklerini özellikle cazip kılarak genellikle işi iyi yaparlar.

Yayın ve çoklu yayın (Broadcasting and Multicasting):

Bir bilgisayar ağında kurulu dağıtık sistemde yayın ve çoklu yayın yapmak oklukça verimlidir. Tipik olarak, bu tür vasıtalar, tüm makinelerin tek bir kabloya veya bunların mantıksal eşdeğerine bağlı olduğu yerel alan ağları tarafından sunulmaktadır. Ayrıca, yerel alan kablosuz ağları da bu kategoriye girmektedir. Böyle bir ortamda bir varlığı bulmak basittir. Varlığın tanımlayıcısını içeren bir mesaj her bir makineye yayınlanır ve her bir makinenin, o varlığa sahip olup olmadığını kontrol etmesi istenir. Sadece bir erişim noktası sunabilen makineler, bu erişim noktasının adresini içeren bir cevap mesajı gönderir.

Bu ilke, sadece bir IP adresi verildiğinde bir makinenin veri-bağlantı adresini bulmak için Internet Adres Çözümleme Protokolünde (ARP) kullanılır. (Plummer, 1982). Özünde, bir makine yerel ağda belirli bir IP adresinin sahibinin kim olduğunu sorarak bir paket yayınlar. Mesaj bir makineye ulaştığında alıcı istenen IP adresini dinleyip dinlememesi gerektiğini kontrol eder. Eğer öyleyse, örneğin Ethernet adresini içeren bir yanıt paketi gönderir.

Yönlendirme işaretleri (Forwarding pointers):

Mobil varlıkları bulmak için bir başka popüler yaklaşım, yönlendirme işaretlerinin kullanılmasıdır (Fowler, 1985). İlke basittir: Bir varlık A'dan B'ye hareket ettiğinde, arkasında A'da, B'deki yeni konumunu belirten bir referans bırakır. Bu yaklaşımın en büyük avantajı basitliğidir: örneğin geleneksel bir isimlendirme hizmetini kullanarak bir varlık tespit edilir edilmez, bir istemci yönlendirici işaret zincirini takip ederek mevcut adresi arayabilir.

2.1.5.2.2. Ev-temelli yaklaşımlar (Home-based approaches)

Yayın ve yönlendirme işaretlerinin kullanımı ölçeklenebilirlik sorunları doğurmaktadır. Yayın ya da çoklu yayını geniş ölçekli ağlarda verimli bir şekilde uygulamak zordur. Yönlendirme işaretlerinin uzun ağları, performans sorunlarına neden olmaktadır ve kırık linklere maruz kalmaktadır. Büyük ölçekli ağlarda mobil varlıkları desteklemenin popüler bir yaklaşımı bir varlığın mevcut yerini takip eden bir ev konumu kullanmaktır. Ağ veya işlem hatalarına karşı korunmak için özel teknikler uygulanabilir. Uygulamada, ev konumu genellikle bir varlığın yaratıldığı yeri bulmak için seçilmektedir.

Ev-tabanlı yaklaşıma başka bir örnek de Mobil IP'dir (Johnson ve diğerleri, 2004). Her mobil sunucu sabit bir IP adresi kullanır. Bu IP adresindeki tüm iletişim ilk olarak mobil sunucunun adresine iletişim evdeki aracıya (home agent) yönlendirilir. Evdeki aracı, mobil sunucunun IP adresini içeren ağ adresine karşılık gelen yerel ağ içinde yer almaktadır. IPy6 örneğinde olduğu gibi, bu bir ağ katmanı bileşeni olarak gerçekleştirilir. Her ne zaman mobil sunucu başka bir yığa taşınsa, iletişim için kullanabileceği geçici bir adres ister. Bu adres evdeki araçıda kayıtlıdır.

2.1.5.3. Alan adı adresleme yöntemi (Structured naming)

Hiyerarşik adresleme yöntemi makineler için iyidir, ancak genellikle insanların kullanması için çok uygun değildir. Alternatif olarak, adresleme sistemleri genellikle basit, insan tarafından okunabilir yapılandırılmış adresleri destekler. Sadece dosya

adlandırma değil, ama aynı zamanda internet üzerinde sunucu adlandırma da bu yaklaşımı takip eder. Alan adı adresleme yönteminin çeşitleri aşağıda incelenmiştir.

Ad alanları (Name spaces)

Adlar genellikle ad alanı olarak isimlendirilir ve düzenlenir. Yapılandırılmış adlar için ad alanları etiketlenmiş, iki tip düğümlü (yaprak ve dizin) bir grafik olarak yönlendirilmiş şekilde temsil edilirler. Bir yaprak düğümü (leaf node), adlandırılmış bir varlığı temsil eder ve dışarı giden kenarlar olmayan bir özelliği vardır. Bir yaprak düğümü, temsil ettiği varlıkla ilgili bilgileri depolar. Örneğin, bir istemcinin erişebilmesi için adresini depolar. Alternatif olarak, dosya sistemi durumunda olduğu gibi, o varlığın durumunu depolayabilir.

Bir yaprak düğümünün aksine, bir dizin düğümünün (directory node) her biri bir isimle etiketlenen birçok dışa giden kenarı vardır. Adlandırma grafiğindeki her bir düğüm dağınık bir sistemde başka bir varlık olarak kabul edilir. Bir dizin düğümü, dışa giden kenarın bir çift olarak temsil edildiği (kenar etiketi, düğüm tanımlayıcı) bir tabloyu depolar.

Ad çözümlemesi (Name resolution)

Ad alanları varlıklar ve adları hakkında bilgi depolamak ve almak için uygun bir mekanizma sunar. Daha genel olarak, bir yol adı verildiğinde, o isim tarafından temsil edilen düğüm içinde depolanmış herhangi bir bilgiyi aramak mümkün olmalıdır. Bu şekilde bir ismi arama işlemine ad çözümlemesi denir.

İsim alanlarının uygulanması (The implementation of a name space)

Ad alanı, adlandırma hizmetinin kalbini oluşturur; bu şekilde kullanıcıların ve süreçlerin ad ekleme, kaldırma ve arama yapmalarına izin verir. Bir adlandırma hizmeti ad sunucuları tarafından uygulanır. Eğer dağıtılmış bir sistem yerel bölge ağı ile sınırlandırılmışsa, sadece tek bir ad sunucusu yoluyla bir adlandırma servisini

uygulamak mümkündür. Bununla birlikte, birçok varlığa sahip büyük ölçekli dağıtılmış sistemlerde, birden çok ad sunucusu üzerinden bir ad alanının uygulanmasını dağıtmak gereklidir.

2.1.5.4. Hiyerarşik ve merkezi olmayan adresleme yöntemi

Hiyerarşik ve alan adı adresleme yönetimleri genellikle varlıklara benzersiz ve konumdan bağımsız bir şekilde atıfta bulunmalı yolu sağlarlar. Ayrıca alan adları, genellikle kendilerine kolayca erişilsin diye varlıkları insani ve dostça isimlendirmek için tasarlanmıştır. Çoğu durumda, ismin sadece bir tek varlığı kastettiği düşünülür. Ancak, yer bağımsızlığı ve insana kullanması için uygun olması sadece varlıkları adlandırma kriteri değildir. Özellikle, daha fazla bilgi sunulduğu sürece, mevcut varlıkları etkin bir şekilde aramak için daha önemli hale gelir. Bu yaklaşım, kullanıcının yalnızca aradığı şeyin açıklamasını sağlayabilmesini gerektirir.

Kısaca dağınık sistemlerde bir varlığın özellik, değer vb. özneliği bakımından tanımlanması öz nitelikli adresleme yöntemi olarak adlandırılmaktadır. Bu yaklaşımda, bir varlığı ilişkilendirilmiş bir seri özneliğe sahip olduğu varsayılır. Her özellik o varlık hakkında bir şey demektir. Belirli bir özelliğin hangi değerlere sahip olması gerektiğini belirterek, bir kullanıcı, ilgilendiği varlıkları kısıtlar. Kullanıcının tarif etmesine uyan bir veya daha fazla varlığın iade edilmesi adresleme sistemine bağlıdır.

Bu çalışmada, bölgelerin adlandırılması işlemi, hiyerarşik modelleme yönteminin alt başlığı olan ev temelli yaklaşımlar modeli ve alan adı adresleme modeli yöntemleri beraber kullanılmıştır. Sistemin fiziki olarak yapılandırılması bir switch aracılığı ile yapılmıştır. Dijital sayım cihazlarına erişim sağlanıp, cihazların IP adresleri statik hale getirilmiş ve her cihaza sıralı biçimde el ile IP numarası verilmiştir. IP numarasına sahip olan cihazlara ayrıca isimlendirme yapılmış olup, IP numaraları ile isimler beraber işlenmiştir. Dolayısıyla IP numarası veya isim ile mesaj gönderme yönteminin ikisi de çalışmaktadır. Fakat cihazlara isim ile mesaj gönderme işlemi daha kolay ve hızlı yapılır.

2.1.6. Senkronizasyon problemine çözüm metodu

İletişim ve işlemler çok önemli olmasına rağmen, her şey demek değildir. İşlemlerin birbirleriyle nasıl iş birliği yaptığı ve senkronize olduğu da oldukça önemlidir. Örneğin, birden çok işlemin bir yazıcı gibi paylaşılan kaynaklara aynı anda erişmemesi önemlidir. Ancak onun yerine birbirlerine geçici özel erişim sağlama konusunda iş birliği yaparlar. Başlıca senkronizasyon problemini çözme metotları aşağıda incelenmiştir.

2.1.6.1. Sistem saat senkronizasyon metodu

Merkezi bir sistemde zaman belirsizdir. Bir işlem zamanı bilmek istediğinde, bir sistem çağrısı yapar ve çekirdek bunu söyler. Eğer A işlemi zamanı sorarsa ve çok az bir süre sonra işlem B de zaman sorarsa, B'nin alacağı değer A'nın sahip olduğu değerden eşit veya daha yüksek olacaktır; kesinlikle daha düşük olmayacaktır. Dağınık bir sistemde, zamanında anlaşma sağlanması önemlidir. Örnek olarak, bir an için, küresel zamanın eksik olmasının UNIX programı üzerindeki etkilerini düşünün. Normalde, UNIX'te, büyük programlar birden çok kaynak dosyaya bölünür, böylece tek bir dosyaya yapılan bir değişiklik, tüm dosyaların değil, yalnızca bir dosyanın yeniden derlenmesini gerektirir. Bir program 100 dosyadan oluşuyorsa, bir dosya değişti diye tüm her şeyi yeniden derlemeye gerek yoktur. Bu durum da programcıların çalışabileceği hızı büyük ölçüde artırır.

Fiziksel saatler (Physical clocks)

Neredeyse tüm bilgisayarlarda zamanın kaydını tutmak için bir devre bulunmaktadır. Bu tür cihazların yerini tutacak şekilde “saat” sözcüğü çok geniş olarak kullanılmasına rağmen, aslında genel anlamda saat değildirler. Zamanlayıcı belki daha iyi bir kelimedir. Bir bilgisayar zamanlayıcısı genellikle hassas bir şekilde işlenmiş kuvars kristalidir.

Sistem başlatıldığında, genellikle kullanıcının tarih ve saati girmesini ister. Daha sonra bilinen bazı başlangıç tarihinden sonra işaret sayısına dönüştürülür ve bellekte saklanır. Çoğu bilgisayarın özel bir batarya-yedekli CMOS RAM'i vardır. Böylece,

sonraki ön yüklemelere tarih ve saat girilmesine gerek kalmaz. Her saat işaretinde, kesinti servis prosedürü bellekte saklanan süreye bir ekler. Bu şekilde, (yazılım) saat güncel tutulur.

Küresel konumlandırma sistemi (Global positioning system)

Gerçek saat senkronizasyon problemlerine doğru bir adım olarak, önce ilgili bir problemi ele alıyoruz, yani dünyanın herhangi bir yerinde coğrafi konum belirliyoruz. Bu konumlandırma problemi, oldukça spesifik dağınık bir sistem olan küresel konumlandırma sistemi GPS ile çözülmüştür. GPS, 1978 yılında başlatılan bir uydu tabanlı dağınık bir sistemdir. Esas olarak askeri uygulamalarda kullanılmasına rağmen, son yıllarda özellikle trafik navigasyonu gibi birçok sivil uygulamada kullanılmaya başlamıştır. Ancak daha birçok uygulama alanı vardır. Örneğin, artık GPS telefonları, arayanların birbirlerinin pozisyonlarını izlemelerine izin vermektedir. Bu özellik, kaybolduğunuzda ya da sıkıştığınızda çok kullanışlı olabileceğini göstermektedir. Bu prensip, evcil hayvanlar, çocuklar, arabalar, tekneler vb. birçok şeyin takibinde kolayca uygulanabilir (Zogg, 2002).

Bu çalışmada sistem saat senkronizasyon metodunun alt başlığı olan fiziksel saat yöntemi kullanılmıştır. Bu yöntem sayesinde senkronizasyon daha ucuz olarak sağlanmıştır. Küresel konumlandırma sistemi yöntemi, yapılan işlemlerin saatini daha kesin sonuç olarak vereceği aşikârdır. Fakat bu çalışmada tartışılan konu saatin tam doğruluğu değildir. Daha çok yapılan işlemlerin saatinin yaklaşık olarak birbirleri ile tutarlı olması sonuca varmak için yeterli olacaktır.

2.1.6.2. Lojik saat sistemi metodu

Saat eşleştirmesinin doğal olarak gerçek ile bağlantılı olduğunu varsayılmaktadır. Bununla birlikte, her bir düğümün o andaki zaman üzerinde yeterli olabilir. Örneğin, make (orjinalini değiştirmedim) çalıştırmak için, iki düğümün input.o'nun input.c'nin yeni bir sürümü tarafından güncelleştirildiğinin kabul etmeleri yeterlidir. Bu durumda, birbirlerinin olaylarını takip etmek (input.c'nin yeni bir versiyonun

üretilmesi gibi) önemli olan şeydir. Bu algoritmalar için, saatleri mantıksal saatler olarak adlandırmak gelenekseldir.

Klasik makalesinde, Lamport (1978) saat senkronizasyonunun mümkün olmasına rağmen, mutlak olması gerektiğini ileri sürmüştür. Eğer iki işlem etkileşmezse, saatlerinin mutlaka senkronize edilmesi gerekmez. Çünkü senkronizasyon eksikliği gözlenebilir değildir ve bu nedenle sorunlara neden olmaz. Ayrıca, Lamport'a göre önemli olan, tüm işlemlerin mutlak saatin kaç olduğu konusunda aynı fikirde olmaları değil, olayların gerçekleştiği sıra hakkında aynı fikirde olmaları gerektiğidir. Make örneği, input.c'nin input.o'dan daha eski veya daha yeni olup olmadığı önemlidir. Mutlak yaratma süreleri önemli değildir.

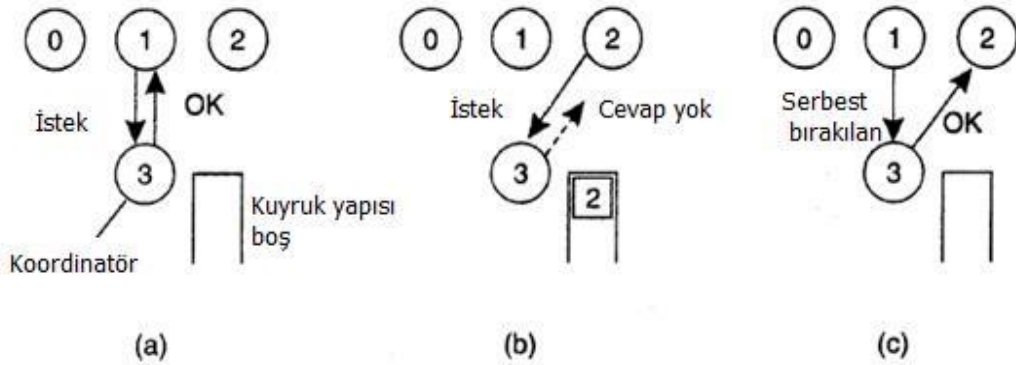
2.1.6.3. Dışlama metodu

Dağınık sistemlerin temelini çoklu işlemler arasındaki eşzamanlılık ve iş birliği oluşturur. Çoğu durumda, bu aynı zamanda işlemlerin aynı kaynaklara aynı anda erişmesi gerekeceği anlamına gelir. Bu tür eşzamanlı erişimlerin kaynağı bozması ya da tutarsız hale getirmesini önlemek için, işlemler tarafından karşılıklı dışlama sağlamak için çözümlere ihtiyaç duyulmaktadır.

Dağınık karşılıklı dışlama algoritmaları iki farklı kategoriler halinde sınıflandırılabilir. Simge-tabanlı (token-based) çözümlerde karşılıklı dışlama, bir simge olarak bilinen işlemler arasında özel bir mesajın paylaşılması ile başarılıdır. Uygun sadece bir adet simge vardır ve ona sahip olan paylaşılan kaynağa erişebilir. Bittiğinde, simge bir sonraki işleme geçirilir. Eğer simgeye sahip işlem kaynağa erişmeyle ilgilenmez ise, sadece onu aktarır. Alternatif olarak, birçok dağınık dışlama algoritması izin temelli yaklaşımını takip eder. Bu durumda, kaynaklara erişmek isteyen bir işlem, ilk olarak diğer işlemlerin iznini alması gerekmektedir. Bu şekilde bir izin vermenin çok farklı yolları vardır. Bunlardan öne çıkanlar aşağıda incelenmiştir.

Merkezi algoritma (A centralized algorithm)

Dağınık bir sistemde dışlamanın başarılmasının en basit yolu tek işlemcili bir sistemde nasıl yapıldığını benzetim yoluyla elde etmektir. Bir işlem, koordinatör olarak seçilir. Bir süreç, paylaşılan bir kaynağa erişmek istediğinde, koordinatöre hangi kaynağa erişmek istediğini ve izin istediğini belirten bir istek mesajı gönderir. O anda bu kaynağa erişen başka bir işlem yoksa koordinatör izin verdiğini belirten geri bir cevap gönderir. Cevap geldiğinde, istek süreci devam edebilir. Bu süreç aşağıdaki Şekil 2.8.'de gösterilmiştir.



Şekil 2.8. Merkezi algoritma (a) Koordinatöre paylaşılan bir kaynağa erişim izni için sorar. İzin verilir.(b) Aynı kaynağa erişmek için izin ister. Koordinatör cevap vermemektedir.(c) Süreç 1 kayıt kaynağı olduğunda, koordinatöre söyler ve sonra süreç 2'ye cevap verir. (Tanenbaum ve Van Steen, 2006)

Merkezi olmayan algoritma (A decentralized algorithm)

Tek bir koordinatöre sahip olmak genellikle zayıf bir yaklaşımdır. Bu nedenle tamamen merkezi olmayan çözüme bakmak gerekebilir. Lin ve ark. (2004) DHT tabanlı bir sistem kullanılarak çalıştırılabilen bir önerme (vote) algoritmasını kullanmayı önermektedir. Özünde, çözümleri merkezi koordinatörü şu şekilde genişletir. Her kaynağın n kez kopyalandığı varsayılır. Her kopya, eşzamanlı işlemlerin erişimi kontrol etmek için kendi koordinatörüne sahiptir.

Ancak, bir süreç kaynağa erişmek istediğinde, sadece $n/2$ koordinatörlerinden çoğunluk oyu alması gerekecektir. Merkezi alitmadan farklı olarak, bir koordinatörün bir kaynağa erişim izni vermediği zaman istekte bulunana söyleyeceği düşünülmektedir.

Bu algoritma esas olarak tek bir koordinatörün başarısızlıklarına karşı orijinal merkezi çözümü daha az savunmasız hale getirmektedir. Buradaki varsayım şudur: Bir koordinatör çöktüğünde, çabucak iyileşir fakat çökmeden önce verdiği herhangi bir oyu unutmuş olacaktır. Bunu görmenin başka bir yolu, bir koordinatörün gelişigüzel olarak kendini sıfırlamasıdır. Bu durumda aldığımız risk, bir sıfırlamanın koordinatöre daha önce kaynağa erişmek için bazı işlemlere izin verdiğini unutturacak olmasıdır. Sonuç olarak, kurtarıldıktan sonraki süreçte bu izni yanlış bir şekilde başka bir işleme verebilir.

Dağıntık algoritma (Distributed algorithm)

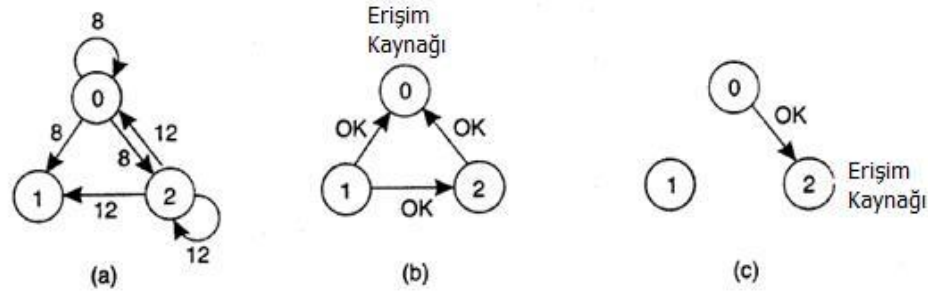
Dağıntık algoritma Ricart ve Agrawala (1981) tarafından geliştirilmiştir. Ricart ve Agrawala'nın algoritması, sistemdeki tüm olayların hepsinin toplam bir düzeni olmasını gerektirir. Yani, mesajlar gibi herhangi bir olay çifti için hangisinin daha önce gerçekleştiği kesin belirli olmalıdır. Algoritma aşağıdaki gibi çalışır. Bir işlem paylaşılan bir kaynağa erişmek istediğinde, kaynağın adını, işlem numarasını ve geçerli (mantıksal) zamanı içeren bir mesaj oluşturur. Daha sonra mesajı, kavramsal olarak kendisi de dâhil olmak üzere diğer tüm süreçlere gönderir. Mesajların gönderilmesinin güvenilir olduğu varsayılmaktadır; yani, hiçbir mesaj kaybolmaz.

Bir işlem başka bir işlemde bir istek mesajı aldığı anda, mesajda yer alan kaynak isme göre kendi durumunu belirler. Üç farklı durum açıkça ayırt edilmelidir:

1. Alıcı kaynağa erişmiyorsa ve erişmek istemiyorsa göndericiye bir OK mesajı gönderir.
2. Alıcının zaten kaynağa erişimi varsa, cevap vermez. Bunun yerine, isteği sıraya alır.
3. Alıcı, kaynağa da erişmek istiyorsa ancak henüz erişmediyse gelen mesajın zaman bilgisini herkese gönderdiği mesajın içinde karşılaştırır. En yavaş olan kazanır.

Gelen mesajın daha düşük bir zaman bilgisi varsa, alıcı OK mesajını geri gönderir. Kendi mesajı daha düşük zaman bilgisine sahipse, alıcı gelen isteği sıraya alır ve hiçbir şey göndermez. Bir işlem, izin isteyen istekler gönderdikten sonra, geri oturur

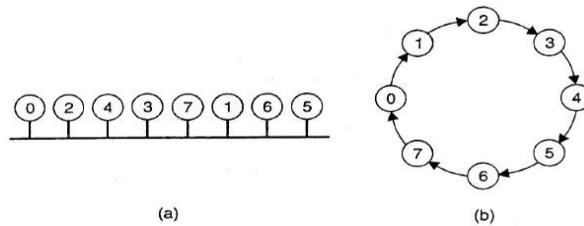
ve herkes izin verene kadar bekler. Tüm izinler verili verilmez, devam edebilir. Tamamlandığında, tamamlanmış mesajlar üzerinde tüm süreçlere OK mesajları gönderir ve sıradan hepsini siler. İzin isteyen istekler gönderildikten sonra, bir işle beklemeye geçer ve herkes izin verene kadar bekler. Tüm izinler girer girmez, devam edebilir. Tamamlandığında, OK mesajlarını sırada bekleyen tüm işlemlere gönderir ve hepsini sıradan siler. Şekil 2.9.'da örnek gösterim verilmiştir.



Simgeli halka algoritması

Dağınık bir sistemde dışlamayı kararlılıkla elde etmenin tamamen farklı bir yaklaşım aşağıdaki şekilde gösterilmiştir. Ethernet örneğinde olduğu gibi, burada işlemlerin kendine özgü sırası olmayacak şekilde bir veri yolu ağı vardır. Şekil 2.10.(b)'de gösterildiği gibi yazılımda, bir mantıksal döngü, her işlemin döngü içinde bir konuma atandığı şekilde oluşturulur.

Döngü pozisyonları, ağ adreslerinin veya diğer bazı yolların sayısal düzeninde tahsis edilebilir. Siparişin ne olduğu önemli değil. Asıl önemli olan, her işlemin kendinden sonra sırada kim olduğunu bilmesidir.



Şekil 2.10. Simgeli halka algoritması (a) Bir ağdaki sırasız bir süreçler grubu. (b) Yazılı bir mantıksal halka.(Tanenbaum ve Van Steen, 2006)

Bu çalışmada dışlama metodu iptal edilmiştir. Dışlama metodu, sayıma giren personelin bilgilerinin sisteme yazılmaması problemini ortaya çıkarmaktadır. Merkezi yapılı dışlama sistemi ters çevrilmiştir. Bu sayede merkezi yapılı kabul sistemi oluşturulmuştur. Gelen verileri sunucu saatine göre sisteme yazılmaktadır. Sahada bulunan dijital sayım cihazlarının saatleri göz ardı edilmektedir. Fakat her ay bu cihazların ve sunucu bilgisayarının saat ayarlarının düzenli olarak eşleştirilmesi, oluşacak acil durumlarda sistemin hazır durumda olması ve daha verimli çalışmasını sağlayacaktır.

2.1.7.Toleranslar

Dağınık sistemleri tek-makine sistemlerinde ayıran en önemli özellik kısmi hata durumudur. Dağınık sistemdeki bir bileşen hata yaptığında kısmi hata durumu meydana gelebilir. Bu hata diğer bileşenlerin düzgün çalışmasını etkileyebilir. Aksine, dağınık olmayan sistemlerde bir hata genellikle tüm bileşenleri etkileyebilir ve kolayca çökmesine neden olabilir. Dağıtılmış sistem tasarımında önemli bir amaç, sistemi genel performansını ciddi bir şekilde etkilemeden sistemi kısmi hatalardan otomatik olarak kurtulabilecek şekilde oluşturmaktır. Özellikle, bir hata meydana geldiğinde, dağınık sistem onarım sırasında kabul edilebilir bir şekilde çalışmaya devam etmelidir. Yani, hataları tolere etmeli ve onların varlığında bile olabildiğince çalışmaya devam etmelidir.

Dağınık sistemlerde hata toleransının rolünü anlamak için ilk olarak dağınık bir sistemde hataların düzeltilmesinin ne anlama geldiğini daha yakından incelemek gerekmektedir. Hatalara toleranslı olmak, güvenilir sistemler olarak adlandırılan sistemlere yakından ilişkilidir. Güvenilirlik, dağınık istemlerde aşağıdakileri içeren bir dizi yararlı gereksinimi kapsayan bir terimdir (Kopetz ve Verissimo, 1993):

1. Kullanılabilirlik (Availability)
2. Güvenilirlik (Reliability)
3. Güvenlik (Safety)
4. Sürdürülebilirlik (Maintainability)

Kullanılabilirlik:

Kullanılabilirlik, bir sistemin hemen kullanılmaya hazır olması anlamına gelir. Genel olarak, sistemin herhangi belirli bir anda doğru bir şekilde çalışma durumunu ve yürütmüş olduğu işlevlerini yerine getirebilme durumunu ifade eder. Diğer bir deyişle, yüksek oranda kullanılabilir bir sistem, kendisine verilen herhangi bir zaman diliminde çalışabilir olmasına bağlıdır.

Bu çalışmada kullanılabilirlik durumu için öngörülen koşul şu şekildedir. Dijital sayım cihazları acil durum toplanma alanlarında bir kabin içerisinde, güneşten korunacak şekilde muhafaza edilmelidir. Ayrıca bu sistemlere enerji verilmesi bu kabin içerisindeki şebeke veya güç kaynağı aracılığı ile yapılabilir.

Güvenilirlik:

Güvenilirlik, bir sistemin hata yapmadan kesintisiz olarak çalışabilmesini ifade eder. Kullanılabilirliğin aksine, güvenilirlik belirli bir an yerine bir zaman aralığı cinsinden tanımlanır. Son derece güvenilir bir sistem, büyük olasılıkla nispeten uzun bir süre boyunca kesintisiz çalışmaya devam eden sistemdir. Bu durum kullanılabilirlik ile karşılaştırıldığında ince ama önemli bir farktır. Eğer bir sistem saatte bir milisaniye geriye giderse, % 99.9999 'den fazla kullanılabilirliği vardır; ancak yine de güvenilmez. Benzer şekilde, asla çökmeyen ancak her Ağustos ayında iki hafta boyunca kapanana bir sistem yüksek güvenilirliğe sahiptir ama sadece yüzde 96 kullanılabilirliği vardır. İkisi aynı değildir.

Bu çalışmada güvenilirlik yapısı sürekli çalışmaya bağlı değildir. Sisteme ihtiyaç duyulduğu zaman çalışması esastır. Bu sebeple sistemin ayda bir çalıştırılması ve deneme işlemlerinin gerçekleştirilmesi şarttır. Elektronik cihazlar ve bu cihazları bağlayan kablolar, doğa koşullarında arıza yapabilir. Oluşabilecek hatalar kontroller esnasında ortaya çıkarılacaktır.

Güvenlik:

Güvenlik, bir sistemin doğru olarak çalışması geçici olarak kesintiye uğrasa bile çok büyük olumsuzluğun meydana gelmemesi durumunu ifade eder. Örneğin, nükleer santralleri kontrol etmek veya uzaya insan göndermek için kullanılanlar sistemler gibi birçok işlem kontrol sistemlerinde yüksek derecede güvenlik sağlanması gereklidir. Eğer bu tür kontrol sistemleri geçici olarak sadece çok kısa bir süre için hata yapsa, sonuçları felaket olabilir. Geçmişten gelen birçok örnek (ve muhtemelen daha birçokları) güvenli sistemleri kurmanın ne kadar çok zor göstermektedir.

Bu çalışmada, güvenlik konusu sistemin ayda bir kere test edilmesiyle kontrol altında tutulabilir. Fakat sistem sızmalara karşı engelleyici unsurlara sahip değildir. Sistemin yerel ağa sahip olması sebebiyle, dışarıdan sızmalara karşı korunaklıdır. Fakat sunucu bilgisayarından isteyen istediği bilgiyi kopyalama imkânına sahiptir. Gelecekte yapılacak çalışmalarda, bu eksiklikler düzeltilmelidir.

Süreklilik:

Süreklilik, hata yapan bir istemin ne kadar kolay tamir edilebileceğini ifade eder. Özellikle hataların otomatik olarak tespit edildiği ve onarıldığı durumlarda, yüksek oranda sürekliliği olan bir sistem aynı zamanda yüksek oranda kullanılabilirlik gösterebilir. Ancak, yapacağımız gibi hatalardan otomatik olarak kurtarmak daha kolaydır.

Çoğu zaman, özellikle dürüstlük gibi konular söz konusu olduğunda güvenilir sistemlerde yüksek derecede güvenlik sağlamak çok önemlidir.

Bu çalışmada süreklilik konusu için, sistem açık grup iletişim sistemi olarak tasarlanmıştır. Sahada çalışan bir dijital sayım cihazı hata yaparsa veya çalışmazsa, sayıma katılan personel sahadaki en yakın diğer sayım cihazına kendini saydırabilir. Dolayısıyla sistemin sürekliliği sağlanmış olacaktır. Fakat asıl olan sahadaki tüm dijital sayım cihazlarının çalışır vaziyette tutulmasıdır.

Hata Modelleri (Failure Models)

Bir hatanın ne kadar ciddi olduğuna dair daha iyi anlamak için, birçok sınıflandırma geliştirilmiştir. Cristian (1991) ve Hadzilacos ve Toueg (1993) tarafından yapılan sınıflandırma şu şekildedir: çökme hatası, ihmal hatası, zamanlama hatası, yanıtlatma hatası. Hata modelleri Tablo 2.1.'de gösterilmiştir.

Tablo 2.1. Hata tipleri ve tanımları(Tanenbaum ve Van Steen, 2006)

Hata Tipi	Tanımı
Çökme Hatası	Sunucu durur, fakat durana kadar düzgün çalışır
İhmal Hatası	Sunucu gelen taleplere cevap vermede başarısız
Alıcı İhmali	Sunucu gelen mesajı almada başarısız
Gönderici İhmali	Sunucu mesaj göndermede başarısız
Zamanlama Hatası	Sunucu cevap verme zamanı istenen aralıkta değil
Yanıtlatma Hatası	Sunucu cevabı hatalı
Değer Hatası	Cevap değeri yanlış
Durum Geçiş Hatası	Sunucu doğru kontrol akımından saptı
Gelişigüzel Hata	Sunucu geliş güzel bir zamanda geliş güzel cevaplar verir

Çökme hatası:

Bir çökme hatası, bir sunucu zamanından önce durduğunda, ancak duruncaya kadar doğru çalışırken oluşur. Kaza arızalarının önemli bir yönü, sunucu durduğu zaman artık ondan hiçbir şey duyulmamasıdır. Bir işletim sisteminden bir çökme hatasının tipik bir örneği, aksama noktasına geldiği anda sadece bir çözüm olmasıdır: yeniden başlatma. Birçok kişisel bilgisayar sistemi o kadar çökme hatası yaşıyor ki insanlar artık normal karşılar hale gelmiştir. Sonuç olarak, sıfırlama düğmesine basılması gerekmektedir.

Bu çalışmada konu olan sunucu sistemin ve yazılımın bir kopyası sürekli olarak yedekte bekletilmelidir. Sunucunun çökme hatasına karşılık, yedek sunucu devreye alınabilir. Felaket senaryosu sunucusu olarak da bilinen yedek sunucu mümkünse

farklı bir binada konumlandırılmalıdır. İhtiyaç halinde acil durum komuta merkezine getirilmeli ve devreye alınmalıdır.

İhmal hatası:

Bir ihmal hatası, bir sunucu bir isteğe yanıt vermediğinde oluşur. Birçok şey yanlış gitmiş olabilir. Bir alım ihmal hatası durumunda, muhtemelen sunucu ilk isteği hiç almamıştır. Bir istemci ve sunucu arasındaki bağlantı doğru şekilde kurulmuş olmasında rağmen gelen istekleri dinleyen bir iş parçacığı yoktur. Ayrıca, sunucu kendisine gönderilen herhangi bir mesajın farkında olmadığı için bir ihmal hatası genellikle sunucunun mevcut durumunu etkilemez. Benzer şekilde, gönderim hatası ise sunucu işini bitirdiğinde fakat bir şekilde yanıt vermede başarısız olduğunda meydana gelir.

Zamanlama hatası:

Zamanlama hataları, yanıt, belirtilen bir gerçek zaman aralığının dışında yer aldığı zamanlarda meydana gelir. Verilerin çok kısa sürede sağlanması, tüm gelen verileri tutmak için yeterli arabellek alanı yoksa alıcıda kolayca bir soruna neden olabilir. Ancak, daha genel olanı, bir sunucunun çok geç cevap verdiği durumda bir performans hatasının gerçekleştiği söylenir.

Bu çalışmada zamanlama hatası olarak bahsedebileceğimiz konu şudur; Sistem tarih ve saat bazlı çalışmaktadır. En büyük problem gece saat 00:00 civarında yaşanacak bir afet veya tatbikattır. Sistem günlük olarak çalışmaktadır. Sistemin saati 00:00 itibari ile yani güne başlandığı ve verilerin bu duruma göre işleneceği bilgisini kendi içerisinde tutmaktadır. Bu iyileştirilmesi gereken ve yaşanabilecek bir hatadır.

Yanıtlama hatası:

Yanıtlama hatası ise, sunucunun verdiği yanıtın yanlış olma durumudur. İki çeşit müdahale hatası meydana gelebilir. Değer hatası durumunda, bir sunucu, bir istek

için yanlış cevap verir. Bir arama motorunun arama terimleriyle ilgili olmayacak bir şekilde web sayfalarını sistematik olarak döndürmesi değer hatasına örnektir. Durum geçiş hatası ise, sunucunun bir isteğe beklenmedik şekilde tepki verdiğinde gerçekleşir. Örneğin, bir sunucu algılayamadığı bir mesaj aldığında, bu tür mesajları işlemek için hiçbir önlem alınmadığında durum geçiş hatası gerçekleşir. Özellikle, hatalı bir sunucu asla başlamaması gereken hatalı işlemleri yanlış bir şekilde alabilir.

Gelişigüzel hatalar:

Gelişigüzel hatalar ise bir sunucunun gelişigüzel zamanlarda gelişigüzel yanıtlar vermesi demektir. Gelişigüzel hatalar en ciddi hata sınıfıdır çünkü bu hatalar genellikle çökme hataları ile yakından ilişkilidirler. Gerçekte, gelişigüzel hatalar meydana geldiğinde, istemciler en kötüye hazır olmalıdır.

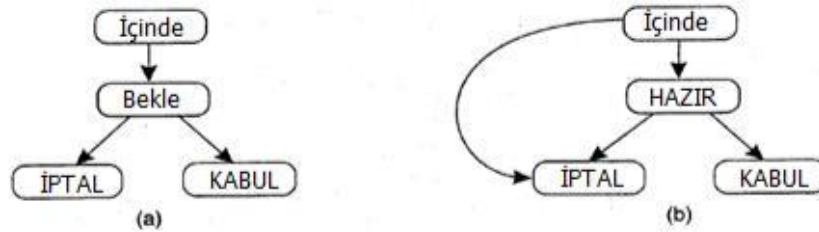
2.1.7.1. İki aşamalı sistem kurulumu

İki aşamalı sistem kurulumu (2PC), dağıtılmış bir sistemdeki tüm düğümlerin bir işlem gerçekleştirmeyi kabul etmesini sağlayan dağınık bir algoritmadır (Gray, 1978). Herhangi bir arıza meydana gelmediğini varsayarsak, protokol iki aşamadan ve iki adımdan oluşur [ayrıca bkz Bernstein ver ark. (1987)]:

1. Koordinatör, tüm katılımcılara bir GÖNDER-AL mesajı gönderir.
2. Bir katılımcı bir GÖNDER-AL mesajı aldığında ya işlemin bir kısmını yerel olarak yerine getirmeye hazır olduğunu belirten bir GÖNDERİ-KABUL iletisini ya da başka bir GÖNDERİ-RED iletisini koordinatöre geri gönderir.
3. Koordinatör, katılımcıların tüm önermelerini toplar. Eğer tüm katılımcılar işlemi yürütmek için önermede buldu ise koordinatör de aynısını yapar. Bu durumda, tüm katılımcılara bir GÖNDERİ-KABUL iletisi gönderir. Ancak, bir katılımcı işlemi iptal etmek için oy vermişse, koordinatör ayrıca işlemi iptal etme kararı alır ve bir GÖNDERİ-RED mesajı gönderir.

4. Bir taahhüt için oy veren her katılımcı koordinatör tarafından verilecek nihai tepkiyi bekler. Bir katılımcı bir EVRENSEL-KABUL mesajı alırsa, yerel olarak işlemleri taahhüt eder. Aksi halde, EVRENSEL-RED mesajı alındığında işlem yerel olarak iptal edilir.

İlk aşama oylama aşamasıdır ve 1. ile 2. adımlardan oluşur. İkinci aşama ise karar aşamasıdır ve 3. ile 4. adımlardan oluşur. Bu dört adım Şekil 2.11. sonlu durum diyagramları olarak gösterilmiştir.



Şekil 2.11. İki aşamalı sistem (a) 2 aşamalı sistemdeki koordinatör için sonlu durum makinesi. (b) Katılımcı için sonlu durum makinesi. (Tanenbaum ve Van Steen, 2006)

Bu çalışmada iki aşamalı sistem kurulumu şeklindedir; Birinci aşamada personeller kendi parmak izlerini veya personel kartlarını sahadaki dijital sayım cihazlarına okuturlar. İkinci aşamada ise C# ile hazırlanan program ile karar vererek listeleri oluşturur. Bu listeler içerisindeki en önemli liste eksik personel listesidir ki bu çalışmanın asıl amacı eksiklerin belirlenmesi ve acil durum müdahale ekiplerine bu listenin verilmesidir.

2.1.8. Kurtarma

Hata toleransının temelini bir sorundan (error) kurtulma oluşturur. Sorun, bir sistemin hata yapmasına neden yol açabilecek kısımdır. Sorundan kurtulmanın ana fikri, sorun yaratan bir durumun sorunsuz bir durumla değiştirilmesidir. Esasen geri dönüş kurtarması ve ileri kurtarması olmak üzere iki tür sorun kurtarma şekli vardır.

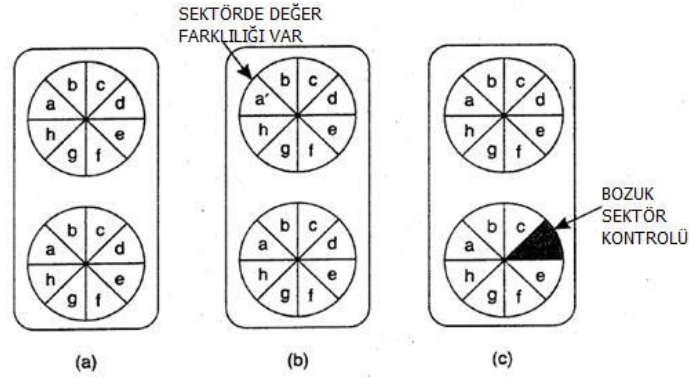
Geri dönüş kurtarmasında ana konu, sistemi mevcut sorunlu durumundan daha önceki sorunsuz bir duruma geri getirmektir. Bunu yapmak için, sistemin durumunu

zaman zaman kaydetmek ve işler ters gittiğinde böyle bir durumu geri yüklemek gerekli olacaktır. Sistemin mevcut durumu her seferinde (bir kısmı) kaydedildiğinde, bir kontrol noktası yapılır.

Bir başka hata kurtarma şekli ileri kurtarmadır. Sistem, sorunlu bir duruma girdiğinde, sistemi bir önceki haline geri dönmek yerine, çalışmaya devam edebileceği doğru yeni bir duruma getirmek için girişimde bulunulur. İleri hata giderme mekanizmalarıyla ilgili temel problem, hangi hataların meydana gelebileceğinin önceden bilinmesi gerektiğidir. Sadece bu durumda bu sorunları düzeltmek ve yeni bir duruma geçmek mümkündür.

İstikrarlı depolama (Stable storage)

Bir önceki duruma geri dönebilmek için, kurtarma sağlamak için gerekli bilgilerin güvenli bir şekilde saklanması gereklidir. Bu bağlamda güvenli demek kurtarma bilgilerinin bilgi işlem çökmeleri ve site hatalarından etkilenmemesi anlamına gelmektedir. İstikrarlı depolama, dağınık sistemlerde kurtarma söz konusu olduğunda önemli bir rol oynar. Depolama üç kategoride olur. İlk olarak güç kesildiğinde veya makine çökerken silinen sıradan bir RAM belleği vardır. İkinci olarak CPU arızalarından kurtulabilen ancak disk kafasındaki çökmelerde kaybolabilecek disk depolama vardır. Son olarak ise, sel ve deprem gibi büyük felaketler dışında meydana gelebilecek olumsuzlardan etkilenmeyecek şekilde tasarlanmış istikrarlı depolama vardır. Şekil 2.12.'de de gösterildiği üzere bir çift sıradan disk ile uygulanabilir. Sürücü 2'deki her blok, sürücü 1'deki ilgili bloğun tam bir kopyasıdır. Bir blok güncellendiğinde, önce sürücü 1'deki blok güncellenir ve doğrulanır. Daha sonra sürücü 2'deki aynı blok tamamlanır.



Şekil 2.12. İstikrarlı depolama (a) Stabil depolama. (b) İşleyiş sonrası arıza yenilendi (c) Bozuk sektör.

(Tanenbaum ve Van Steen, 2006)

Bu çalışmada geriye dönük kurtarma işlemleri istikrarlı depolama operasyonunun yapılması için, C# ile hazırlanan ara yüze sahip uygulamada, veri tabanlarının bir yedeği oluşturulmaktadır. Yedeklenen veri tabanları ve C# uygulaması sayesinde, geçmişe dönük olan operasyonların tümüne erişim sağlanabilir.

2.1.8.1. Mesajların yedeklenmesi

Mesajların yedeklenmesi işleminin altında yatan temel fikir, eğer mesajların iletimi tekrarlanabiliyorsa, küresel olarak tutarlı bir duruma erişilebilir. Ancak sabit depolamadan geri yükleme yapılamaz. Bunun yerine, bir kontrol noktası başlangıç noktası olarak alınır ve o zamandan beri gönderilen tüm mesajlar buna göre yeniden iletilir ve işlenir.

Bu yaklaşım, parçalı deterministik model olarak adlandırılan varsayım altında iyi çalışır. Böyle bir modelde, her bir işlemin yürütülmesi, etkinliklerin gerçekleştiği bir dizi aralık olarak yer aldığı varsayılır. Örneğin, bir etkinlik bir komutun yerine getirilmesi, bir mesajın gönderilmesi, vb. olabilir. Parçalı deterministik modeldeki her bir aralığın, bir mesajın alınması gibi deterministik olmayan bir olay ile başlayacağı varsayılmaktadır. Ancak, o andan itibaren, sürecin yürütülmesi tamamen deterministiktir. Bir aralık, deterministik olmayan bir olay gerçekleşmeden önceki son olay ile biter.

Global olarak tutarlı bir durumun geri yüklenmesi için bir işlem çökmesinden kurtulmak için ileti kayıtlarının gerekli olduğunu düşünürsek, ileti kayıtlarının zamanının tam olarak bilmek önemlidir. Alvisi ve Marzullo'nun (1998) yaklaşımında belirtildiği gibi, eğer yetim işlemler ile nasıl başa çıkılacağına odaklanırsak, mevcut olan birçok mesaj-kayıt şemasının kolaylıkla karakterize edilebileceği ortaya çıkmaktadır. Bir askıdaki işlem, başka bir işlemin çöküşünden kurtulan fakat ancak durumu iyileştikten sonra çöken işlem ile süreci ile durumu tutarsız olan bir işlemdir.

Bu çalışmada, gönderilen mesajların bir önemi olmadı için mesajlar yedekleme sistemine dâhil edilmemiştir.

2.1.8.2. Yedekleme bölgesinin seçimi

Teze konu olan sistemin yedekleri C# ile yazılan program ile alınmaktadır. Ayrıca bu yedeklerin güvenli bir alana taşınması gerekmektedir. Bilgisayar sistemlerinde olduğu gibi felaket senaryosu yedekleme operasyonları sürekli olarak ayrı bir bölgede veya binada tutulması gerekmektedir. Yedekleme bölgesinin seçiminde iki temel kural vardır. Birinci kural erişimi kolay bir yer olmalı, ikinci kural yedeklerin tutulduğu bina yeterince sağlam olmalıdır. Aksi takdirde yedeklere erişim imkânı mümkün olmayacaktır.

2.2. Bulanık Mantık Metodu İle Dijital Sayım Benzetimi

Bulanık mantık, değişkenlerin gerçek değerlerinin 0 ile 1 arasında herhangi bir gerçek sayı olduğu bir mantık şeklidir. Tamamen yanlış ve tamamen doğru arasında değişebilecek kısmi gerçeklik kavramını ele almak için kullanılır. Buna karşılık, Boole mantığında, değişkenlerin doğruluk değerleri sadece 0 veya 1 tamsayı değerleri olabilir (Novak ve ark. 1999).

Bulanık mantık terimi 1965 Lotfi Zadeh tarafından bulanık küme teorisi önerisi ile tanıtılmıştır (Zadeh, 1965). Ancak bulanık mantık düşüncesinin kökleri antik Yunan filozoflarında kadar gitmektedir. Çünkü Plato, doğru ve yanlış terimler arasındaki bir noktayı üçüncü bir alan olarak kabul etmiştir. Bu, matematiğin kesinliğini ortaya

koyan çağdaş Aristoteles'in aksine, bir ifadenin sadece doğru ya da yanlış olabileceğidir. Modern bulanıklık kavramına yapılan atıflar, Georg Wilhelm Friedrich Hegel'in "Gedoppelten Mitte" (ikileştirilmiş ortalama) kavramına dayanmaktadır. Bulanık mantık 1920'lerden beri, özellikle de Łukasiewicz ve Tarski tarafından sonsuz değer mantığı olarak incelenmektedir (Pelletier, 2000). Bulanık mantık, kontrol teorisinden yapay zekâyâ birçok alanda uygulanmaktadır. Başlarda sistemlerin açıklayıcı tasvirlerinin bulanıklaşmasını modellemek için kullanılmıştır. Ancak bugün kontrol mühendisliği gibi uygulamalı alanlarda önemli bir rol oynamaktadır.

Klasik mantık sadece doğru ya da yanlış olan sonuçlara izin verir. Bununla birlikte, bir grup insanın bir rengi tanımlamasını istediğinde bulabileceği gibi değişken cevaplara sahip önermeler de vardır. Bu gibi durumlarda, gerçek, örneklenen cevapların bir spektrumda eşleştirildiği eksik veya kısmi bilgiden kaynaklanan akıl yürütmenin bir sonucu olarak ortaya çıkar. Her iki derece de gerçek ve olasılık dereceleri 0 ile 1 arasında değişir ve bu nedenle ilk bakışta benzer görünebilir, fakat bulanık mantık gerçekliğin derecelerini matematiksel bir belirsizlik modeli olarak kullanır, olasılık ise matematiksel bir görmezden gelme modelidir.

Temel bir uygulama, sürekli değişkenin çeşitli alt aralıklarını karakterize edebilir. Örneğin, kilitlemeyi önleyici frenler için bir sıcaklık ölçümü, frenleri doğru şekilde kontrol etmek için gerekli olan belirli sıcaklık aralıklarını tanımlayan birkaç ayrı üyelik fonksiyonuna sahip olabilir. Her işlev, aynı sıcaklık değerini 0 ile 1 aralığında bir gerçek değere eşler. Bu gerçek değerler daha sonra frenlerin nasıl kontrol edileceğini belirlemek için kullanılabilir.

Bu tezde bulanık mantık kullanılmasının amacı, dijital sayım sisteminin orta veya büyük ölçekli bir tesiste, sahada kaç adet cihaz ile sistemin yönetilebileceği hakkında bilgiye ulaşmaktır. Sayıma giren kişilerin, sayım bölgelerine uzaklığı ve dijital sayım cihazların aralarındaki uzaklığa göre cihaz sayım adedi bulunması için bulanık mantık yöntemiyle MATLAB programı kullanılarak çözüm aranmıştır. Bu yöntemin kullanılmasındaki birinci kısıt sayım bölgelerindeki sayım işleminin beş dakika

içerisinde tamamlanma zorunluluğudur. İkinci kısıt ise, bir adet dijital sayım cihazına elli beş kişinin düşmesidir. Genelme yapılacak olursak, bir çalışan için basit fizik formülü olan $yol = hız \times zaman$ formülü kullanılmaktadır. Bu formül sayesinde evrensel yürüme hızı olan 1 saniyede 130 santimetre alınarak, çalışanın ne kadar sürede sayım alanında olabileceği ve sayım işlemine katılacağı süre hesaplanmaktadır. MATLAB programı sayesinde, bu işlemlerin matematiksel modellemeleri yapılmaktadır. MATLAB programının bulanık mantık çözümleme editörü kullanılarak MAMDANI modelleme sistemi ile sahada kaç adet cihaz kullanılması problemine çözüm aranmıştır. Çözümün bulunması için iki girişli ve bir çıkışlı sistem oluşturulmuştur. İki giriş ve bir için bulanık kümeler oluşturulmuş ve daha sonra üyelik fonksiyonları MATLAB bulanık mantık editörüne girilmiştir. Daha sonra üyelik fonksiyonunun değerleri kural editörüne girilmiştir. Kural girişleri tamamlandıktan sonra kural görüntüleme aracı ile sonuçlar görüntülenmiştir. Sonuçların oluşturulması esnasında durulaştırma metodu olarak ağırlık merkezi metodu kullanılmıştır.

2.2.1. Bulanık kümeler

Matematikte, bulanık kümeler (belirsiz kümeler), elemanları dereceye sahip kümeler gibidir. Fakat bu kümelerin 1920'lerde Polonyalı mantıkçı Jan Łukasiewicz tarafından geliştirilen Çoklu Değer Mantığından (ÇDM) ayırt edilmesi gerekmektedir. Bulanık kümeler, 1965'te klasik set kavramının bir uzantısı olarak Zadeh (1965) ve Klaua (1965) tarafından bağımsız olarak tanıtılmıştır. Aynı zamanda, Sali (1965) soyut bir cebirsel bağlamda incelemiş ve bir L-ilişkisi adı verilen daha genel bir yapıyı tanımlamıştır. Dilbilimi (De Cock, Bodenhofer & Kerre, 2000), karar verme (Kuzmin 1982) ve kümelenme (Bezdek 1978) gibi farklı alanlarda kullanılan bulanık ilişkiler, L'nin özel ilişkilerdir. Fakat burada L'nin birim aralığının $[0, 1]$ olması gerekmektedir.

Klasik küme teorisinde, bir kümedeki elemanların üyeliği, iki değerli koşula göre ikili terimlerle değerlendirilir. Yani bir öge bir kümeye ya aittir veya değildir. Bunun aksine, bulanık küme teorisi, bir kümedeki elementlerin üyeliğinin kademeli olarak değerlendirilmesine izin vermektedir. Bunu, gerçek birim aralığında $[0, 1]$ değerli bir

üyelik fonksiyonu yardımı ile açıklanmaktadır. Bulanık kümeler, klasik kümeleri genelleştirmektedir. Klasik kümelerin gösterge işlevleri (karakteristik fonksiyonlar), bulanık kümelerin üyelik fonksiyonlarının özel durumlarıdır. Burada tek şart değerleri sadece 0 veya 1 alınması gerekmektedir (Dubois ve Prade, 1988). Bulanık küme teorisi, biyoinformatik gibi bilginin eksik veya kesin olmadığı geniş bir alan yelpazesinde kullanılabilirler (Liang ve ark 2006).

Bu çalışmada iki girişli ve bir çıkışlı bulanık mantık kümeleri kullanılmıştır. Birinci küme olarak terminaller arası uzaklık alınmıştır. İkinci küme olarak, bir çalışanın terminale olan uzaklığı alınmıştır. Çıkış kümesi olarak, sahadaki toplam terminal sayısı alınmıştır.

Hesaplamalar yapılırken birinci küme terminaller arası uzaklıklar evrensel yürüme hızı olan saatte beş kilometre yürüme hızı ve ortalama ölçülere sahip bir yetişkinin hareket hızı 1,2 m/s - 1,4 m/s olan (Schneider ve Kirchberger, 2007) göre hesaplanmıştır. Bu hız verilerine göre sağlıklı bir insan bir saniyede yaklaşık yüz otuz santimetre yol almakta olup, acil durum toplanma alanlarına bu hıza göre yürüdüğü ön görülmektedir. Dolayısıyla, eğer çalışan kendi bölgesinde sayıma katılamıyorsa, kendi bölgesine en yakın olan bölgeye de bu hız sabiti ile yürüdüğü kabul edilmektedir.

Terminaller arası uzaklık kümesi dört farklı değişkenden oluşmaktadır. Birinci değişken ifadesi sıfır ile yetmiş beş metre arasında kalan mesafeyi ifade eder. İkinci değişken ifadesi ise elli metre ile yüz yetmiş beş metre arasındaki mesafeyi ifade eder. Üçüncü değişken yüz altmış ile üç yüz on metre arasındaki mesafeyi ifade eder. Dördüncü ve son değişken ise iki yüz elli metreden daha uzak olan mesafeleri ifade eder.

İkinci küme olan terminale uzaklık kümesinde de, birinci değişken ifadesi sıfır ile yetmiş beş metre arasında kalan mesafeyi ifade eder. İkinci değişken ifadesi ise elli metre ile yüz yetmiş beş metre arasındaki mesafeyi ifade eder. Üçüncü değişken yüz

altmış ile üç yüz on metre arasındaki mesafeyi ifade eder. Dördüncü ve son değişken ise iki yüz elli metreden daha uzak olan mesafeleri ifade eder.

Ayrıca birinci kısıt olan sayım bölgelerindeki sayım işleminin beş dakika içerisinde tamamlanma zorunluluğu, müdahaleye ihtiyacı olan personele hızlı şekilde ulaşmak içindir. İkinci kısıt olan bir adet dijital sayım cihazına elli beş kişinin düşmesidir. Bu hesaplama bir kişinin sahada bulunan ve dijital sayım için kullanılan terminale kart okutma veya biyometrik parmak izi okutma süresi bir buçuk ila iki saniye arasında olduğu göz önüne alınarak hesaplanmıştır. Genel kural olarak hesaplama iki saniye okutma süresi göz önüne alınarak yapılmıştır. İki okutma süresi arasında ise iki saniyelik zaman kaybı bulunmaktadır. Bu zaman kaybı, veri tabanına işlemin aktarılması için de gereklidir.

Çıkış kümesi olan sahadaki toplam terminal sayısı kümesi, bu tezde konu olan orta ve büyük ölçekli bir endüstriyel tesisteki yapılacak olan dijital sayıma göre hesaplanmıştır. Orta ve büyük ölçekli bir endüstriyel tesisteki kişilerin sayımı için giriş kümelerinde belirtilen zaman kısıtları göz önünde tutularak hesaplama yapılmıştır.

Sahadaki toplam terminal sayısı kümesi dört farklı değişken tarafından oluşmaktadır. Birinci değişken ifadesi on ila on beş adet terminalin sahada bulunması durumudur. İkinci değişken ifadesi sahada sekiz ila otuz adet terminalin bulunmasını ifade eder. Üçüncü değişken ifadesi yirmi beş ila kırk beş adet terminalin bulunmasını ifade eder. Dördüncü ve son değişken ifadesi sahada kırk adet ila sonsuz adet terminalin bulunması durumunu ifade eder.

2.2.2. Dilsel değişkenler, terimler

Matematikte değişkenler genellikle sayısal değerleri baz alırken, bulanık mantık uygulamalarında, sayısal olmayan değerler baz alınır. Genellikle kuralların ve olguların ifadesini kolaylaştırmak için kullanılır (Zadeh, 1996). Doğal diller her zaman bulanık bir değer ölçeğini ifade etmek için yeterli değer terimleri

içermediğinden, dilsel değerleri sıfatlar veya zarflarla değiştirmek yaygın bir uygulamadır. Örneğin, doğal yaşını belirtmek yerine oldukça yaşlı ve biraz daha genç olan ek değerleri ifade edilebilir.

Bulanıklaştırma işlemleri, matematiksel giriş değerlerini bulanık üyelik fonksiyonlarına eşleştirmek için kullanılırken, tersine bulanıklaştırma işlemleri, bulanık çıkış üyelik fonksiyonlarını karar veya kontrol amacıyla kullanılabilen “keskin” bir çıktı değerine eşlemek için kullanılmaktadır.

Bu tezde iki giriş ve bir çıkış kümesinin dil ifadeleri tablo halinde aşağıda belirtilmiştir. Birinci giriş kümesi olan terminaller arası uzaklık kümesi dilsel değişkenleri Tablo 2.2.’de gösterilmiştir.

Tablo 2.2. Terminaller arası uzaklık kümesi dilsel değişkenleri

Uzaklık (Metre)	Dilsel Değişkenler
0-75	Çok Yakın
50-175	Yakın (Komşu)
160-310	Uzak
250- ~	Çok Uzak

İkinci giriş kümesi olan terminale uzaklık kümesi dilsel değişkenleri Tablo 2.3.’de gösterilmiştir.

Tablo 2.3. Terminale uzaklık kümesi dilsel değişkenleri

Uzaklık (Metre)	Dilsel Değişkenler
0-75	Çok Yakın
50-175	Yakın (Komşu)
160-310	Uzak
250- ~	Çok Uzak

Çıkış kümesi olan sahadaki toplam terminal sayısı kümesi dilsel değişkenleri Tablo 2.4.’de gösterilmiştir.

Tablo 2.4. Toplam terminal sayısı kümesi dilsel değişkenleri

Terminal Sayısı (Adet)	Dilsel Değişken
0-15	Çok Az
10-30	Az
25-45	Fazla
40- ~	Çok Fazla

2.2.3. Üyelik fonksiyonları

Bulanık mantık, üyelik değerleriyle Boole mantığını taklit edecek şekilde çalışır. Bunun için temel işlemler “VE”, “VEYA”, “DEĞİL” olarak gerçekleşir. Bunun birkaç yolu var. Ortak bir yerine Zadeh operatörleri denir: Tablo 2.5.’de bu operatörler karşılaştırılmıştır.

Tablo 2.5. Üyelik fonksiyonlarında VE, VEYA, DEĞİL

Boole	Fuzzy
VE (x,y)	VE
VEYA (x,y)	VEYA
DEĞİL	DEĞİL

Doğru / 1 ve yanlış / 0 için, bulanık ifadeler, Boole ifadeleriyle aynı sonucu üretir. Ayrıca, doğada daha dilsel olan ve uygulanabilecek çit denen diğer operatörler de vardır. Bunlar genellikle matematiksel bir formül kullanarak bir kümenin anlamını değiştiren çok, ya da biraz gibi zarflardır. Bununla birlikte, isteğe bağlı bir seçim tablosu her zaman bulanık mantık işlevi tanımlamaz. Belirli bir seçim tablosunun bulanık mantık fonksiyonunu tanımlayıp tanımlamadığını ve minimum ve maksimum bileşenlerin tanımlanan kavramlarına dayanarak bulanık mantık fonksiyon sentezinin basit bir algoritmasının önerildiğini tanımlamak için bir ölçüt oluşturulmuştur. Bulanık bir mantık fonksiyonu, asgari bir bileşenin, bu alandaki işlev değerinden büyük veya ona eşit olan akım alanının değişkenlerinin bir birleşimi olduğu minimum (minimum) bileşenlerin birleşimini temsil eder (eşitsizlikteki işlev değerinin sağında) işlev değeri).

Başka bir AND / OR operatörü kümesi çarpmaya dayanır. Şöyle ki:

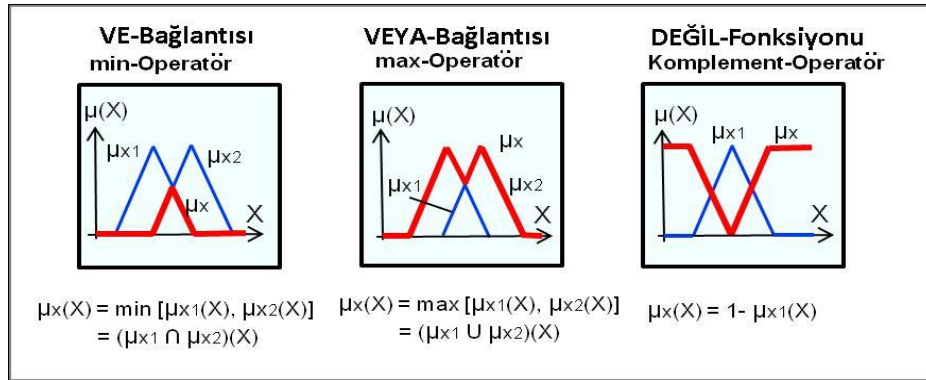
$$x \text{ VE } y = x * y \text{ VEYA } y = 1 - (1 - x) * (1 - y) = x + y - x * y$$

Burada belirtmek gerekir ki $1 - (1 - x) * (1 - y)$ c ise şurdan gelir:

$$x \text{ OR } y = \text{DEĞİL}(\text{VE}(\text{DEĞİL}(x), \text{DEĞİL}(y))) \text{ VEYA } y = \text{DEĞİL}(\text{VE}(1 - x, 1 - y))$$

$$x \text{ VEYA } y = \text{DEĞİL}((1 - x) * (1 - y)) \text{ VEYA } y = 1 - (1 - x) * (1 - y)$$

Şekil 2.13.'de üyelik işlemleri bağlama operatörleri gösterilmiştir.



Şekil 2.13. Üyelik işlemlerini bağlamak için VE, VEYA, DEĞİL operatörleri (Alt kümeler)

2.2.4. Bulanık kurallar

Bulanık mantığın en önemli kuralı “EĞER/ÖYLEYSE” (IF/THEN) kuralıdır. Bu kurallar girdi veya hesaplanmış doğruluk değerlerini istenen çıktı doğrulaması değerleriyle eşleştirir.

Örnek: EĞER ortam soğuksa ÖYLEYSE fanın hızı durduruldu. EĞER ortam soğuksa ÖYLEYSE fanın hızı yavaş. EĞER ortam ılımlı ise ÖYLEYSE fanın hızı istenen düzeyde EĞER ortam sıcaksa ÖYLEYSE fanın hızı yüksek.

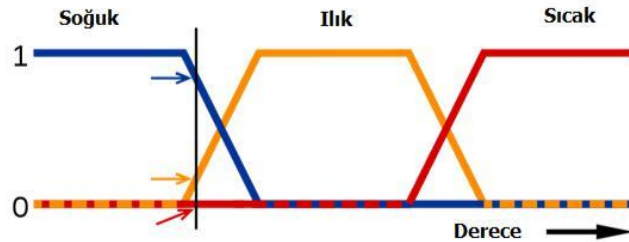
Belirli bir sıcaklık verildiğinde, bulanık değişkenin yüksek değışkene kopyalanan belirli bir doğru değeri vardır. Bir çıkış değışkeni birkaç ÖYLEYSE parçasında meydana gelirse, ilgili EĞER bölümlerinden gelen değerler VEYA operatörü kullanılarak birleştirilir.

2.2.5. Bulanık mantık uygulaması

Bulanık mantık uygulaması üç aşamadan oluşur. Bunlar sırasıyla:

- Tüm giriş değerlerini bulanık üyelik fonksiyonlarına sokulur,
- Bulanık çıktı işlevlerini hesaplamak için kural tabanındaki tüm geçerli kurallar yürütülür
- "Net" çıkış değerlerini almak için bulanık çıkış fonksiyonlarını ortadan kaldırılır.

Bulanık mantığı aşağıdaki Şekil 2.14.'te daha net bir şekilde açıklanmaktadır.



Şekil 2.14. Bulanık mantık sıcaklık

Bu Şekil 2.14.'de, soğuk, sıcak ve kızgın ifadelerin anlamları, bir sıcaklık ölçeğini haritalayan işlevlerle temsil edilmektedir. Bu ölçekte bir noktanın, üç fonksiyonun her biri için bir "gerçek değeri" vardır. Görüntüdeki dikey çizgi, üç okun (doğruluk değerlerinin) ölçüldüğü belirli bir sıcaklığı temsil eder. Kırmızı ok sıfıra işaret ettiğinden, bu sıcaklık "sıcak değil" olarak yorumlanabilir. Turuncu ok (0.2 ile işaret ediliyor) bunu "hafifçe ılık" ve mavi ok (0,8 işaret ediyor) "oldukça soğuk" olarak tanımlanabilir.

Bulanık kümeler genellikle üçgen veya yamuk şekilli eğriler olarak tanımlanır, çünkü her bir değer, değer arttığı bir eğime, değer 1'e eşit olduğu bir tepe noktasına (0 veya daha fazla bir uzunluğa sahip olabilir) ve bir eğim değer azalmaktadır. Sigmoid fonksiyonu kullanılarak da tanımlanabilirler (Wierman, 2010).

En yaygın kullanılan bulanık mantık önermeleri şu şekildedir:

- Monoidal t-norm-tabanlı önermeli bulanık mantık (MTL),
- Temel önermeli bulanık mantık (BL),
- Łukasiewicz bulanık mantığı (LL),

- Gödel bulanık mantığı (GL),
- Ürün bulanık mantığı (PL),
- Bazen Pavelka mantığı olarak da adlandırılan değerlendirilen sözdizimi ile bulanık mantık (EVL).

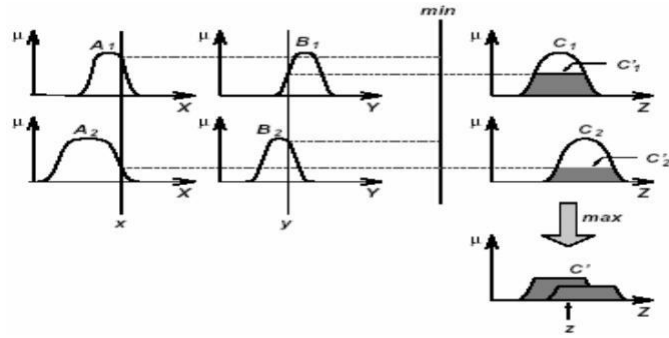
2.2.6. Mamdani bulanık modeli:

Bu çalışmada Mamdani Bulanık Modelini uygulayacağımız için bu modelin testler öncesinde tanıtılması faydalı olacaktır. Mamdani sistemleri, doğal dilde ifade edilen EĞER-ÖYLEYSE kuralları biçiminde bir girdi-çıkı ilişkisi hakkında uzmanlık bilgisini içermektedir. Bu, modelleme ve benzetim için özellikle çekici bir özelliktir. Mamdani Bulanık modelini kısaca kesin olmayan sonuçların bir araya getirilmesinden olası bir kesin sonucun çıkarıldığı süreç veya süreçler olarak tanımlanabilir (Pal, 2015). Mamdani, resmi çıkarım kurallarını kullanarak diğer bazı bilinenlerden yeni gerçeklerin türetilmesini amaçlayan bir prosedürdür. Yani, mantıksal bir çıkarım değildir. Hiçbir mantıksal implikasyon içeride değildir ve bu nedenle, Mamdani’de herhangi bir “modus ponens pro-ceeds” söz konusu değildir (Klawonna ve Novák, 1996).

Bu model İngiliz bilim adamı Ebrahim Mamdani tarafından geliştirilmiştir. Mamdani bulanık sistemler, özgün endüstriyel süreçleri kontrol etmekle görevli insan operatörlerinin performansını taklit etmek için tasarlanmıştır (Moraga, 2012). İlk kez buhar makinelerinin kazan bileşimini denetlemek için önerilmiştir.

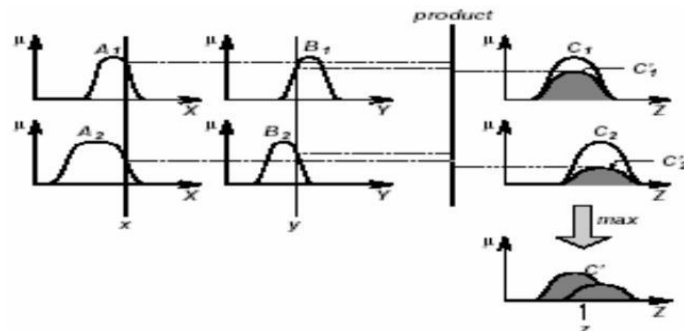
Mamdani sistemleri "EĞER X A olursa ÖYLEYSE Y B olur" şeklindeki "EĞER-ÖYLEYSE" kurallarından oluşur. Örneğin "EĞER basınç yüksekse ÖYLEYSE ses düşüktür". Bu formülleştirmede EĞER öncü ve ÖYLEYSE tamamlayıcıdır.

Aşağıda Şekil 2.15. ve Şekil 2.16.’da iki-kurallı bir bulanık girişim sistemlerdir. Sistem çıkış z’nin iki kesin giriş olan x ve y’den nasıl elde edildiğini açıklamaktadır.



Şekil 2.15. Mamdani bulanık mantık (T-norm ve S-norm - min ve max)

Yukarıdaki şekilde de açıklandığı üzere Mamdani bulanık mantık modeli iki girişli, iki kurallı ve tek çıkışlı bir bulanık sistemdir. Mekanizmanın işleyiş biçimini şu şekilde açıklanabilir. Öncelikle sistemin iki girişi olan x ve y'nin kesin değer olarak adlandırdığımız herhangi noktadaki değerlerine göre önce kuralın tanımladığı giriş bulanık kümesinde her bir kural için ayrı ayrı olmak üzere bu girişlerin üyelik dereceleri belirlenir. Oluşturulan bu keskin üyelik dereceleri min operatöründen geçirilir. Geçiş sonrasında ortaya çıkan ve kuralın tanımladığı en küçük üyelik derecesi kadar seviyede kırpılmış çıkış bulanık kümesi oluşturulur. Uygulanan bu işlemin her bir kural için ayrı ayrı uygulanması ve uygulama sonunda kural sayısı kadar çıkış bulanık kümesi oluşturulur. Elde edilen çıkış bulanık kümelerinin birleşiminin alınması için max operatöründen geçirilir. Bu işlem sonunda yine bir bulanık küme elde edilir. Bu işlem sonunda aslında hala daha kesin diyebileceğimiz bir değere ulaşmış sayılmamaktadır. Bu nedenle işleme devam edilmesi ve çıkış bulanık kümesi durulandırma işleminden geçirilmesi gerekmektedir.



Şekil 2.16. Mamdani bulanık mantık (T-norm ve S-norm - cebirsel çarpım - max)

Şekil 2.16.'da da resmedildiği gösterildiği gibi işlemde eğer cebirsel çarpım T-norm operatörü olarak ve max işlemi ise S-norm operatörü olarak seçilirse bu işlemin

sonucu her bir kuralın cebirsel çarpımı ile işaretlenen ateşleme gücü tarafından azaltılan bir bulanık kümeye eşleme yapılmak suretiyle belirlenebilir.

Bu çalışmada Mamdani bulanık mantık yöntemi MATLAB programı aracılığı ile kullanılmıştır. Mamdani uygulanarak ağırlık merkezi durulaştırma metoduna göre sonuç ekranılar oluşturulmuştur.

BÖLÜM 3. SİSTEM TANITIMI VE YÖNTEM

3.1. Dijital Sayım Programı Geliştirilmesi

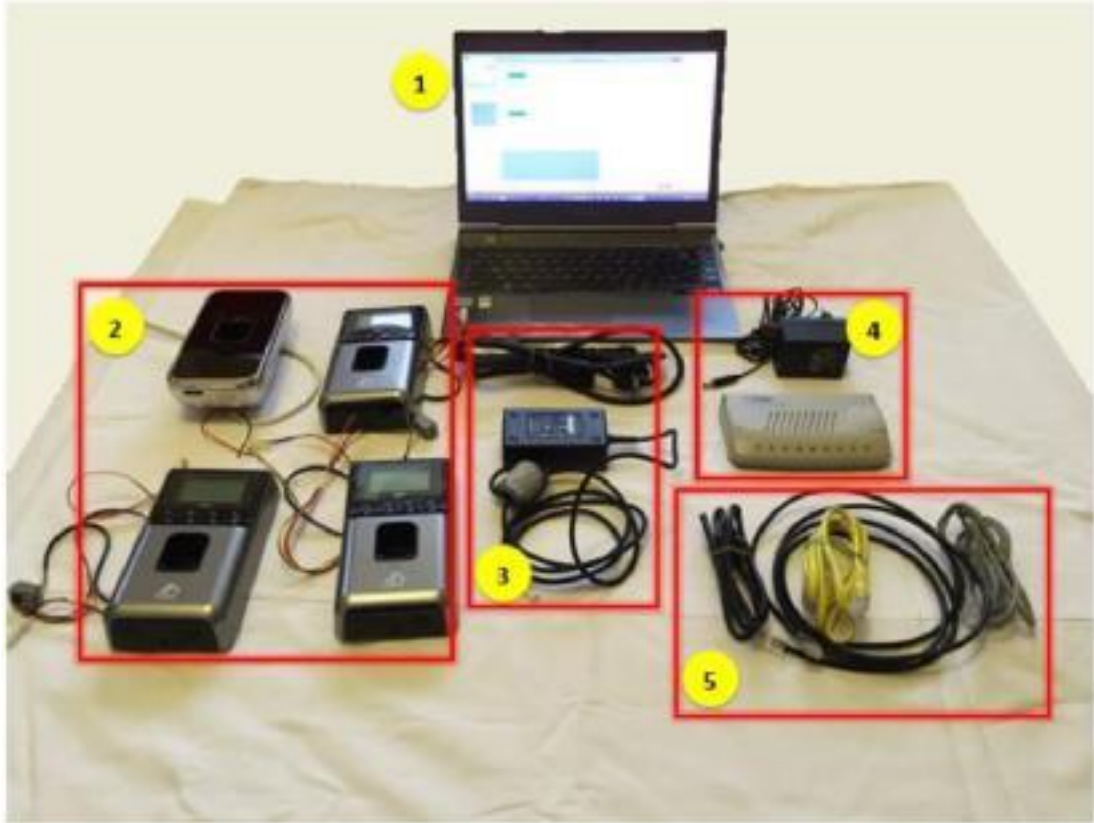
Bu tezde verilerin işlenmesi, eksik kişilerin belirlenmesi sonuçların çıkarılması, veri tabanına erişim, haberleşme, yedekleme işlemlerinin yapılması ve uygulamanın çalışma sürenin hesaplaması için Visual Studio C# programı ile uygulama yazılmıştır. Bu uygulama amacı detaylı olarak şu şekildedir.

- A. Kullanıcı gelen verileri en kolay ve anlayabilir şekilde görsün.
- B. Tek bir tuşa basarak sayıma katılan ve katılmayan çalışanlar hakkında bilgi sahibi olsun.
- C. Eksik kişilerin bilgileri ekranda gözüksün.
- D. Acil durum müdahale ekiplerine bilgi aktarılmasında özet liste çıkarılsın
- E. Sayıma katılan kişilerin, kendi bölgelerinde, yakın/komşu bölgede, uzak bölgede veya çok uzak bölgede girip girmediklerinin özeti liste olarak gözüksün.

işlemleri yapılmaktadır. Ayrıca tesiste/işletmede çalışanların görev bölgelerinin gösterilmesi, gerekirse görev bölgesi değişikliğinin yapılabilmesi, yeni çalışanın sisteme eklenmesi, ayrılan çalışanın sistemden çıkarılması işlemleri ayrı bir form ekranı üzerinden yapılmaktadır. Bu sayede tesise/işletmeye yeni katılan veya ayrılan kişinin sistem sayesinde yönetilmesi sağlanmaktadır

3.1.1. Sistem tanıtımı

Aşağıdaki Şekil 3.1.'de sistemin genel görüntüsü bulunmaktadır. Aşağıda maddeler halinde sistem parçalarının isimleri verilmiştir.



Şekil 3.1. Sistemin genel görüntüsü

1. Sunucu bilgisayar.
2. Dijital sayım cihazları VIRDI AC 2100: 3 adet , VIRDI AC 2500: 1 adet.
3. Dijital sayım cihazları güç adaptörleri.
4. Yerel ağ bağlantısı için dağıtıcı parça ve güç adaptörü (SWITCH).
5. Çeşitli uzunluklarda CAT5 cinsi kablo.

3.1.2. Uygulama hazırlama aşaması

Uygulamanın yazılması için Viridi AC 2100 terminallerinin kullanmış olduğu paket programdan faydalanılmıştır. Viridi marka terminallerini kullanmış olduğu paket programın ismi UNIS Remote Manager olup, sahada kullanılan terminallerin çalışıp çalışmadığı denetimi, toplu kullanıcı eklenme işlemleri, izinli personel işlemleri bu program sayesinde yapılmaktadır. Hazırlanan bu tezde UNIS Remote Manager programının veri tabanı tespit edilmiş ve veri tabanına erişilmek istenmiştir. Veri tabanı belirli bir parolaya sahip olduğu için ilk aşamada veri tabanı içerisine girilememiştir. Veri tabanı şifresinin belirlenmesi işleminden sonra veri tabanı içerisine girilmiştir. Fakat veri tabanı içerisine girildiğinde herhangi bir tablo ile karşılaşmamıştır. Sistemin sürekli veri tabanı tablolarını kullandığı tespit edilmiştir.

Tablolara erişimin sağlanması için sistemin durdurulması yani OSI katmanları içerisinde olan oturum katmanının durdurulması gereklidir. Oturum katmanını durdurmak için Bilgisayar Yönetimin açılmış ve hizmetler bölümüne girilmiştir. Hizmetler bölümünde UNIS_SERVER, UPnP ve UWorkServer hizmetleri durdurulmuştur. Veri tabanı tekrar kontrol edilmiştir ve erişim sağlanmıştır. Veri tabanı içerisindeki 77 adet tablo tarandığında, gerekli kayıtların tutulması işleminin TUSER tablosu ile TENTER tablosu tarafından yapıldığı belirlenmiştir.

Terminallerin birbirleri ile ve sunucu ile iletişime geçmesi için her terminal üzerinden statik IPv4 tanımlanması ve IPv4 alt ağ maskesi belirlenmesi gereklidir. Ayrıca varsayılan ağ geçidi çıkışı da belirlenmelidir. Bu atamalar sırasıyla statik sistemde varsayılan ağ geçidi için 169.254.226.252 alt ağ maskesi olarak 255.255.0.0 adresleri belirlenmiştir. Her terminal için varsayılan ağ geçidi son numarası bir arttırılarak terminal makinelerinin adresleri belirlenmiştir.

Sisteme kayıtlı kullanıcılar eklendikten sonra bu çalışanların kartlarının veya parmak izlerinin okutulması işleminin yapılması gereklidir. Bu işlem için terminallerde bulunan parmak izi tanımlama yuvasından faydalanılmıştır. Bu işlem için terminal üzerindeki F3 tuşuna uzun süre basılı tutularak menü ekranına girilmiştir. Açılan menü ekranından kullanıcı alt menüsüne girilerek, tanımlama yapılacak kullanıcının belirlenmesi gereklidir. Belirlenen kullanıcı numarası girildikten sonra, kullanıcının parmak izini okutması istenmektedir. Bu işlem sonrasında tesiste/işletmede çalışan kişi sisteme tam olarak kayıt altına alınmış olacaktır.

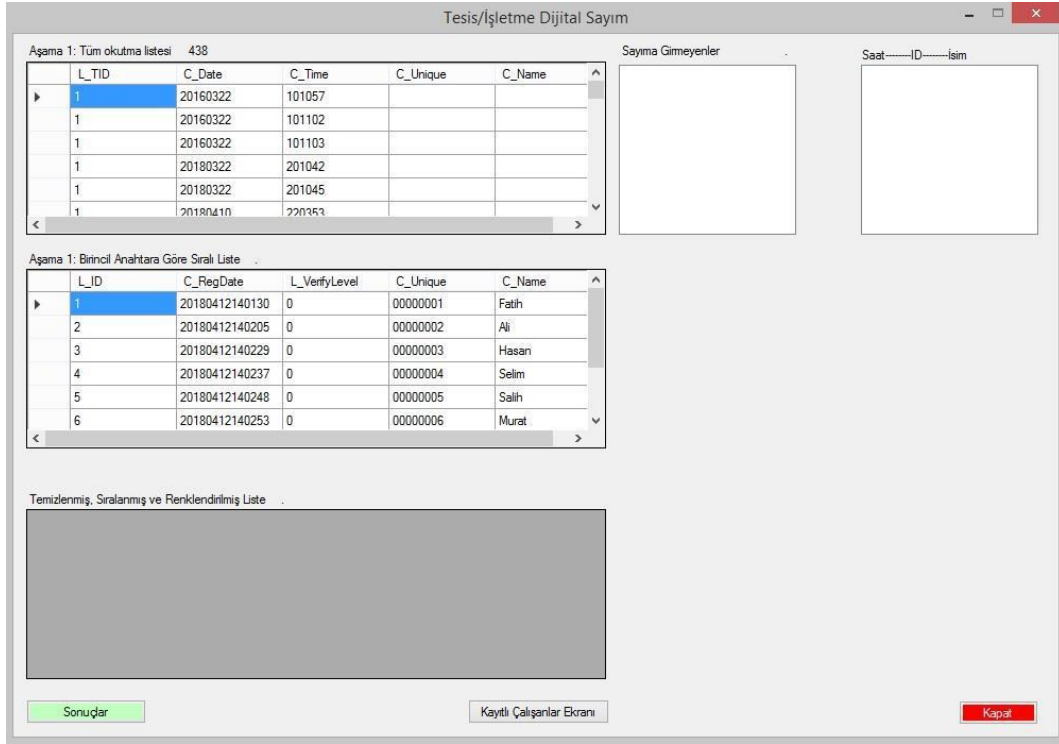
C# ile yazılan form uygulamasında veri tabanından veriler okunmuş ve ilk kayıtlar elde edilmiştir. Daha sonraki işlem olan tesiste/işletmede çalışanların tekrar okutulması işleminde verilerin ilgili tablolara yazılmadığı görülmüştür. Oturum katmanında durdurulmuş olan hizmetlerin otomatik olarak çalışmadığı tespit edilmiştir. Durdurulmuş olan hizmetler tekrar geri açıldığında veri tabanı tablosuna erişimin yapılamadığı tespit edilmiştir.

Ortaya çıkan bu sorununun giderilmesi UNIS Remote Manager sisteminin sürekli olarak kullandığı veri tabanını bir kopyası alınarak çözülmüştür. Yapılmış olan yazılımın sonuçları göstermesi için kullanılan tuşun tıklanması esnasında dosya silme ve yaz komutları kullanılarak veri tabanının bir kopyasının oluşturulması sağlanmıştır. Her tıklama yapıldığında kopya olan veri tabanı ilk önce silinmekte ve daha sonra UNIS Remote Manager sisteminin veri tabanının kopyası tekrar alınmaktadır. Veri tabanının kopyası alınarak yapılan işlemlerin doğru çalıştığı belirlendi ve C# ile yazılan uygulama bu metot ile yapıldı. Bu işlem yerine tabloların tekrar güncellenmesi işlemi de kullanılabilirdi. Fakat tesiste/işletmede çalışan sayısı arttıkça güncelleme işlemi zaman kayıplarına yol açacaktı.

3.1.3. Yazılan uygulamanın işleyişi

Tesis/işletme Dijital Sayım yazılımı çalıştırıldığı zaman açılan ara yüzünde sadelik için sadece 2 adet tuş kullanılmıştır. Uygulamanın açılmasıyla beraber sistemin ilk kurulum tarihinden itibaren kayıtlı olan tüm veriler Tüm okutma listesi (datagridview1) tablosunda gösterilmekte ve toplam kayıt sayısı belirtilmektedir. Tüm okutma listesinin yanında bulunan sayı ise, sistemin kuruluş tarihinden sorgulamanın yapıldığı tarihe kadar olan süre içerisinde kaç okutma işleminin yapıldığını göstermektedir.

Birincil Anahtara Göre Sıralı Listesi (datagridview2) tablosunda çalışanların birincil anahtara göre sıralı listesi verilmektedir. Bu veri liste kontrol amaçlı tutulmuş olup, sisteme kayıtlı olan önemli kullanıcıların kolayca bulunmasına yardımcı olmaktadır. Programı kullanan kişi bu tablo sayesinde fare yardımı ile istediği kullanıcıyı hızlı şekilde görüntüleme imkânına sahip olur. Ayrıca programı kullanan kişi sonuçlar tuşuna bastığında bu tabloda yeşil renkle hücreler renklendirilecek olup, sayıma katılmış çalışanlar belirtilecektir. Birincil Anahtara Göre Sıralı Listesi yanında bulunan sayı ise, sisteme kayıtlı olan çalışan sayısını belirtmektedir. Bu sayının tutulma amacı, sonuçlar tuşuna basıldığında sayıma girenler ve sayıma çeşitli bölgelerde girenlerin toplamının kontrol edilmesini sağlar. Şekil 3.2.'de genel görüntü paylaşılmıştır.



Şekil 3.2. Tesis/İşletme dijital sayım ara yüzü (Boş listeler)

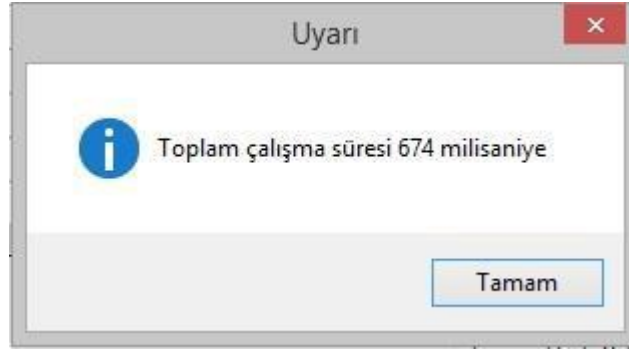
Sonuçlar tuşu sayesinde algoritma çalıştırılarak, tesiste/işletmede sayıma girmiş ve girmemiş kişilerin listelere dökülmesi sağlanmaktadır. Sayıma giren kişiler dört farklı liste içerisinde gösterilmektedir. Eğer çalışan tanımlı olduğu bölgede sayıma girdiyse Sayıma Kendi Bölgesinde Giren listesinde kayıt çıkmaktadır. Eğer çalışan tanımlı olduğu bölgeye yakın olan bölgede sayıma girdiyse, Sayıma Yakın Bölgede Giren listesinde kayıtlı çıkmaktadır. Eğer çalışan tanımlı olduğu bölgeye uzak bölgede sayıma girdiyse Sayıma Uzak Bölgede Giren listesinde kayıtlı çıkmaktadır. Eğer çalışan tanımlı olduğu bölgeye çok uzak bölgede sayıma girdiyse Sayıma Çok Uzak Bölgede Giren listesinde kayıtlı çıkmaktadır. Eğer çalışan hiçbir bölgede sayıma girmediyse, kırmızı liste olan Sayıma Girmeyenler listesinde kayıtlı çıkmaktadır. Sayıma Girmeyenler listesinin toplam kişi sayısı hesaplatılıp liste üzerinde gösterilmektedir. Bu kırmızı liste, acil durum komuta merkezine kılavuz olacaktır. Şekil 3.3.'de dolu listelerin görseli paylaşılmıştır.

The screenshot displays the 'Tesis/İşletme Dijital Sayım' software interface. It features several data grids and summary panels. The top-left grid, 'Ajanlar 1: Tesis/İşletme Dijital Sayım', lists employees with columns for ID, Date, Time, Unique ID, and Name. The top-right panel, 'Sayım Gözetim', shows a large red area. The middle-left grid, 'Ajanlar 2: Birim/Avantaj Göre Sıralı Liste', lists employees with columns for ID, Date, Level, Unique ID, and Name. The middle-right panel, 'Büyük Algoritma Tablosu', shows a list of employees. The bottom-left grid, 'Temizlenmiş, Sıralanmış ve Renklendirilmiş Liste', lists employees with columns for ID, Date, Time, Unique ID, and Name. The bottom-right panel, 'Sayım Çıkış Bilgileri', shows a list of employees. The interface includes a 'Başlat' button at the bottom left and a 'Kapat' button at the bottom right.

Şekil 3.3. Tesis/İşletme dijital sayım ara yüzü (Dolu listeler)

Sonuçlar tuşuna basılması ile en altta çıkan tablo (datagridview3) olan Temizlenmiş, Sıralanmış ve Renklendirilmiş Liste'de sayıma katılan çalışanların kendilerini en son hangi bölgede sayıma dâhil ettiği bilgisi ekrana dökülmektedir. Bu bilgi dökümü, sistem saati kontrol edilerek her zaman en son sayıma girme bölgesini belirlemektedir. Son sayım bölgesine göre çalışanın isminin geçtiği hücre yeşil renk ile renklendirilip programı kullanan kişiye görsel olarak kolaylık sağlamaktadır. Sağ tarafta gözüken diğer altı küçük liste, bu son tablo sayesinde doğru olarak doldurulmaktadır.

Sonuçlar tuşunun tıklanması sonucunda en son ekrana çıkan mesaj kutusu ile yapılan işlemlerin kaç milisaniyede bittiği ekrana çıkmaktadır. Bu hesaplamanın tutulmasının sebebi ise arama algoritmasının cevap verme süresinin tutulması, iyileştirme gerekip gerekmediğinin sayısal olarak gözükmesi ve acil durum sürecinin sayım işlemi ve program işleminin toplamda ne kadar süre tutacağını hesaplanmasını kolaylaştırmaktır. Şekil 3.4.'de hesaplanmış süre ekranı paylaşılmıştır.



Şekil 3.4. Uyarı ekranı

Ara yüz içerisinde bulunan tablolarda L_TID kolonunda tesiste/işletmede bulunan çalışanın hangi terminalde sayıma girdiği, C_DATE kolonunda hangi günde sayıma girdiği, C_TIME kolonunda hangi saatte sayıma girdiği, C_UNIQUE kolonunda hangi birincil anahtar numarası, C_NAME ise çalışanın isim soy isim bilgilerini döküm yapmaktadır. Ayrıca tablolarda yeşil ile işaretlemeler yapılmaktadır. Bu işaretlemeler sayesinde arama algoritmasının rahatlatılması planlanmıştır. Yeşil işaretlemelerin olduğu tablolar acil durum komuta merkezindeki program kullanıcılarına, tesiste/işletmede bulunan çalışanların sayıma girdiği bilgisini karışık olarak dökmektedir.

Sonuçlar tuşuna basıldığı zaman ilk tabloda sayımın yapıldığı gün sistem tarihinden alınıp tüm tablodaki C_DATE kolonu ile karşılaştırma yapılmaktadır. Karşılaştırma sonucunda doğru tarihli olan satırların C_DATE hücresi yeşil renk ile boyanmaktadır. Bu boyamanın amacı, olabilecek sistem hatalarının görsel olarak tespit edebilmektir. Ayrıca birinci tabloda kart okuma, parmak izi okuma hatası yapılış ise bu hatalar ayıklanmakta ve tablodan silinmektedir. Yapılan tüm ayıklamalardan sonra tabloda kaç adet veri bulunduğu ara yüz tasarımının sağ üst köşesinde sayısal olarak verilmektedir.

Sonuçlar tuşuna basıldığı zaman asıl önemli işleyiş ikinci tabloda görüntülenmektedir. İkinci tabloda sayıma katılan çalışanların en son sayıma katıldığı bölge sistem tarafından kabul edilecek şekilde algoritma düzenlenmiştir. Bu düzenleme sonucunda çalışan eğer birkaç farklı noktada kendini sayıma dâhil etti ise, son sayım noktasını sisteme dâhil edip sonucu ikinci tabloda gösterecektir. İkinci

tablonun görsel olarak daha anlaşılabilir olması için C_NAME hücreleri yeşil renk ile boyanmıştır. C_NAME hücrelerinin yeşil renk ile boyanmasının sebebi, sisteme kendini okutan çalışanın son okuttuğu saati alma amacıdır. Bu sonuç doğrultusunda çalışanın birden fazla bölgede kendisini oluşturulan sisteme dahil etmesi ihtimaline karşı son okutma bölgesinin kayıt altında tutulmasıdır. Önceki okutmalar veri tabanında ve ara yüz tasarımında kayıt altında olmasına rağmen, oluşturulan listelerde son kayıt noktası göz önüne alınacaktır.

Diğer bir tuş olan kayıtlı çalışanlar tuşu ile yeni bir ara yüz açılmakta olup bu ara yüz arka planında farklı bir veri tabanı çalıştırılmaktadır. Çalıştırılan bu veri tabanının ismi listeler olup bu veri tabanı ile sistemin kullandığı UNIS veri tabanı birbirleri ile karşılaştırılmaktadır. Listeler veri tabanına yeni kayıt ekleme ve mevcut kayıt silme işlemleri ilgili tuşlar sayesinde yapılmaktadır. Doldurulacak olan alanların kontrolleri algoritma sayesinde kontrol edilmekte olup, programı kullanacak olan kişinin hata yapmasını engellemektedir. Şekil 3.5.'te Kayıtlı çalışanlar liste kontrol ekranı gösterilmiştir.



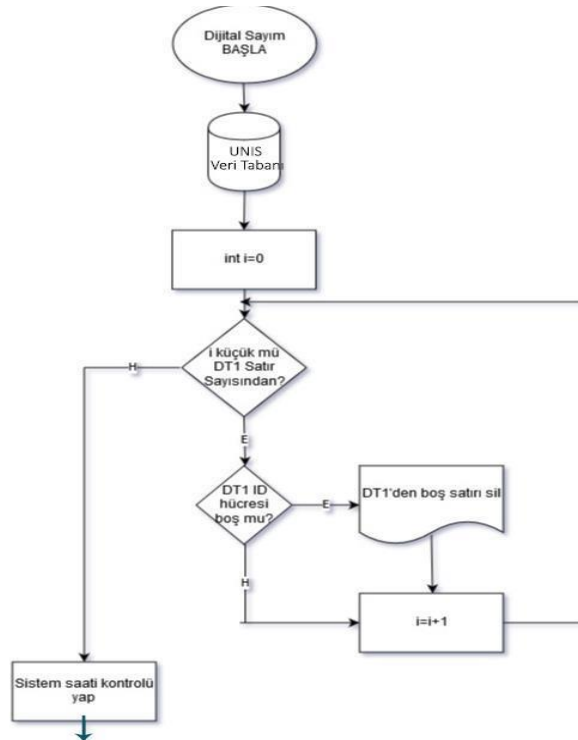
Şekil 3.5. Kayıtlı çalışanlar listesi

3.1.4. Uygulamanın akış diyagramı

Uygulamaya ait akış diyagramları kısımları halinde aşağıda gösterilmiş ve kısa izahları yapılmıştır.

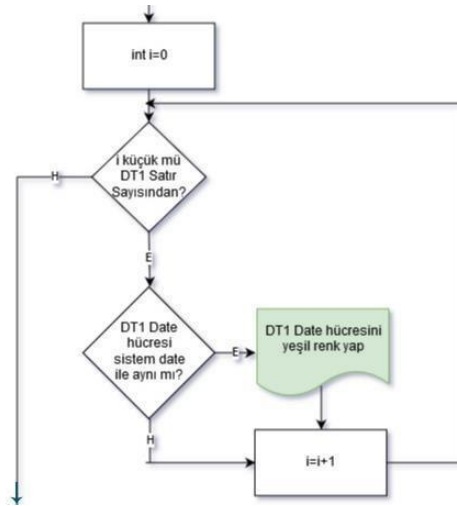
3.1.4.1. Dijital sayım bölümü

Programın execute dosyası çalıştırıldığında ilk olarak bilgilerin tamamı UNIS veri tabanında sorgulama ile çekilir. Veri kümesi halinde toplanan veriler, veri tablosu 1 içine doldurulur. Sayıma katılan kişilerin kartlarını veya parmak izlerini okutmalarında hızlı geri çekme işlemi olursa, sistem okumasını yapar fakat isim alanı ve birincil anahtar alanı boş kalır. Yukarıda tanımlı ilk veri tablosu 1 içerisindeki satır sayısı kadar arama işlemi yapılmaktadır. İkinci döngüde ise, hatalı okutmaların oluşturduğu boş satırların temizliği yapılmaktadır. Boş satırların temizlenme işlemi bittikten sonra, sistem saat kontrolü yapılmaktadır. Sistem saat kontrolünün yapılmasındaki amaç, sahadaki cihazların saat senkronizasyonunu sağlamaktır. Aşağıdaki Şekil 3.6.'da akış diyagramı paylaşılmıştır.



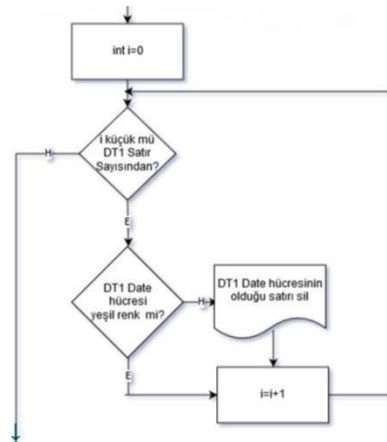
Şekil 3.6. Dijital sayım bölümü girişi

Veri tablosu 1 satır sayısı tekrar tutulup, sistem tarihi ile aynı olan satırların seçim işlemi ikinci döngü içerisinde sağlanmaktadır. Sistem tarihi ile veri tablosu 1 içerisindeki tarihler bir birlerini karşılıyorsa, tarih hücresi yeşil renk ile boyanır. Bu işlem satır sayısı tamamlanincaya kadar devam eder. Satır sayısı kadar yapılan bu işlem sonucunda programı kullanan kişiye, yeşil renkli hücreler gösterilmiş olacaktır. Şekil 3.7.'de anlatılanların akış diyagramı paylaşılmıştır.

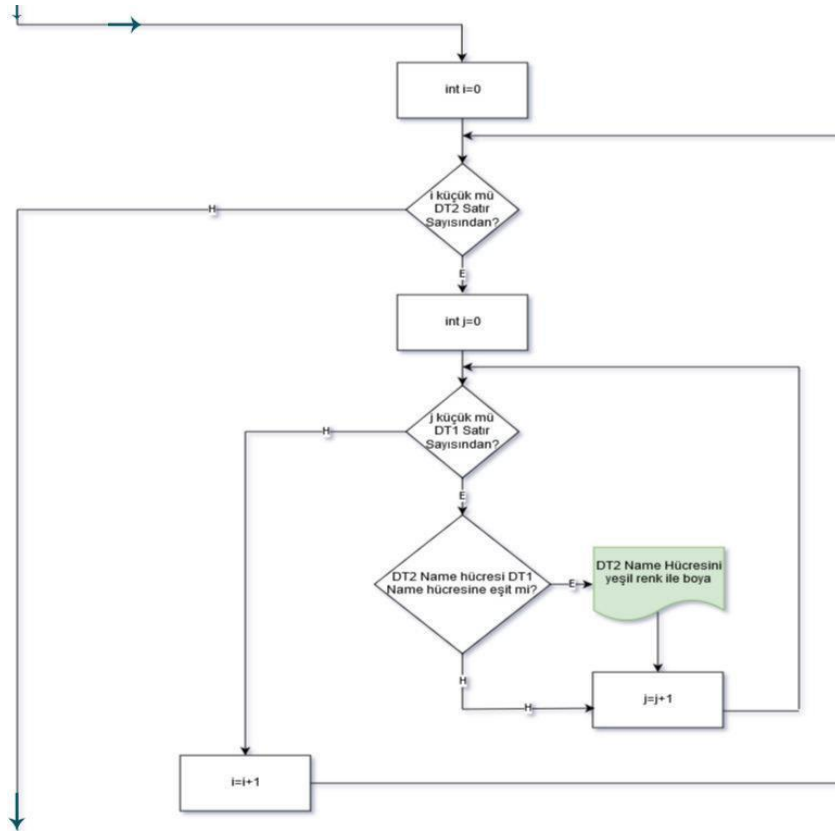


Şekil 3.7. Veri tablosu 1 tarih hücresi kontrolü

Veri tablosu 1 satır sayısı tekrar tutulup, veri tablosu 1 içerisindeki tarih hücresi yeşil olmayan satırların silinmesi işlemi ikinci döngü içerisinde yapılmaktadır. Böylelikle tarih hücresi yeşil olmayan veri satırları silinmektedir. Şekil 3.8.'de akış diyagramı gösterilmiştir.



Şekil 3.8. Veri tablosu 1 tarih hücresi yeşil olmayan satırların silinmesi

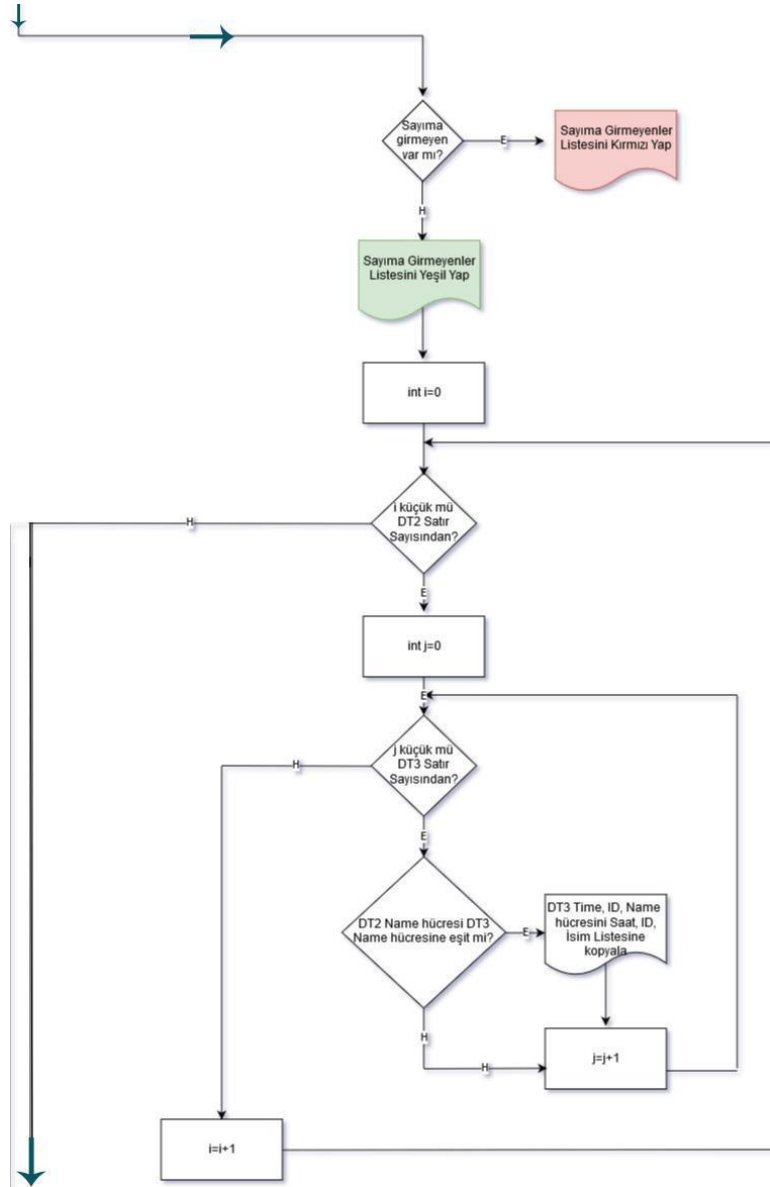


Şekil 3.9. Veri tablosu 1 ile veri tablosu 2'nin isim hücrelerinin karşılaştırılması

Sayım yapılan tarihe göre temizlenmiş olan veri tablosu 1 işlem yapılması için hazır durumdadır. Veri tablosu 2 satır sayısı tutulup ve veri tablosu 1 satır sayısı tutulup, bu iki tablo isim hücresi bulunduğu bölümler birbirleri ile karşılaştırılmaktadır. Şekil 3.9.'da görülen üçüncü karar verme bölümünde, veri tablosu 2 isim hücresinde veri tablosu 1 isim hücresine eşit ise veri tablosu 2 isim hücresinin yeşil ile renklendirilmesi yapılmıştır. Bu işlem veri tablolarının son satırına kadar devam eder.

Veri tablosu 2 isim hücresinde yeşil renklendirilmemiş hücreler var ise sisteme kayıtlı olan çalışanın sayıma katılmadığı bilgisini programı kullanan kişi göstermektedir. Şekil 3.10. birinci karar verme bölümünde veri tablosu 2 isim hücresinde yeşil ile renklendirilmemiş alan varsa, sayıma girmeyenler listesinin kırmızı ile renklendirilmesi yapılmaktadır. Veri tablosu 2 isim hücresinde yeşil ile

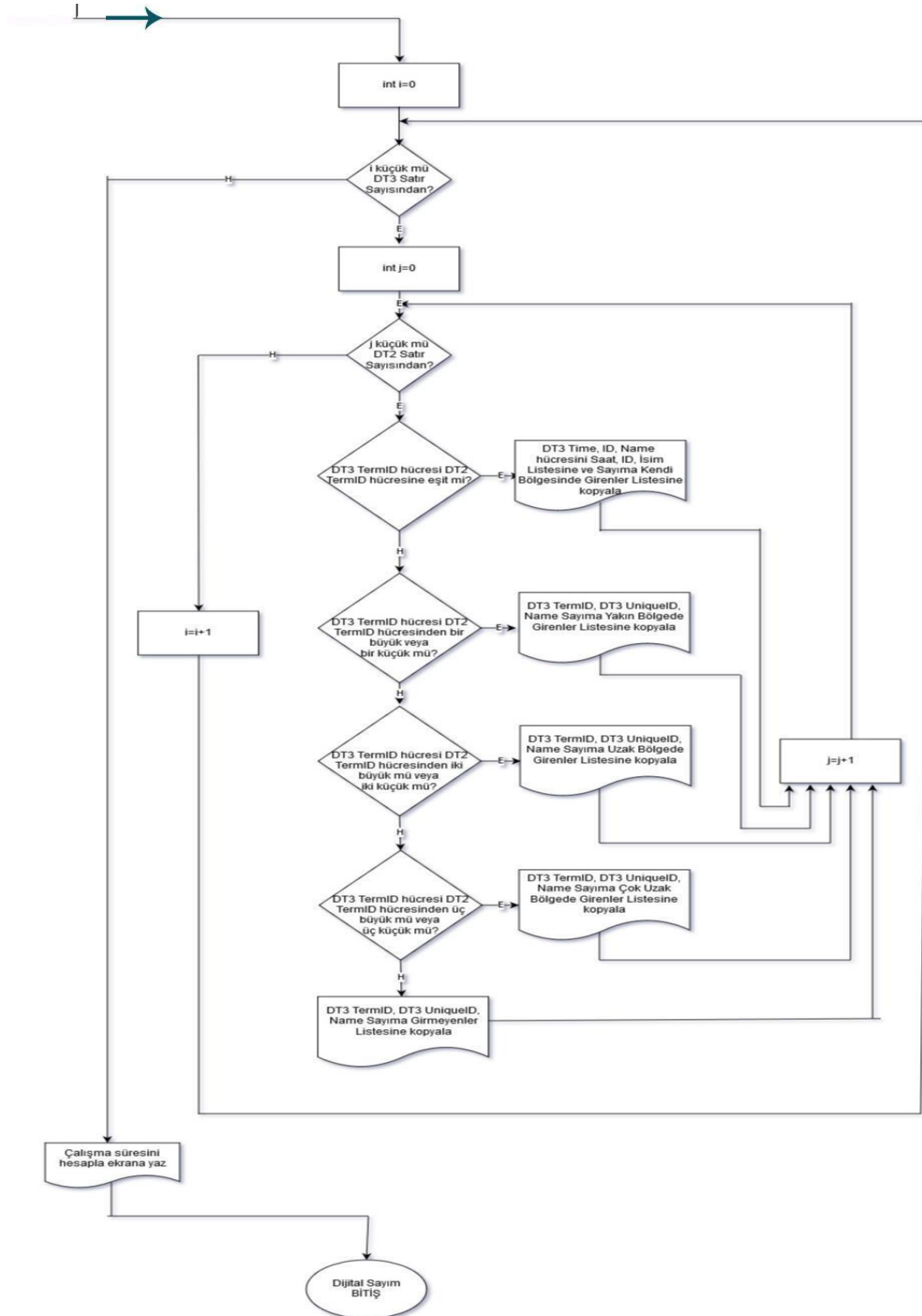
renklendirilmemiş alan yoksa sayıma girmeyenler listesinin yeşil ile renklendirilmesi yapılmaktadır.



Şekil 3.10. Sayıma girmeyenler listesinin renklendirilmesi ve veri tablosu 2 ile veri tablosu 3'ün karşılaştırılması

İkinci döngüde veri tablosu 2'nin satır sayısı ve üçüncü döngüde veri tablosu 3'ün satır sayısı tutulmakta olup bu iki veri tablosu dördüncü karar verme döngüsünde isim hücrelerinin birbirlerine eşit olup olmadığı kontrol edilmektedir. Bu kontrol sayesinde veri tablosu 3'ün isim hücresi yeşil renk ile renklendirilmektedir. Ayrıca veri tablosu üç isim hücresi yeşil renkli olan satırlardaki zaman, özgün numara ve

isim, Saat, ID, İsim listesine kopyalanmaktadır. Bu işlem veri tablosu 2 ve veri tablosu 3 satır sayısı bitine kadar yapılmaktadır.



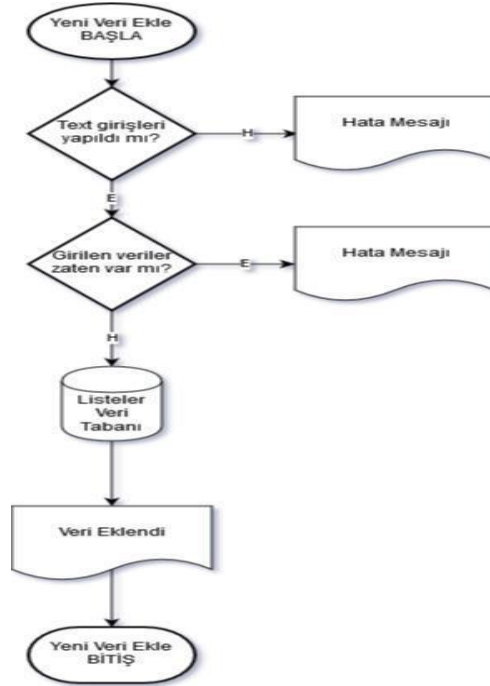
Şekil 3.11. Listelerin oluşturulması ve sıralı şekilde yazdırılması

Şekil 3.11.'de birinci döngüde, veri tablosu 3 satır sayısı tutulmaktadır. İkinci döngüde ise veri tablosu 2'nin satır sayısı tutulmaktadır. Üçüncü karar verme

döngüsünde ise veri tablosu 3 terminal numarası hücresi ile veri tablosu 2 terminal numarası karşılaştırılmaktadır. Eğer terminal numaraları birbirine eşit ise veri tablosu 3 ilgili satırındaki Terminal numarası, kişiye özgün birincil anahtarı ve kişi ismini sayıma kendi bölgesinde girenler listesine kopyalama yapılmaktadır. Dördüncü karar verme döngüsünde ise veri tablosu 3 terminal numarası hücresi ile veri tablosu 2 terminal numarası karşılaştırılmaktadır. Eğer terminal numaraları bir büyük veya bir küçük ise veri tablosu 3 ilgili satırındaki Terminal numarası, kişiye özgün birincil anahtarı ve kişi ismini sayıma yakın bölgede giren listesine kopyalanmaktadır. Beşinci karar verme döngüsünde ise veri tablosu 3 terminal numarası hücresi ile veri tablosu 2 terminal numarası karşılaştırılmaktadır. Eğer terminal numaraları iki büyük veya iki küçük ise veri tablosu 3 ilgili satırındaki Terminal numarası, kişiye özgün birincil anahtarı ve kişi ismini sayıma uzak bölgede giren listesine kopyalanmaktadır. Altıncı karar verme döngüsünde ise veri tablosu 3 terminal numarası hücresi ile veri tablosu 2 terminal numarası karşılaştırılmaktadır. Eğer terminal numaraları üç büyük veya üç küçük ise veri tablosu 3 ilgili satırındaki Terminal numarası, kişiye özgün birincil anahtarı ve kişi ismini sayıma çok uzak bölgede giren listesine kopyalanmaktadır. Kopyalama işlemleri tamamlandıktan sonra dijital sayım işlemi tamamlanmış olmaktadır. Son olarak yapılan tüm işlemlerin, ne kadar süre içerisinde yapıldığı hesaplanmakta ve ekrana milisaniye cinsinden yazdırılmaktadır.

3.1.4.2. Yeni veri girişi

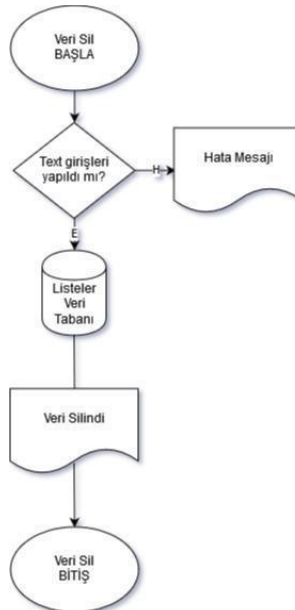
Veri girişi işlemi programı kullanan kişiye ayrı bir ekran açılmakta olup bu ekranın aktif edilmesi için kullanıcıyı Kayıtlı Çalışanlar Ekranı tuşuna basması gereklidir. Bu ekran aracılığı ile programı kullanan kişi sistemin kullandığı UNIS adlı veri tabanını ve programcılıkta çapraz doğrulama amaçlı kullandığımız listeler adlı veri tabanının kontrol edilmektedir. Listeler veri tabanına yeni veri ekle tuşu sayesinde veri ekleme işlemi yapılması sağlanmaktadır. İşlemin hatalarının kapatılması amaçlı uyarı mesajları ekranda gösterilmektedir. Şekil 3.12.'de akış diyagramı gösterilmiştir.



Şekil 3.12. Yeni verilerin sisteme girilmesi işlemi

3.1.4.3. Veri silme işlemi

Veri silme işlemi sayesinde ise programı kullanan kişi, tesisten/işletmeden ayrılan personelin veri tabanından silinmesi işlemini yapmaktadır. Uygun girişler yapılmadığı takdirde, ekranda hata mesajları çıkacak olup, kullanıcıya doğru işlem yapılması hakkında bilgi verilmektedir. Şekil 3.13.'te akış diyagramı gösterilmiştir.

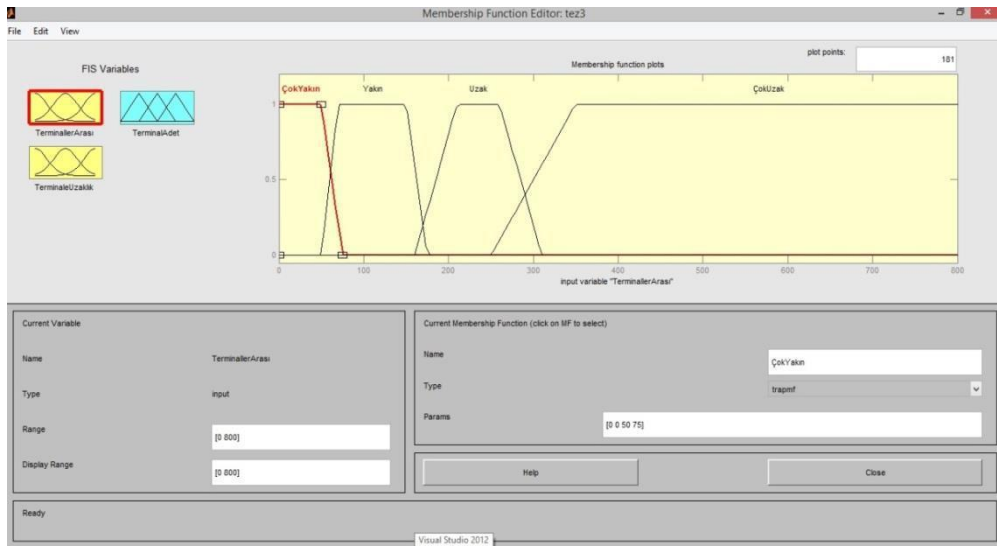


Şekil 3.13. Veri silme işlemi

3.2. Bulanık mantık uygulanarak sistemin modellenmesi

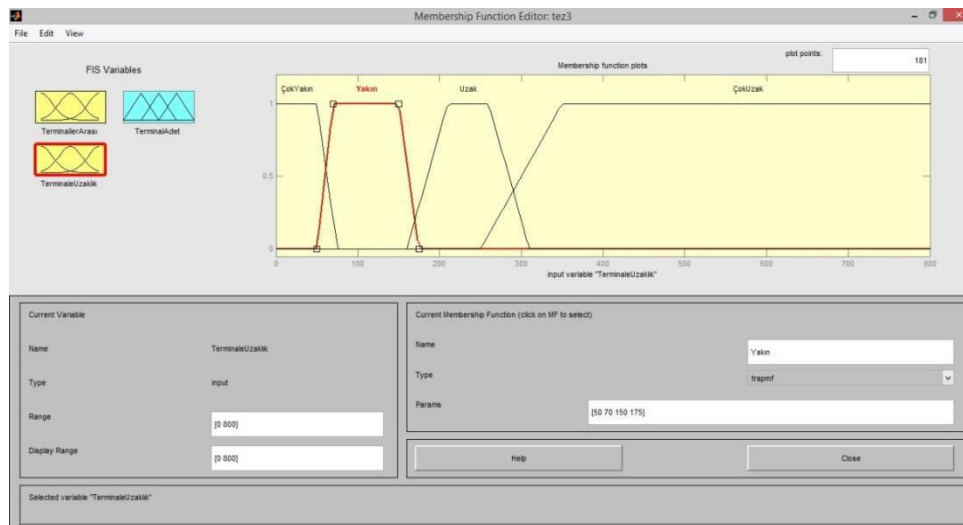
Bu çalışmadaki MATLAB programı ile hazırlanan bulanık üyelik fonksiyonları şu şekildedir.

Birinci giriş kümesi olan terminaller arası uzaklık üyelik fonksiyonu Şekil 3.14.'teki gibidir.



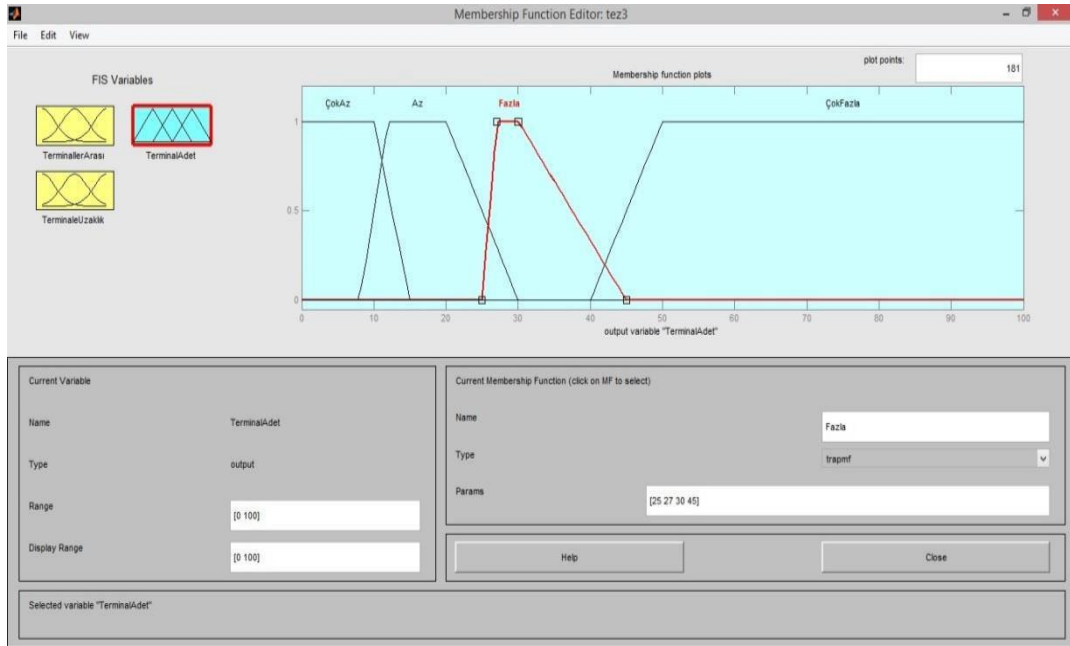
Şekil 3.14. MATLAB programı terminaller arası uzaklık üyelik fonksiyonu

İkinci giriş kümesi olan terminale uzaklık üyelik fonksiyonu Şekil 3.15.'teki gibidir.



Şekil 3.15. MATLAB programı terminale uzaklık üyelik fonksiyonu

Son olarak çıkış kümesi olan terminal adet üyelik fonksiyonu Şekil 3.16.'daki gibidir.



Şekil 3.16. MATLAB programı terminal adet üyelik fonksiyonu

Bu çalışmada toplam on altı adet kural oluşturulmuştur. Bu oluşturulan kurallara örnek verilirse terminaller arası uzaklık çok yakınsa ve terminale uzaklık çok yakınsa; çok az sayıda terminal kullan kuralı birinci kuraldır. Bu çalışmada MATLAB programı aracılığı ile toplam on altı adet bulanık kural Tablo 3.1.'deki gibi oluşturulmuştur.

Tablo 3.1 MATLAB bulanık kural tablosu

1. If (TerminallerArası is ÇokYakın) and (TerminaleUzaklık is ÇokYakın) then (TerminalAdet is ÇokAz)
2. If (TerminallerArası is ÇokYakın) and (TerminaleUzaklık is Yakın) then (TerminalAdet is Az)
3. If (TerminallerArası is ÇokYakın) and (TerminaleUzaklık is Uzak) then (TerminalAdet is Az)
4. If (TerminallerArası is ÇokYakın) and (TerminaleUzaklık is ÇokUzak) then (TerminalAdet is ÇokFazla)
5. If (TerminallerArası is Yakın) and (TerminaleUzaklık is ÇokYakın) then (TerminalAdet is ÇokAz)
6. If (TerminallerArası is Yakın) and (TerminaleUzaklık is Yakın) then (TerminalAdet is Az)
7. If (TerminallerArası is Yakın) and (TerminaleUzaklık is Uzak) then (TerminalAdet is Fazla)
8. If (TerminallerArası is Yakın) and (TerminaleUzaklık is ÇokUzak) then (TerminalAdet is ÇokFazla)
9. If (TerminallerArası is Uzak) and (TerminaleUzaklık is ÇokYakın) then (TerminalAdet is Az)
10. If (TerminallerArası is Uzak) and (TerminaleUzaklık is Yakın) then (TerminalAdet is Fazla)
11. If (TerminallerArası is Uzak) and (TerminaleUzaklık is Uzak) then (TerminalAdet is ÇokFazla)
12. If (TerminallerArası is Uzak) and (TerminaleUzaklık is ÇokUzak) then (TerminalAdet is ÇokFazla)
13. If (TerminallerArası is ÇokUzak) and (TerminaleUzaklık is ÇokYakın) then (TerminalAdet is Az)
15. If (TerminallerArası is ÇokUzak) and (TerminaleUzaklık is Uzak) then (TerminalAdet is ÇokFazla)
16. If (TerminallerArası is ÇokUzak) and (TerminaleUzaklık is ÇokUzak) then (TerminalAdet is ÇokFazla)

Matlab programında oluşturulan bulanık kurallar Tablo3.2.'deki gibidir.

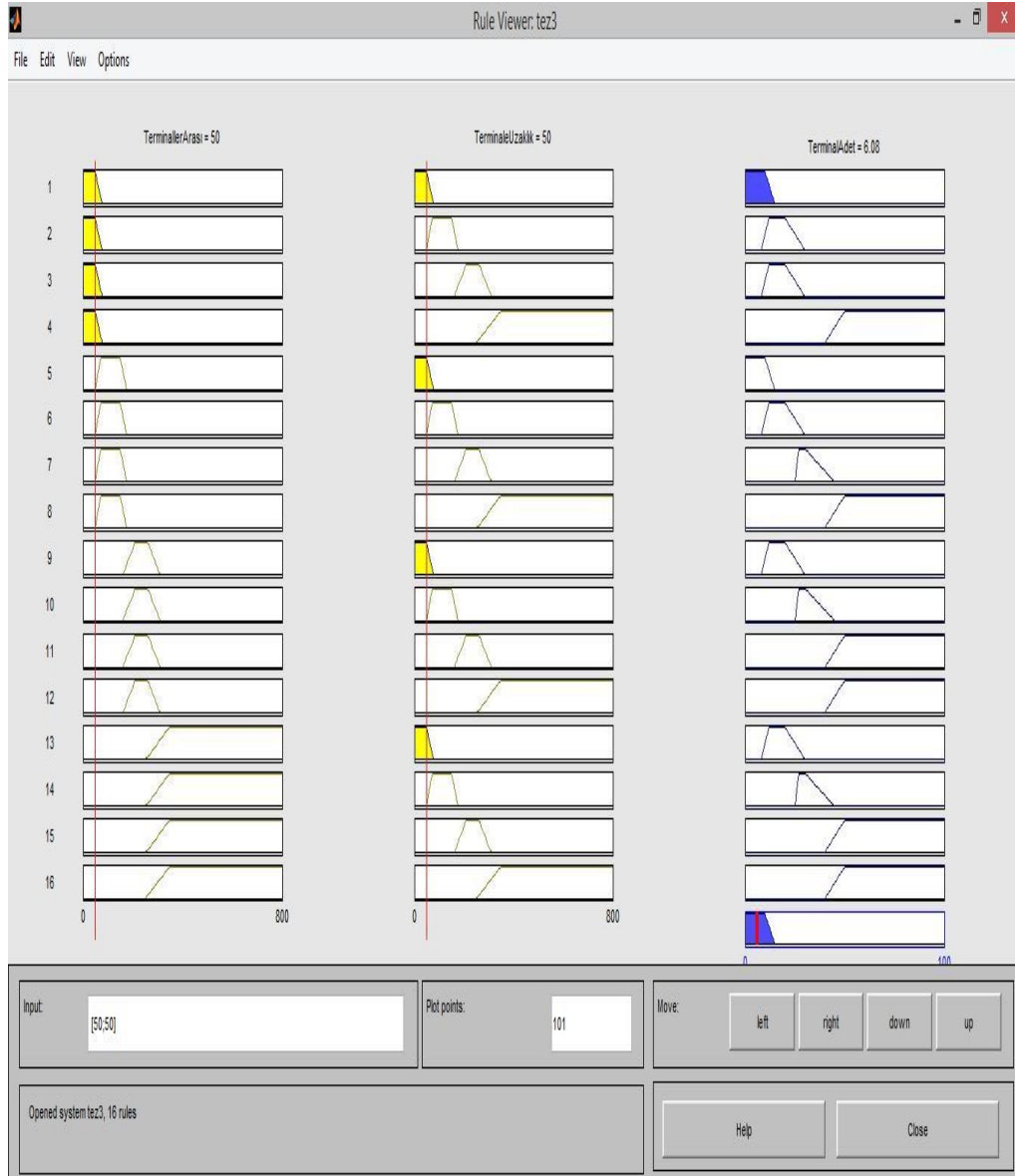
Tablo 3.2 Bulanık kurallar

	Terminaler Arası Uzaklık			
	Çok Yakın	Yakın	Uzak	Çok Uzak
Çok Yakın	Çok Az	Az	Az	Çok Fazla
Yakın	Çok Az	Az	Fazla	Çok Fazla
Uzak	Az	Fazla	Çok Fazla	Çok Fazla
Çok Uzak	Az	Fazla	Çok Fazla	Çok Fazla

3.3. Bulanık Kuralların Modellenmesi Ve Görüntülenmesi

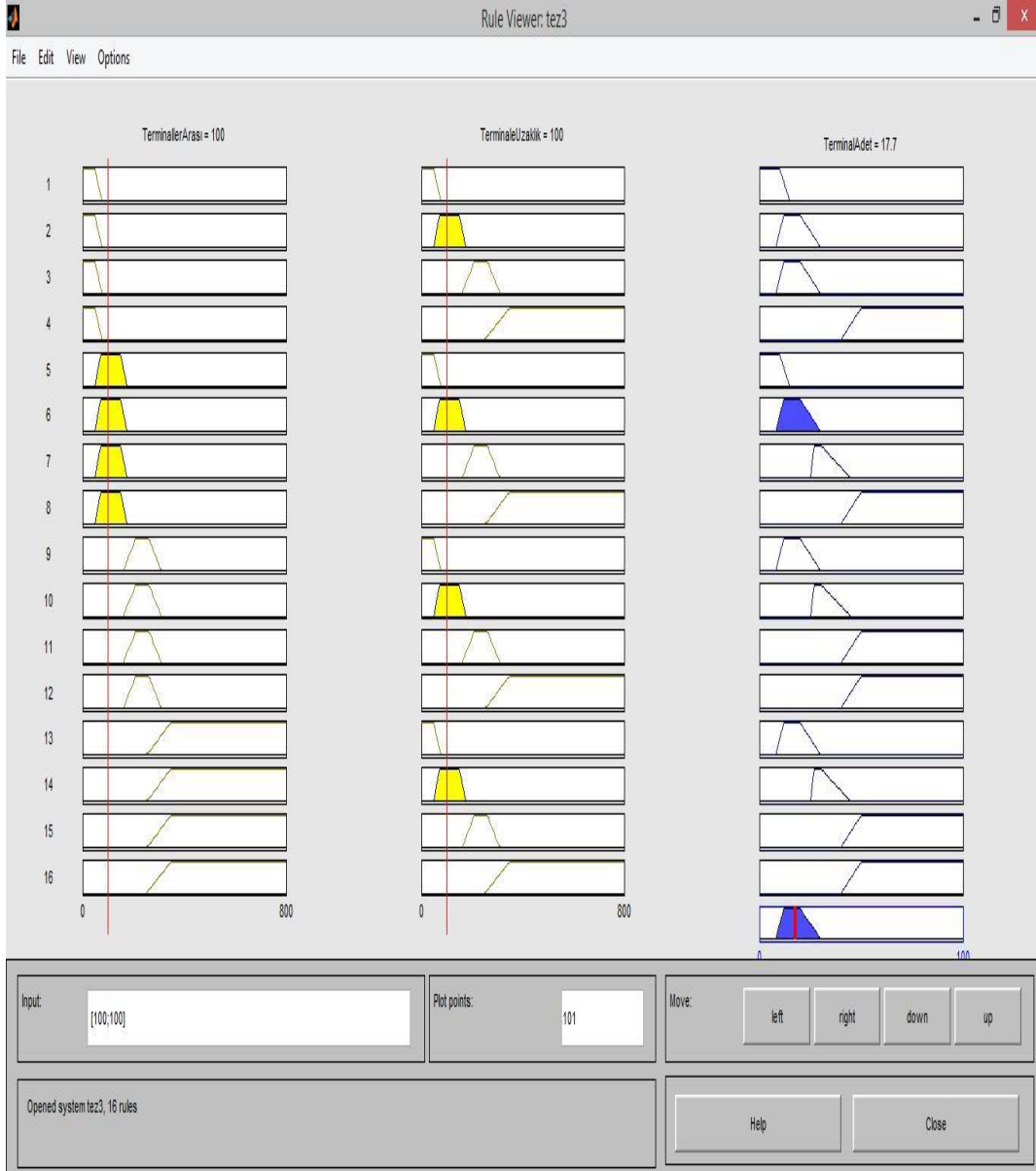
Çalışmanın bu bölümünde MATLAB programı kullanılarak modelleme sonuçları ekranları paylaşılmıştır.

Sonuç ekranı Şekil 3.17.'de görüldüğü gibi terminaler arası uzaklık 50 metre ve terminale olan uzaklık 50 metre olursa çıkış sonucu olan terminal sayısı değeri 7 olarak gözükmemektedir. Bu sistemde anlatılan işletmenin/tesis yüz ölçümü alanının küçük olduğu ve personelin çok yakın mesafelerde terminale ve terminaler arasına erişim sağladığı belirtilmektedir. Bu sonuçtaki bir sistem ile 350-400 personele sahip bir işletme/tesis, beş dakika içerisinde dijital sayımı gerçekleştirebilir. Ayrıca herhangi bir terminalin arızası durumunda terminaler arası mesafe çok yakın olduğu için, sayım işleminde herhangi bir aksama gerçekleşmeyecektir. Bu sonuçta diğer tüm sistemlerin de doğru olarak çalıştığı kabul edilmiştir.



Şekil 3.17. MATLAB programı Mamdani ve ağırlık merkezi durulaştırma metodu sonuç ekranı 1

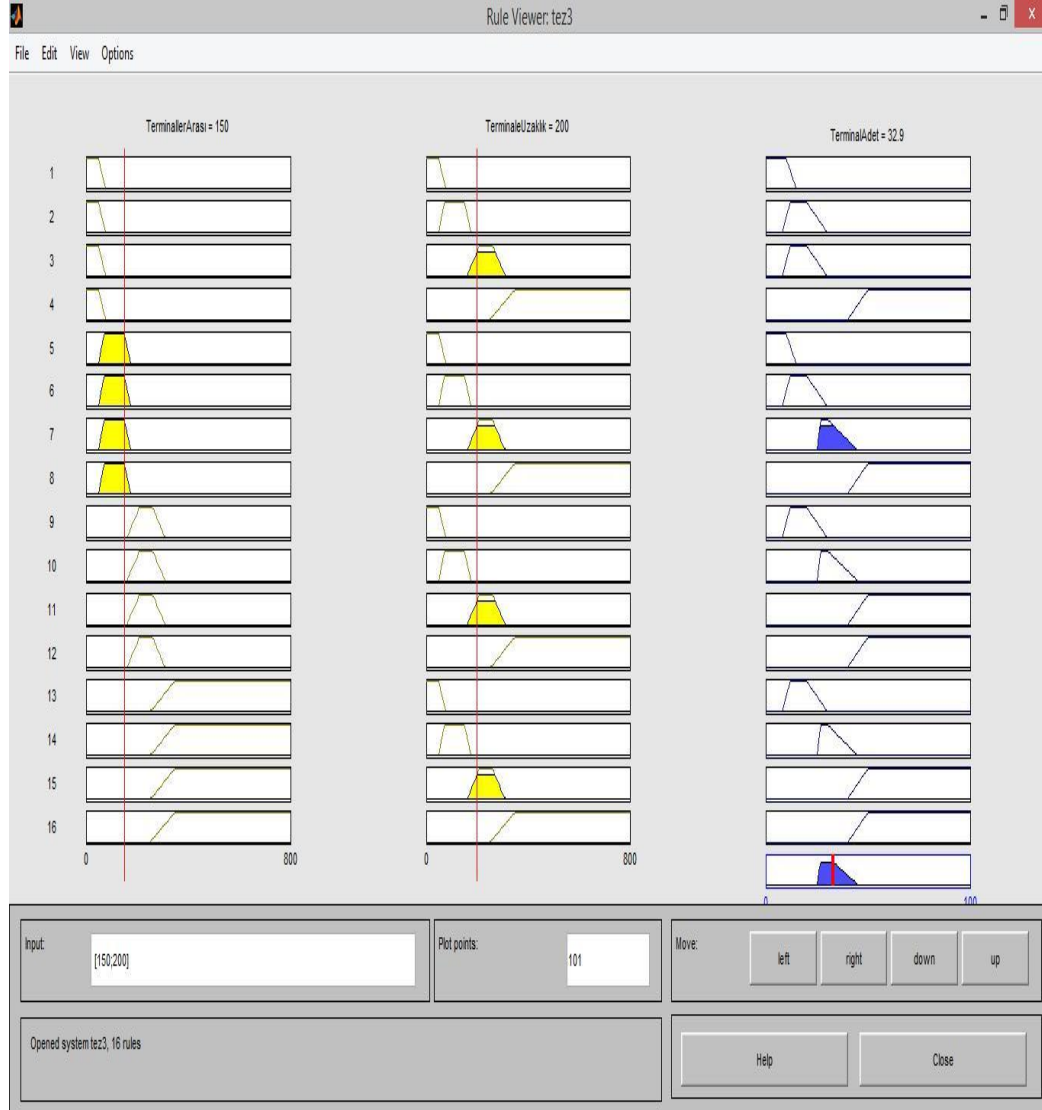
Sonuç ekranı Şekil 3.18.'de görüldüğü gibi terminaller arası uzaklık 100 metre ve terminale olan uzaklık 100 metre olursa çıkış sonucu olan terminal sayısı değeri 18 olarak gözükmemektedir. Bu sistemde anlatılan işletmenin/tesisin yüz ölçümü alanının orta ölçekte olduğu ve personelin yakın mesafelerde terminale ve terminaller arasında erişim sağladığı belirtilmektedir. Bu sonuçtaki bir sistem ile 900-1000 personele sahip bir işletme/tesis, beş dakika içerisinde dijital sayımı gerçekleştirebilir. Ayrıca herhangi bir terminalin arızası durumunda terminaller arası mesafe yakın olduğu için, sayım işleminde aksama gerçekleşmeyecektir. Bu sonuçta diğer tüm sistemlerin de doğru olarak çalıştığı kabul edilmiştir.



Şekil 3.18. MATLAB programı Mamdani ve ağırlık merkezi durulaştırma metodu sonuç ekranı 2

Sonuç ekranı Şekil 3.19.'da görüldüğü gibi terminaller arası uzaklık 150 metre ve terminale olan uzaklık 200 metre veya tam tersi olursa çıkış sonucu olan terminal sayısı değeri 33 olarak gözükmemektedir. Bu sistemde anlatılan işletmenin/tesisinin yüz ölçümü alanının büyük ölçekte olduğu ve personelin uzak mesafelerde terminale ve terminaller arasına erişim sağladığı belirtilmektedir. Bu sonuçtaki bir sistem ile 1700-1800 personele sahip bir işletme/tesis, beş dakika içerisinde dijital sayımı gerçekleştirebilir. Fakat herhangi bir terminalin arızası durumunda terminaller arası

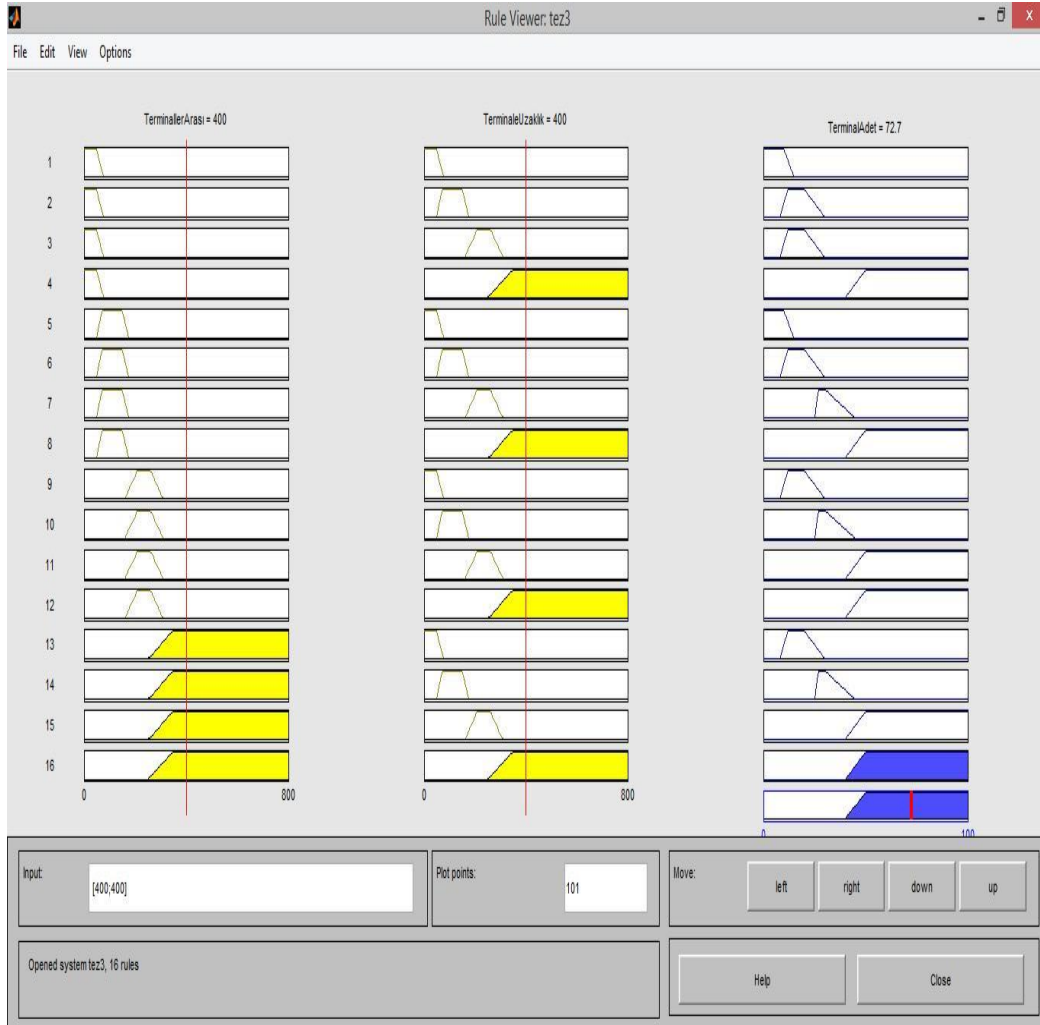
mesafe uzak olduğu için, sayım işleminde aksamalar gerçekleşecektir. Bu sonuçta diğer tüm sistemlerin de doğru olarak çalıştığı kabul edilmiştir.



Şekil 3.19 MATLAB programı Mamdani ve ağırlık merkezi durulaştırma metodu sonuç ekranı 3

Sonuç ekranı Şekil 3.20.'de görüldüğü gibi terminaller arası uzaklık 400 metre ve terminale olan uzaklık 400 metre veya tam tersi olursa çıkış sonucu olan terminal sayısı değeri 73 olarak gözükmemektedir. Bu sistemde anlatılan işletmenin/tesisin yüz ölçümü alanının çok büyük ölçekte olduğu ve personelin çok uzak mesafelerde terminale ve terminaller arasına erişim sağladığı belirtilmektedir. Bu sonuçtaki bir sistem ile 3900-4000 personele sahip bir işletme/tesis, beş dakika içerisinde dijital sayımı gerçekleştiremeyecektir. Ayrıca herhangi bir terminalin arızası durumunda

terminaller arası mesafe çok uzak olduğu için, sayım işleminde aksamalar gerçekleşecektir. Yapılması gereken yürüme mesafelerinin daha kısa olacak şekilde terminallerin yerlerinin düzenlenmesi gereklidir. Aksi takdirde yapılacak olan harcamalar boşa gidecektir. Bu sonuçta diğer tüm sistemlerin de doğru olarak çalıştığı kabul edilmiştir.



Şekil 3.20. MATLAB programı Mamdani ve ağırlık merkezi durulaştırma metodu sonuç ekranı 4

BÖLÜM 4. ARAŞTIRMA BULGULARI

4.1. Elde Edilen Bulgular Ve Genel Yorum

Bu çalışmada hazırlanmış olan sistem, dağınık grup haberleşme sistemlerinden açık grup haberleşme yöntemi ve merkezi yapılı sistem şeklinde çalışmaktadır. Dağınık grup haberleşme sistemlerinden olan açık grup haberleşme sayesinde, tesiste/işletmede çalışan personel herhangi bir acil durum oluşması sonucunda kendi bölgesine sayım için yetişmek zorunda değildir. Dolayısıyla, tesiste/işletmede çalışan personel daha sakin şekilde sayıma katılacak, acil durumda olmasına rağmen personele ek telaş oluşturulmayacaktır. Merkezi yapılı sistem sayesinde ise dağınık grup haberleşme sisteminin yönetimi kolaylaştırılmıştır. Yönetimi kolaylaştırılan sistem sayesinde, acil durum komuta merkezi, acil müdahale ekiplerini daha hızlı şekilde yönlendirebilecektir. Fakat merkezi yapılı sistemin dezavantajı olarak, sunucu ile sahada dijital sayımı gerçekleştiren cihazlar arası mesafe arttıkça sistemde gecikmeler yaşanabilir. Hatta bu durumda gönderilen ve alınan mesajlarda yaşanan gecikmeler dolayısıyla ileilmeme durumları ortaya çıkabilir. Bu durumu ortadan kaldırmak içinse merkezi yapılı sistemden hibrit yapılı sisteme geçilmesi gerekmektedir. Hibrit yapılı sistem daha karmaşık olduğu ve yönetim işlemi için ek personeller ve sunucular gerektirdiği kesindir.

Hazırlanan C# uygulaması sayesinde, sahadaki dijital sayım cihazlarından veri tabanına ulaşan bilgiler hızlı bir şekilde işlenmektedir. Yapılan sistem testlerinde yaklaşık 400 verinin 1 saniyeden az süre içerisinde işlenip sonuçlar listelerinin oluşturulduğu görülmektedir. Bu uygulamanın ortaya çıkarttığı sonuçlarda, eksikler listesi doğru olarak hesaplanabilmekte ve eğer acil durum oluşursa acil durum müdahale ekiplerinin aranacak personel hakkında net bilgilere sahip olmaktadır. Fakat hazırlanan uygulamada sadece eksikler listesini ve diğer listeleri

göstermektedir. Eksik personelin yeri sistem tarafından bildirilmemektedir. Tesiste/işletmede aranacak olan eksik personeller, acil durum müdahale ekiplerinin geçmişteki tecrübelerine dayanılarak kurtarılabilir. Dolayısıyla acil durum müdahale ekipleri sürekli tatbikatlar yaparak tecrübelerini arttırmalıdır. Diğer bir eksiklik ise, büyük ölçekli tesis/işletmelerde, hazırlanmış olan bu sistemin veri tabanı yeterli cevap veremeyecektir. Daha nitelikli veri tabanı ve bu veri tabanının düzenli bakımlarının yapılması gerekmektedir.

Bu çalışmada uygulanan bulanık mantık metodu ile sahada sayım için kullanılması gereken dijital sayım terminallerinin kaç adet olduğu, terminallerin birbirlerine olan uzaklığı ve personelin terminale olan uzaklığına bağlı olarak çözümlenmeye çalışılmıştır. MATLAB programında yapılan modelleme sayesinde dört farklı sonuç hakkında yorum yapılmıştır. Bulanık dilsel ifadeler kullanılarak çok yakın ve yakın olarak nitelendirilen dijital sayım terminallerinin dağılımı sayesinde sistem verimli olacak şekilde sonuç verecektir. Uzak ve çok uzak ifadelerde, yeterli zaman içerisinde sonuçlar alınamayacak olup, sistemin terminal sayısının artırılması, personele daha yakın yerlere yerleştirilmesi ve terminaller arası mesafenin kısaltılması gerekmektedir. Farklı bir senaryo olarak tatbikat veya acil durum sayım işleminin süresi uzatılabilir. Ama bu durumda gerçek hayatta müdahale süresi uzadığı can kayıpları yaşanabilir.

Özet olarak gerçek hayatta orta ve büyük ölçekli tesis veya işletmelerde yapılan acil durum tatbikatlarında aksaklıklar, sayım hataları, uzun sürede sayım sonuçlarının elde edilmesi, gerçek olmayan sonuçlarla karşılaşmaktadır. Bu çalışmada hazırlanan dijital sayım sistemi gibi sistemler, insan kontrolündeki sayım işlemini, dijitalleştirerek, kesin sonuçların elde edilmesinde faydalı olacaktır.

BÖLÜM 5. SONUÇ VE ÖNERİLER

Türkiye Cumhuriyeti Devleti'nin 2013 yılında kabul ettiği İşyerlerinde Acil Durumlar Hakkında Yönetmelik'e göre, işverenin her yıl düzenli olarak acil durum tatbikatı yaptırması zorunludur. Bu tatbikatların sonucunda, tesiste/işletmede bulunan kişilerin sayımı gerçekleştirilmektedir. Tesiste/işletmelerde yapılan bu tatbikatlarda sayım işlemi acil durum toplanma alanı başkanı tarafından yapılmaktadır. Bu sayımlarda gözle sayma metodu kullanılarak, kâğıt ve kalem aracılığı ile notlar alınmaktadır. Bu notlar acil durum komuta merkezi ile paylaşılmaktadır. Bu uygulanan klasik yöntemde birçok hatalar yapılmakta ve istenilen süre içerisinde sayımlar tamamlanamamaktadır.

Bu çalışmada, acil durum tatbikatları için dijital sayım sistemi tasarlanmıştır. Dijital sayım işlemi VIRDI marka RFT cihazlar kullanılarak yapılmıştır. Cihazların tesis ve işletmeye yerleştirilmesi yönteminde dağınık grup haberleşme yönteminden faydalanılmıştır. Yapılan benzetimlerde VIRDI marka RFT cihazların dağınık grup haberleşme sistemlerinden biri olan açık grup halinde haberleşme ve merkezi sistemli yönetim mekanizması doğru şekilde çalıştırılmış ve haberleşmeler eksiksiz sağlanabilmiştir. Fakat dijital sayım cihazlarının sayısı ve haberleşme mesafeleri arttıkça merkezi yapılı sistem terk edilmesi gerekmektedir. Merkezi yapılı sistem yerine hibrit yapılı yani ara sunuculara sahip yapıda dağınık grup haberleşme sistemi tercih edilmesi gerekmektedir. Bu durum daha karışık bir dağınık grup sisteminin oluşmasına sebep olacağı gibi, sistemin yönetilmesi için ilave maliyet ve çalışan gerekmektedir.

Sistem kurulumu için kullanılan VIRDI marka cihazlarda biyometrik sistem olan parmak izi okuma ve personel kapı giriş kartı okumaktadır. Çıkarılan diğer bir sonuç ise biyometrik sistemin kullanılması kişisel verilerin korunması problemini

çıkartabileceği gibi, aslında acil durumlarda oluşan kargaşa anında biyometrik parmak izi okutma işlemi zaten telaşlı olan çalışan için avantaj sağlayacaktır. VIRDI marka cihazların daha üst modelleri hem daha hızlı hem de daha farklı biyometrik (retina, yüz tanıma, ses tanıma) özelliklere sahipler. Kurulacak olan dijital sayım sistemin farklı parametreleri okuyabilme özelliğinin olması sayesinde, çalışanların personel kartını okutma mecburiyetlerini ortadan kaldıracaktır. Böylece çalışanlar yaşadıkları telaş esnasında biraz daha rahat davranabileceklerdir.

Bu çalışmada elde edilen diğer bir sonuç ise hazırlanmış olan C# uygulaması hakkındadır. Uygulama hızlı şekilde yaklaşık dört yüz veriyi tarama ve içinden gerekli olan bilgileri bulma işlemi bir saniye gibi kısa bir sürede içerisinde tamamlamıştır. VIRDI marka cihazların veri tabanı olan UNIS ile çalışmada oluşturulan listeler adındaki veri tabanı sistemi sağlıklı olarak kontrol etmektedir. Fakat bu iki veri tabanı da büyük verileri işleme hakkında nasıl davranacağı konusu belirsizdir. Ayrıca veri tabanlarının sürekli bakımları ve gereksiz verilerin temizlenme işlemi için teknik bir personel bulundurulması gerekmektedir.

Bu çalışmada paylaşılan son bulgular ise bulanık mantık hakkındadır. Bulanık mantık matlab programı ile benzetim yapılmıştır. Tesiste/işletmede personelin acil toplanma alanlarındaki dijital sayım cihazlarına olan uzaklıkları, dijital sayım cihazları arası uzaklık ve sahada konumlandırılacak olan dijital sayım cihaz sayısı benzetimi yapılmıştır. Önemli olan dijital sayım işleminin beş dakika içerisinde tamamlanması gerektiğidir. Her dijital sayım cihazına elli beş kişi düşecek şekilde yapılan benzetimler doğru sonuçlar çıkarmıştır. Tolerans olarak cihaz başına altmış beş kişi sayıma katılabilir. Fakat bu istisna sadece dilsel ifade ile çok yakın olarak tabir edilen cihazlarda mümkün olacaktır.

Matlab programı ile yapılan benzetimlerde, cihazlar arası uzaklık hesaplanmasının sebebi ise bir bölgedeki dijital sayım cihazının arıza yapma durumunun irdelenmesi içindir. Bu durumda arızalı cihazın olduğu bölgedeki çalışanlar iki gruba ayrılarak, acil toplanma bölgesine en yakın iki farklı bölgeye yönlendirmelerini gerektirir. Çalışmada adı geçen dilsel ifadelerden olan, çok yakın ve yakın tanımlamaları için

sistem doğru çalışacaktır. Fakat dilsel ifade olarak uzak ve çok uzak olarak geçen yapılandırmalarda sistem uygun çalışmayacaktır.

Matlab programı ile benzetimi yapılan diğer bir durum ise çalışanların dijital sayım cihazlarına olan uzaklıklarıdır. Bu kısımda unutulmaması gereken durum ise sayıma katılacak tüm çalışanların tek bir uzaklıktan acil durum toplanma bölgesine gitmeye çalışmamasıdır. Tesiste/işletmede çalışanlar, dağınık olarak veya küçük gruplar halinde farklı uzaklıklardan acil durum toplanma bölgelerine erişim sağlamaktadır.

Öneri olarak bu çalışmaya konu olan dijital sayım işlemi farklı metotlar ve farklı biyometrik cihazlar ile de çalışılabilir. Cihazların maliyeti arttıkça, farklı sayım yöntemlerine imkân verebilecektir. Gelecekte teknolojinin gelişmesiyle iki farklı dijital sayım konusu yapılabilir. Bunlardan birincisi olarak binaların çıkış kapılarının üzerlerine yerleştirilecek kameralar ile yüz tanıma sistemleri geliştirilebilir. Fakat yüz tanıma teknolojisi henüz yeterli seviyede değildir. İkinci konu ise binaların kapılarına yerleştirilecek olan RFT algılayıcılar sayesinde, çalışanlar binayı terk ettiği esnada sayım gerçekleştirilebilir. Bu sistemin benzeri ücretli karayollarında, araçların Hızlı Geçiş Sistemlerinde kullanılmaktadır.

KAYNAKLAR

- Adar, E. And Huberman, B. A. (2000) "Free Riding on Gnutella." Hewlett Packard, InformationDynamics Lab.
- Alvisi, L. and Marzullo, K. (1998) "Message Logging: Pessimistic, Optimistic, Causal, and Optimal." IEEE Trans. Softw. Eng., (24)2:149-159.
- Babaoglu, O. And Toueg, S. (1993) "Non-Blocking Atomic Commitment." In Mullender, S.(ed.), Distributed Systems, pp. 147-168. Wokingham: Addison-Wesley, 2nd ed.
- Bass, L., Clements, P., And Kazman, R. (2003) Software Architecture in Practice. Reading,MA: Addison-Wesley, 2nd-ed.
- Bernstein,P., Hadzilacos, V., and Goodman, N. (1987) Concurrency Control and Recovery in Database Systems. Reading, MA: Addison-Wesley.
- Bezdek, J.C. (1978) "Fuzzy partitions and relations and axiomatic basis for clustering". Fuzzy Sets and Systems. 1. pp. 111–127.
- Birrell, A. and NELSON, B. (1984) "Implementing Remote Procedure Calls." ACM Trans.CompoSyst., (2)1:39-59.
- Chervenak, A., Schuler, R., Kesselman, C., Kor..\.Nda,S., and Moe, B. (2005) "WideArea Data Replication for Scientific Collaborations." Proc. Sixth Int'l Workshop on GridComputing, (Seattle, WA). New York, NY: ACM Press.
- Cohen, B. (2003) "Incentives Build Robustness in Bittorrent." Proc. First Workshop allEconomics of Peer-to-Peer Systems, (Berkeley, CA).
- Cristian, F. (1991) "Understanding Fault-Tolerant Distributed Systems." Commun. ACM, (34)2:56-78.
- De Cock, M.; Bodenhofer, U.; Kerre, E. E. (2000). Modelling Linguistic Expressions Using Fuzzy Relations. Proceedings of the 6th International Conference on Soft Computing.
- Diao, Y., Hellerstein, J., Parekh, S., Griffith, R., Kaiser, G., And Phung, D. (2005) "A Control Theory Foundation for Self-Managing Computing Systems." IEEE J. Selected Areas Commun., (23)12: 2213-2222.
- Diot, C., Levine, B., Lyles, B., Kassem, H., And Balensiefen, D. (2000) "DeploymentIssues for the IP Multicast Service and Architecture." IEEE Network, (14)1:78-88.
- Dubois, D. and Prade, H. (1988) Fuzzy Sets and Systems. Academic Press, New York.

- El-Sayed. A. (2005) A New Approach for Centralized End-System multicast Protocol, Menoufiya University, Egypt.
- Eugster, P., Felber. P., Guerraoui, R., And Kerl\Larrec, A.-M. (2003) "The ManyFaces of Publish/Subscribe." *ACM Comput. Surv.*, (35)2:114-131.
- Foster, I. and Kesselman, C. (2003) *The Grid 2: Blueprintfor a New Computing Infrastructure*. San Mateo, CA: Morgan Kaufman, 2nd ed.
- Foster, I., Kesselman, C., Tsudik, G., and Tuecke, S. (1998) "A Security Architecturefor Computational Grids." *Proc. Fifth Con! Computer and Communications Security*. ACM.
- Fowler, R. (1985) *Decentralized Object Finding Using Forwarding Addresses*. Ph.D. Thesis, University of Washington, Seattle.
- Hadzilacos, V. and Toueg, S. (1993) "Fault-Tolerant Broadcasts and Related Problems." In Mullender, S. (ed.), *Distributed Systems*, pp. 97-145. Wokingham: Addison-Wesley, 2nded.
- Hellerstein, J. L., Diao, Y., Parekh, S., And Tilbury, D. M. (2004) *Feedback Control ofComputing Systems*. New York: John Wiley.
- Janic. M. (2005) *Multicast in Network and Application Layer*. Ph.d. Thesis, Delft University ofTechnology, The Netherlands.
- Kephart, J. O. And Chess, D. M. (2003) "The Vision of Autonomic Computing." *IEEE Computer*, (36)1:41-50.
- Klaaua, D. (1965) *Über einen Ansatz zur mehrwertigen Mengenlehre*. Monatsb. Deutsch. Akad. Wiss. Berlin.
- Klawonna, F. and Novák, V. (1996) The relation between inference andinterpolation in the framework of fuzzy systems, *Fuzzy SetsSyst.* 81 (1996) 331–354.
- Kopetz, H. and Verissimo, P. (1993) "Real Time and Dependability Concepts." In Mullender,S. (ed.), *Distributed Systems*, pp. 411-446. Wokingham: Addison-Wesley, 2nd ed.
- Kuzmin, V.B. (1982) "Building Group Decisions in Spaces of Strict and Fuzzy Binary Relations" , Moscow.
- Lamport, L. (1978) "Time, Clocks, and the Ordering of Events in a Distributed System." *Commun. ACA1*, (21)7:558-565.
- Laprie, J.-C. (1995) "Dependability - Its Attributes, Impairments and Means." In Randell, B.,Laprie, J.-c., Kopetz, H., and Littlewood, B. (eds.), *Predictably Dependable ComputingSystems*, pp. 3-24. Berlin: Springer-Verlag.
- Leff, A. And Rayfield, J. T. (2004) "Alternative Edge-server Architectures for EnterpriseJavaBeans Applications." *Proc. Middleware 2004*, vol. 3231 of Lect. Notes Compo Sc.,(Toronto, Canada). Berlin: Springer-Verlag.
- Liang, L.R., Xuena, S.L., Yi Lu, W., Mandal, V, Patacsil, D and Kumar, D. (2006) "FM-test: A Fuzzy-Set-Theory-Based Approach to Differential Gene Expression Data Analysis", *BMC Bioinformatics*, 7 (Suppl 4): S7.

- Lin, S.-D., Llan, Q., Chen, M. and Zhang, Z. (2004) "A Practical Distributed Mutual Exclusion Protocol in Dynamic Peer-to-Peer Systems." Proc. Third Int'l Workshop on Peer-to-Peer Systems, vol. 3279 of Lect. Notes Compo Sc., (La Jolla, CA). Berlin: Springer-Verlag.
- Mehta, N., Medvioovic, N., And Phadke, S. (2000) "Towards A Taxonomy Of SoftwareConnectors." Proc. 22nd Int'l Conj. on Software Engineering, (Limerick, Ireland). NewYork, NY: ACM Press.
- Moraga, C. (2012) An Essay on the Interpretability of MamdaniSystems, in: E. Trillas, P.P. Bonissone, L. Magdalena, J.Kacprzyk (Eds.), Comb. Exp. Theory SE-5, Springer Berlin Heidelberg, pp. 61–72.
- Nayate, A., Dahlin, M., And Iyengar, A. (2004) "Transparent Information Dissemination." Proc. Middleware, vol 3231 of Lect. Notes Compo Sc., (Toronto, Canada). Berlin:Springer-Verlag, 2004.
- Neumann, P. (1995) "Architectures and Formal Representations for Secure Systems." Technical Report, Computer Science Laboratory, SRI International, Menlo Park, CA.
- Novák, V., Perfilieva, I. and Močkoř, J. (1999) Mathematical principles of fuzzy logic Dodrecht: Kluwer Academic.
- Obraczka, K. (1998) "Multicast Transport Protocols: A Survey and Taxonomy." IEEE Commun.Mag., (36)1:94-102.
- OMG (2004b) "UML 2.0 Superstructure Specification." OMG Document ptc/04-10-02, ObjectManagement Group, Framingham, MA.
- Ozsu, T. And Valduriez, P. (1999) Principles of Distributed Database Systems. Upper SaddleRiver, NJ: Prentice Hall, 2nd ed.
- Pal, S.K. (2015) D.P. Mandal, Fuzzy Logic and Approximate Reasoning: An Overview, IETE J. Res. 37 548–560.
- Pelletier, F. J. (2000). "Review of Metamathematics of fuzzy logics", The bulletin of symbolic logic, Vol. 6, No. 3.
- Pfleeger, C. (2003) Security in Computing. Upper Saddle River, NJ: Prentice Hall, 3rd ed.
- Plummer, D. (1982) "Ethernet Address Resolution Protocol." RFC 826.
- Rabinovich, M. And Spatscheck, O. (2002) Web Caching and Replication. Reading, MA: Addison-Wesley.
- Ricart, G. and Agrawala, A. (1981) "An Optimal Algorithm for Mutual Exclusion in Computer Networks." Commun. ACM, (24)1:9-17.
- Salii, V.N. (1965). "Binary L-relations". Izv. Vysh. Uchebn. Zaved. Matematika (in Russian). 44 (1): 133–145.
- Saroiu, S., Gummadi, P. K., And Gribble, S. D. (2003) "Measuring and Analyzing the Characteristics of Napster and Gnutella Hosts." A CM Multimedia Syst., (9)2: 170-184.

- Schmidt, D., Stal, M., Rohnert, H., And Buschmann, F. (2000) Pattern-Oriented Software Architecture - Patterns for Concurrent and Networked Objects. New York: John Wiley.
- Schneider, U. und Kirchberger, H. (2007) Evakuierungsberechnungen bei Brandereignissen mittels Ingenieurmethoden, in: "Brandschutz Jahrbuch 2006/07 Arbeitssicherheit", Petzenkirchen
- Shaw, M. And Clements, P. (1997) "A Field Guide to Boxology: Preliminary Classification of Architectural Styles for Software Systems." Proc. 21st Int'l Compo Softw. & Appl. Conf.
- Tanenbaum, A.S. and Van Steen, M. (2006) "Distributed Systems: Principles and Paradigms" Amsterdam, 2nd ed.
- Wieringa, R. And De Jonge, W. (1995) "Object Identifiers, Keys, and Surrogates-Object Identifiers Revisited." Theory and Practice of Object Systems, (1)2:101-114.
- Wierman, M. J. (2010) "An Introduction to the Mathematics of Uncertainty: including Set Theory, Logic, Probability, Fuzzy Sets, Rough Sets, and Evidence Theory", Coreighton University.
- Yang, M., Zhang, Z., Li, X., And Dai, Y. (2005) "An Empirical Study of Free-Riding Behavior in the Maze P2P File-Sharing System." Proc. Fourth Int'l Workshop on Peer-to-Peer Systems, Lect. Notes Compo Sc., (Ithaca, NY). Berlin: Springer-Verlag.
- Zadeh, L. A. (1996) Fuzzy Sets, Fuzzy Logic, Fuzzy Systems, World Scientific Press.
- Zadeh, L.A. (1965). "Fuzzy sets". Information and Control. 8 (3): 338–353.
- Zogg, J.-M. (2002) "GPS Basics." Technical Report GPS-X-02007, UBlox.

ÖZGEÇMİŞ

Muhammed Fatih PEKŞEN, 1979'da Sakarya'da doğdu. İlkokulu Sakarya'da, orta ve lise eğitimini İstanbul'da tamamladı. 2004 yılında Yeditepe Üniversitesi Bilgisayar Mühendisliği Bölümü'nden mezun oldu. 2015 yılında Sakarya Üniversitesi ikinci öğretim yüksek lisans Otomotiv Mühendisliği Bölümü'nden mezun oldu.2017 yılında Sakarya Üniversitesi Yangın Güvenliği ve Yanma Bölümü'nde tezli yüksek lisansa başladı. 2005,2015 yılları arasında çeşitli sektörlerde ve işletmelerde çalıştı. 2015, 2018 yılları arasında Toyota Otomotiv Türkiye A.Ş fabrikasında çeşitli projelere bağlı olarak çalıştı. 2018 yılında Sakarya Üniversitesi'nde öğretim görevlisi olarak çalışmaya başladı.