

**ATÖLYE MALİYETİNİ AZALTMA VE
VERİMLİLİĞİNİ ARTIRMAYA YÖNELİK AKILLI
BİLGİ YÖNETİM SİSTEMİ TASARIMI**

**2012
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ**

Beyza YAMAN

**ATÖLYE MALİYETİNİ AZALTMA VE VERİMLİLİĞİNİ ARTIRMAYA
YÖNELİK AKILLI BİLGİ YÖNETİM SİSTEMİ TASARIMI**

Beyza YAMAN

**Karabük Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalında
Yüksek Lisans Tezi
Olarak Hazırlanmıştır**

**KARABÜK
Haziran 2012**

Beyza YAMAN tarafından hazırlanan “ATÖLYE MALİYETİNİ AZALTMA VE VERİMLİLİĞİNİ ARTIRMAYA YÖNELİK AKILLI BİLGİ YÖNETİM SİSTEMİ TASARIMI” başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. İlhami M. ORAK



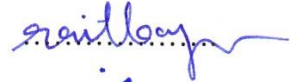
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 28/06/2012

Ünvanı, Adı SOYADI (Kurumu)

İmzası

Başkan : Doç. Dr. Raif BAYIR (KBÜ)



Üye : Yrd. Doç. Dr. İlhami M. ORAK (KBÜ)



Üye : Yrd. Doç. Dr. Salih GÖRGÜNOĞLU (KBÜ)

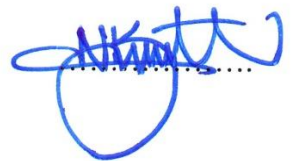


...../...../2012

KBÜ Fen Bilimleri Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Prof. Dr. Nizamettin KAHRAMAN

Fen Bilimleri Enstitüsü Müdürü



“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”

Beyza YAMAN

ÖZET

Yüksek Lisans Tezi

ATÖLYE MALİYETİNİ AZALTMA VE VERİMLİLİĞİNİ ARTIRMAYA YÖNELİK AKILLI BİLGİ YÖNETİM SİSTEMİ TASARIMI

Beyza YAMAN

Karabük Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:

Yrd. Doç. Dr. İlhami M. ORAK

Haziran 2012, 55 sayfa

Bu çalışmada küçük ve orta ölçekli makine atölyeleri için karar verme desteği sağlayan atölye yönetim sistemi tasarlanmıştır. Küçük ve orta ölçekli firmalar maliyet, hizmet kalitesi, hizmet süresi ve verimlilik konularında zorlu bir yarış içerisindeyler. Özellikle ürünleri gerçekleştirme ve teslim zamanları, çalışanların ve kaynakların verimli kullanımı, işletme içi işlerde görülen performans düşüklüğü sektörde görülen başlıca sorunlardır. Bu sebeple çalışma kapsamında gerekli bütün bilgileri toplayarak bir veritabanında kaydeden ve müşterilere hızlı ve doğru bilgi veren bir yazılım gerçekleştirilmiştir.

Bu çalışmada nesneye yönelik veritabanı kullanılmaktadır. Bu veritabanları nesneye yönelik programlama kapsamında inşa edilmektedirler ve bu çalışma için .NET tabanlı db4o veritabanı kullanılmaktadır. Biçimsel Modelleme Dili (Unified

Modelling Language - UML) ile veritabanı sistemi iyi bir şekilde tanımlanabildiği, görüntülenebildiği ve oluşturulabildiği için yazılımın geliştirilmesinde bu dil kullanılmıştır.

Sistemde belirsiz değer kullanımını desteklemek için, bulanık nesneye yönelik veritabanı kullanılmaktadır ki bu sayede kullanıcıya veritabanında kesin olmayan değerleri sorgulama imkânı verilmektedir. Önceki projelerden elde edilen maliyet gibi üretim parametrelerini analiz edebilmek için bu sistem geliştirilmiştir.

Bu çalışmada çok fazla kod yazmadan etkili bir nesneye yönelik veritabanının tasarlanabileceği görülmüştür. İlişkisel veritabanının tersine uygulama ve nesneye yönelik veritabanının her ikisinde aynı dil kullanılabilir. Bu yazılım ve veritabanı atölyelerin özelliklerine bağlı olarak ufak değişikliklerle farklı makine atölyelerinde kullanılabilir.

Anahtar Sözcükler : Atölye yönetim sistemi, nesneye yönelik veritabanı, UML, web tabanlı ara yüz, bulanık nesneye yönelik veritabanı.

Bilim Kodun : 902.1.014

ABSTRACT

M.Sc. Thesis

DESIGNING INTELLIGENT INFORMATION MANAGEMENT SYSTEM INTENDED FOR INCREASING EFFICIENCY AND DECREASING COST IN WORKSHOPS

Beyza YAMAN

Karabük University

Graduate School of Natural and Applied Sciences

Department of Computer Engineering

Thesis Advisor:

Assist. Prof. Dr. İlhami M. ORAK

June 2012, 55 pages

In this study, a workshop production management system supporting decision making is designed for use in small and medium-sized machine workshops. Small and medium sized companies' cost, maintenance quality, maintenance time and efficiency issues are in a tough competitive environment. Especially such as the products' production and delivery times, efficient usage of employees and resources, low performance level inside enterprises seem to be the major problems facing in this type of sector. Through this research it is aimed to design software to keep all the needed information in a suitable database and give the customers fast and accurate information.

In this project Object Oriented database is chosen for implementation. These databases can be constructed through object-oriented language and in the scope of

this work .NET based db4o is used. For software development, Unified Modeling Language (UML) is used since by using UML the database system is well defined, visualized, generated and specified considering complete project.

In order to support the uncertain values in the system, Fuzzy Object-Oriented Database is used such that it will enable the user to easily query the imprecise values in the database. The model is designed thus to analyze the production parameters such as cost from previous project parameters.

It is seen that without using so much coding load, an effective object-oriented database can be designed. It is possible to use the same language for both application and object-oriented databases in contrast to relational databases. The designed database can be used in different machine workshops with small modifications depending on the specifications of workshops.

Key Words : Workshop management system; object oriented database; UML; web based interface; fuzzy object oriented database.

Science Code : 902.1.014

TEŐEKKÜR

Bu tez alıőmasının gerekleőtirilmesinde desteęini esirgemeyen, bana anlayıőla yaklaőan; alıőmanın her aőamasında engin bilgi ve tecrübelerinden yararlandıęım; kiőilięi, deęerli fikirleri, yönlendirme ve bilgilendirmeleriyle bana her yönden bir rehber olan Sayın Yrd. Do. Dr. İlhami M. ORAK'a sonsuz teőekkürlerimi sunarım.

Beni hayatımın her döneminde destekleyen ve her zaman yanımda olan ailem ve arkadaşlarıma tüm kalbimle teőekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
KABUL.....	ii
ÖZET.....	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ.....	xi
ÇİZELGELER DİZİNİ	xii
SİMGELER VE KISALTMALAR DİZİNİ	xiii
BÖLÜM 1	1
GİRİŞ	1
BÖLÜM 2	5
NESNEYE YÖNELİK VERİTABANI YÖNETİM SİSTEMİ	5
2.1. NESNEYE YÖNELİK VERİ TABANLARININ GELİŞİMİ.....	5
2.2. NESNEYE YÖNELİK VERİTABANI KAVRAMLARI	5
2.2.1. Nesneye Yönelik Programlama Özellikleri.....	6
2.2.2. Veritabanı Özellikleri	7
2.3. NESNEYE YÖNELİK VERİTABANI MODELİ	8
2.3.1. NYVTYS Standartları	8
2.3.2. NYVTYS'nin Avantajları ve Dezavantajları	9
2.3.3. NYVTYS'nin Kullanım Alanları	11
2.3.4. Nesne Sorgu Dili	12
2.4. NESNEYE YÖNELİK VERİ TABANLARINDA KUSURLU BİLGİ GÖSTERİMİ	13
BÖLÜM 3	14
BULANIK NESNEYE YÖNELİK VERİTABANI YÖNETİM SİSTEMİ	14
3.1. BULANIK NESNEYE YÖNELİK VERİTABANI YAKLAŞIMLARI	14

	<u>Sayfa</u>
3.2. BENZERLİK TABANLI NESNEYE YÖNELİK VERİ TABANI SİSTEMİNİN MANTIKSAL TASARIMI	15
3.2.1. Nitelik Düzeyinde Hesaplamalar.....	16
3.2.2. Nesne ve Sınıf Düzeyinde Hesaplamalar	18
3.2.3. Sınıf ve Altsınıf Düzeyinde Hesaplamalar	22
BÖLÜM 4	25
ATÖLYE YÖNETİM SİSTEMİ.....	25
4.1. ATÖLYE YAPISI	25
4.2. SİSTEM ANALİZİ.....	27
4.3. SİSTEM MİMARİSİ	28
4.4. NESNEYE YÖNELİK VERİTABANI MODELİ	34
4.4.1. Kullanılan Veritabanı	34
4.4.2. UML Diyagramları.....	35
4.4.3. Veritabanı Sorguları	37
4.5. BULANIK NESNEYE YÖNELİK VERİTABANI MODELİ	39
4.5.1. Genişletilmiş UML Diyagramları	39
4.5.2. Bulanık Değerlerin Mantıksal Kullanımı	40
4.6. SİSTEM ARAYÜZÜ	43
BÖLÜM 5	46
DEĞERLENDİRME.....	46
KAYNAKLAR	50
EK AÇIKLAMALAR A. ÖRNEK MALİYET HESAPLAMA TABLOSU	53
ÖZGEÇMİŞ	55

ŞEKİLLER DİZİNİ

Sayfa

Şekil 3.1. Örnek üyelik fonksiyonu grafiği.	17
Şekil 3.2. Nitelik, sınıf ve alt sınıf UML diyagram gösterimi.....	24
Şekil 3.3. Bulanık kalıtım ilişkisini gösteren UML gösterimi	24
Şekil 4.1. Projenin temel modülleri.	29
Şekil 4.2. Projenin maliyet değerlendirme ve teklif verme aşamasını gösteren akış diyagramı.....	30
Şekil 4.3. Bulanık küme ile veri analizi aşamasını gösteren akış diyagramı.	31
Şekil 4.4. Veritabanının proje modülü.	32
Şekil 4.5. Boru imalat makinesinin ekipmanları.	33
Şekil 4.6. Boru imalat makinesinin maliyet şeması.	33
Şekil 4.7. Proje modülü.	35
Şekil 4.8. Proje modülü detayı.	36
Şekil 4.9. İnsan kaynakları modülü.	36
Şekil 4.10. Satış ve satın alma modülü.....	37
Şekil 4.11. Stok modülü.	37
Şekil 4.12. Bulanık nitelik içeren sınıflar.	40
Şekil 4.13. Ara yüzde bulanık özelliklere sahip parçaların sorgulanması.....	41
Şekil 4.14. Bir sorgunun çalıştırılması akış diyagramı.....	41
Şekil 4.15. Ara yüzde bulanık özelliklere sahip parçaların gösterilmesi	42
Şekil 4.16. Proje giriş sayfası.	44
Şekil 4.17. İncelemek üzere proje seçme.	44
Şekil 4.18. Proje detayı inceleme.	45

ÇİZELGELER DİZİNİ

	<u>Sayfa</u>
Çizelge 3.1. Boy niteliği için benzerlik matrisi.....	20
Çizelge 3.2. Hız niteliği için benzerlik matrisi.	23
Çizelge 3.3. Koku alma niteliği için benzerlik matrisi.....	23
Çizelge 4.1. Büyüklük için benzerlik matrisi.....	42
Çizelge 4.2. Süre için benzerlik matrisi.	43
Çizelge 4.3. İstasyon sayısı için benzerlik matrisi.	43

SİMGELER VE KISALTMALAR DİZİNİ

KISALTMALAR

ANSI	: Amerikan Ulusal Standartlar Enstitüsü (American National Standards Institute)
BNYVM	: Bulanık Nesneye Yönelik Veritabanı Modeli
BNYVT	: Bulanık Nesneye Yönelik Veritabanı
CAD	: Bilgisayar Destekli Tasarım (Computer Aided Design)
CAM	: Bilgisayar Destekli İmalat (Computer Aided Manufacturing)
CODASYL	: Conference on Data Systems Languages
ISO	: International Organization for Standardization (Standardizasyon için Uluslararası Organizasyon)
LINQ	: Language Integrated Query
NQ	: Native Queries
NYP	: Nesneye Yönelik Programlama
NYS	: Nesneye Yönelik Sistem
NYVM	: Nesneye Yönelik Veritabanı Modeli
NYVT	: Nesneye yönelik veri tabanı
NYVTYS	: Nesneye Yönelik Veritabanı Yönetim Sistemleri
ODMG	: Object Data Management Group
SODA	: Simple Object Database Access
SQL	: Yapısal Sorgu Dili (Structured Query Language)
OQL	: Nesne Sorgu Dili (Object Query Language)
QBE	: Query by Example
UML	: Biçimsel Modelleme Dili (Unified Modelling Language)
VTB	: Veritabanı Sistemi
VTYS	: Veritabanı Yönetim Sistemi

BÖLÜM 1

GİRİŞ

Küçük ve orta ölçekli firmalar maliyet, hizmet kalitesi, hizmet süresi ve verimlilik konularında zorlu bir yarış içerisindeyler. Özellikle ürünleri gerçekleştirme ve teslim zamanları, çalışanların ve kaynakların verimli kullanımı, işletme içi işlerde görülen performans düşüklüğü sektörde görülen başlıca sorunlardır.

Üretim sistemlerinin iş yoğunluğu ve karmaşıklığı, işletme içi faaliyetlerin koordinasyonu zorunluluğu, tüketici kitlesinin genişlemesi ve isteklerin değişik olması, malzeme, makine zamanı ve insan gücü kayıpları firmaların karşılaştığı sorunlardan bazılarıdır. Bu sorunlar sebebiyle tahmin edilemeyen iş süreçleri ve maliyetleri aynı zamanda teklif verme aşamasında da problem oluşturmaktadır. İşin maliyeti ve süresi iş yapılmadan önce tam olarak kestirilememektedir. Özellikle müşterilere gerçek teklife yakın bir maliyet hesabı çıkarılmak istendiğinde bu süreç üç veya dört hafta gibi bir zaman dilimini kapsamaktadır ve bu da oldukça uzun bir süredir. Firmaların karşılaştığı bu sorunları aşabilmek amacıyla bu çalışmada küçük ve orta büyüklükteki makine imalat atölyelerinde kullanılmak üzere bir atölye yönetim sistemi tasarlanmıştır.

Atölye işletmede üretimin yapıldığı merkezdir. Şirketler atölye kontrol kapsamında imalat sistemlerinde işlerin, yapılacağı zamanı ve sırasını kontrol etmekte; belirsizliğin oluşturduğu kapasite düşüşlerini ve dar boğazları ortadan kaldırarak üretim performanslarını artırmayı amaçlamaktadırlar [1].

Bununla birlikte imalat sistemlerindeki eğilim fiziksel olarak çalışan işçiler yerine bilgi tabanlı çalışan makineler, insanların ve aktivitelerin yönetimi yerine insanlar ve aktiviteler hakkındaki bilgilerin yönetimi şeklinde; yani otomasyon alanından akıllı şirket entegrasyonuna doğru bir değişim içerisindeyler [2]. Bilgi yönetimi yeni imalat

yönetim sistemlerinde merkeze geçtiğinden dolayı veritabanları da bu sistemler için hayati bir önem arz etmektedir.

İmalat sistemlerinin bilgisel bir yapıya kaymasıyla araç-gereç, ürün ve materyal akışını kapsayan imalat altyapısı üzerine, bilgi sistemi ve karar destek sistemleri eklenerek bilgisayar entegre edilmiş imalat sistemi tasarımına gidilmiştir. Bu sistemin merkezi ise şirketin imalat aktivitelerinin temel yönlerini birleştiren Veritabanı Yönetim Sistemi (VTYS) ve veritabanıdır - ki bu sayede veri alt sistemler içerisinde paylaşılabilir olabilmektedir. Bu tasarımın yapılması için tercih edilen programlama konsepti alanında revaçta olan nesneye yönelik programlamadır. Nesneye yönelik programlama değişik yapıların hiyerarşisini sağlamak açısından otomatik imalat atölyeleri için özel bir önem arz etmektedir. Çünkü sistemin bileşenleri, nesnelere ve görevleri açıkça ve iyi bir şekilde sunulabilmekte; nesneye yönelik programlama konsepti otomatik atölye çeşitlerine kolayca uygulanabilmektedir [2–3]. Bazı yazarlar nesneye yönelik veritabanı sistemlerinin, imalat ve mühendislik sistemlerinde artan kullanımına dikkat çekmektedirler [4–7]. Bu sebeple bu çalışmada nesneye yönelik veritabanı kullanılmıştır.

İlişkisel veritabanlarının da sınırlılıklarından dolayı 1980'lerin sonlarına doğru bilgisayar destekli dizayn gibi kompleks ve iç içe geçmiş uygulamaların ihtiyaçlarını karşılamak için nesneye yönelik veritabanı tasarımı kullanılmaya başlanmıştır [8].

Nesneye yönelik veritabanlarının (NYVT) aşağıdaki özellikleri [9], bu veritabanının seçilmesinde etkili olmuştur:

- a) İyi benzetim (simülasyon) yapılması,
- b) Kalıtım, birleşme gibi daha fazla ilişki sağlanması,
- c) Veri tabanı ve programlama için tek dil kullanılması,
- d) Daha çok ve karmaşık veri tipleri sağlanması,
- e) Nesnelere kalıcı olması,
- f) Kapsülleme, genişleme gibi diğer avantajları.

Diğer yandan sistemin öğrenilmesi kadar, edinilen bilgilerin sonucunda iyi bir tasarım ortaya koymak da önemlidir. Veritabanı modelini daha iyi sunmak ve görebilmek amacıyla UML diyagramları kullanılmıştır. Çalışmalarda UML diyagramlar sayesinde veritabanı modeli daha iyi tanımlanmakta, görüntülenmekte ve oluşturulmaktadır [10].

Veriye dayalı yorumlama özelliği; sisteme yapay zeka kısmının entegre edilmesi ile sağlanmıştır. Sistemde kayıtlı veriler kullanılarak yapay zekâ modülü sayesinde önceden kaydedilen projelerle istenen karakteristiklerin karşılaştırılması sağlanmıştır. Yapay zekâ, insanın düşünme yapısını anlayarak buna benzer bir davranışta bulunan, ya da düşünebilen bilgisayar sistemlerinin geliştirilmesi olarak tanımlanabilir. Kısaca, programlanmış bir bilgisayarın düşünme girişimidir. Daha geniş bir tanıma göre ise, yapay zekâ, bilgi edinme, algılama, görme, düşünme ve karar verme gibi insan zekâsına özgü kapasitelerle donatılmış bilgisayarlardır [11]. Yapay zekâ sistemi sadece makinenin öğrenmiş olduğu bilgiler ve parametreler ışığında sonuç üretebilme özelliğindedir. Dolayısıyla bu modül yeni gelen müşterilerin taleplerini değerlendirerek eğer sistemde eskiden aynı çapta yapılan bir proje varsa bu projelerin maliyetlerini göz önünde bulundurarak maliyet hesabı yapmakta ve hızlı bir şekilde müşteriye teklif vermektedir. Ayrıca yapılan bu teklif yine veritabanına kaydedilmektedir.

Yapay zekâ tekniği olarak 1965'te Prof. L.A. Zadeh tarafından ortaya konan bulanık kümeler teorisi baz alınarak sunulan bulanık nesneye yönelik veritabanı kullanılmıştır. Bulanık kümeler teorisinin amacı kesinlik içermeyen, belirsiz bilgileri işlemektir. Günlük konuşma dilinde hemen her konudaki ifadeler bulanıklık içerdiğinden dolayı bulanık mantık ve bulanık küme teorisi kesinlik içermeyen bilgilerin gösterimi için önemli bir boşluğu doldurmuştur [12]. Klasik veri tabanları ve veri modelleri de belirsiz ifadeleri gerçekleştirmekte yetersiz kalmaktadır. Özellikle, kesin ve tam olmayan bilgilerin veritabanına kaydedilmesi klasik veritabanlarıyla pek mümkün olmamaktadır. Bu sorunu çözmek amacıyla bulanık veri tabanları geliştirilmiş ve kullanıma sunulmuştur.

Mevcut veritabanında bulanıklık yeni projelerin oluşturulması sırasında karşılaştırılıp kaydedilmesi amacıyla kullanılmıştır. Özellikle projenin teklif verme aşamasında her zaman bir projenin aynısını bulmak mümkün olmayacaktır. Nesneye dayalı veritabanı ve bulanık mantık kavramlarının en iyi örtüştüğü uygulama Bulanık Nesneye Yönelik Veritabanı (BNYVT) yapısıdır. Çeşitli çalışmalar Bilgisayar Destekli Tasarım (Computer Aided Design - CAD) ve Bilgisayar Destekli İmalat (Computer Aided Manufacturing - CAM) uygulamalarında kompleks ve kesin olmayan bilgilerin yönetilmesinde bulanık nesneye yönelik veritabanı üzerinde yoğunlaştığından bahsetmektedir [13–14]. Çalışma kapsamında projeye yakın olan diğer projeleri inceleyebilmek için bulanık nesneye yönelik veritabanı kullanılmıştır.

Bu tez çalışmasında, ilk bölümde çalışma yapılan konu ile ilgili literatür çalışmalarına yer verilmiş ve konuya temel bir giriş yapılmıştır. İkinci Bölüm Nesneye Yönelik Veritabanları hakkında genel bilgi vermektedir. Üçüncü bölümde Bulanık Nesneye Yönelik Veritabanları açıklanarak çalışmayla ilgili referans olacak bilgiler yer almaktadır. Dördüncü bölüm yapılan çalışmanın detaylı olarak anlatılıp detaylandırıldığı kısımdır. Son bölümde ise yapılan çalışmayla ilgili sonuç ve çıkarılan yorumlara yer verilmektedir.

BÖLÜM 2

NESNEYE YÖNELİK VERİTABANI YÖNETİM SİSTEMİ

2.1. NESNEYE YÖNELİK VERİ TABANLARININ GELİŞİMİ

Verilerin dosya yapılarında saklanmasına 1950'lerde başlanmış olmasına rağmen "Veri Tabanı" terimi ilk kez 1960'larda askeri bilgi sistemlerinde kullanılmaya başlanmıştır. 1960'larda IBM firması tarafından "Hiyerarşik" ve sonrasında Conference on Data Systems Languages (CODASYL) adı verilen birlik tarafından sunulan "Ağ" veritabanı modelleriyle birlikte "Veritabanı Yönetim Sistemi" tanımı netlik kazanmıştır. 1970'lere gelindiğinde ise İlişkisel ve Varlık-İlişki modelleri sunularak İlişkisel Veritabanı Yönetim Sistemlerinin temelleri atılmıştır. Her bir veritabanının zayıf noktaları görülerek farklı bir model geliştirilmesi hedeflenmiştir. Diğer taraftan bu kayıt odaklı ve basit veri tiplerini destekleyen veritabanlarının yanında 1980'lerde daha karışık verileri destekleyebilen nesne odaklı Nesneye Yönelik Veritabanı Yönetim Sistemleri (NYVTYS) yaygınlaşmaya başlamıştır. Veri tabanı yönetim sistemleri uygulamasında klasik veri modelleri için yapılan en önemli eleştiri, veri kümelerinin teoride bütünleştirilebilmesi, ancak pratikte yetersizliklerin ortaya çıkmasıdır [9]. Ayrıca gerçek nesnelerin ilişkisel modele haritalanması, veri tiplerinin ve veriye erişimin sınırlı olması gibi nedenler bu veritabanlarının gelişmesinde önemli rol oynamıştır. Bu bölümde NYVTYS ile ilgili bilgilere yer verilecektir.

2.2. NESNEYE YÖNELİK VERİTABANI KAVRAMLARI

1989'da farklı enstitülerden bir grup akademik araştırmacı, E.F. Code'un 1970'lerde ilişkisel veritabanları için on iki kural tanımladığı gibi NYVTYS için bir grup

kurallar belirlemişlerdir. Bu kurallarda NYVTYS'lerinin temel olarak iki kriteri sağlaması gerektiği tanımlanmıştır [5]:

- a) Veri Tabanı Yönetim Sistemi olmalıdır.
- b) Nesneye Yönelik Sistem olmalıdır.

Bu temel özellikler aşağıda açıklanmıştır:

2.2.1. Nesneye Yönelik Programlama Özellikleri

- a) *Kompleks Nesnelere (Complex objects)*: NYVTYS, basit veri tiplerinin yanında başka nesnelere veya liste, dizi gibi veri tiplerini içerebilmektedir. Kompleks nesne yapısı geçişli bir yapıya izin verdiğinden performans artırımına ve ayrıca herhangi bir düzeyde nesneye ulaşmaya imkân sağlamaktadır.
- b) *Nesne Kimliği (Object identity)*: Her nesneyi bir diğerinden ayıran ve var olduğu sürece nesneyi niteleyen bir kimlik numarasıdır. Her nesne bir nesne kimliğine sahiptir ve bu nesne kimliği üzerinden ulaşım veya güncelleme yapılabilir.
- c) *Kapsülleme (Encapsulation)*: Kapsülleme en geniş manasıyla veri ve işlemlerin saklanması ve onlara metotlar yoluyla ulaşılmasını sağlayan bir yöntemdir. Bu da veri bağımsızlığını sağlayarak diğer metotların değişmesine gerek kalmadan sınıf uygulamalarının değişebilmesine imkân sağlamaktadır [15].
- d) *Tipler ve Sınıflar (Types and Classes)*: Tip, bir Nesneye Yönelik Sistemde (NYS) aynı karakteristiklere sahip nesne grubunun ortak özelliklerini özetler. Sınıf ise hemen hemen aynı olmakla birlikte aynı nitelik ve metotlara sahip nesne koleksiyonuna işaret eder. Her ikisi de benzer nesnelere gruplandırmak için kullanılır fakat bir sistem için ikisinden birini seçmek normal olanıdır [16].
- e) *Tip veya Sınıf Hiyerarşileri (Class or Type Hierarchies)*: Bir alt sınıf veya alt tip; üst sınıf veya üst tipten gelen nitelikleri kalıtım yoluyla alır [16].

- f) *Aşırı yükleme, Geçersiz kılma ve Geç bağlanma (Overloading, Overriding and Late binding)*: Aşırı yükleme tek bir metodun birkaç defa ve birkaç şekilde; genellikle farklı parametrelerle yeniden gerçekleştirilmesini sağlar. Geçersiz kılma, değişen tipler için işlemlerin yeniden tanımlanmasını sağlar. Geç bağlanma ise bu iki NYVT prensibini tamamlamak için, hangi nesnenin istendiğini kararlaştırır ve çalışma zamanında işlev görür [15].
- g) *Hesapsal Bütünlük (Computational completeness)*: NYS, Yapısal Sorgu Dilinin (Structured Query Language – SQL) yakalayamadığı bütünlüğü yakalamak amacıyla, program ve veritabanı arasındaki ilişkiyi iyi bir şekilde sağlamak, farkı en aza indirmek ve kompleks nesnelere başa çıkabilmek için güçlü bir yaklaşıma sahip olmalıdır.
- h) *Genişletilebilirlik (Extensibility)*: Veritabanında, sistem tarafından tanımlanmış sabit tipler vardır. Bunun yanında genişletilebilirlik özelliği hiçbir ayırım olmadan kullanıcı tanımlı tipler oluşturulmasına izin vermektedir. Tip genişletilebilirliği programlama diline kullanıcı tanımlı yeni tipler eklenmesini sağladığı ve ayrıca kullanımı kolay olduğu için programlama dillerinde güzel bir fırsattır.

2.2.2. Veritabanı Özellikleri

- a) *Süreklilik (Persistence)*: Süreklilik, bir nesnenin başka bir işlemde yeniden kullanılabilmesi için oluşturulduktan sonra geçerliliğini korumasıdır.
- b) *İkincil Saklama Yönetimi (Secondary storage management)*: İkincil bellek yönetimi Veritabanı Sistemlerinin (VTS) temel özelliklerinden birisidir. İndeks yönetimi, sorgu optimizasyonu gibi bazı mekanizmalar tarafından desteklenirler. Performansı etkilemekle birlikte kullanıcıya görünür değildir [5].

- c) *Eşzamanlılık (Concurrency)*: Klasik VTS'nin sağladığı eşzamanlılık mekanizmasını sağlamalıdır. Ayrıca kullanıcılar arasındaki uyum ve bütünlüğü de sağlamalıdır.
- d) *Kurtarma (Recovery)*: Yazılım veya donanım hatalarına karşı sistem klasik VTS'lerinde olduğu gibi kendini kurtarabilmelidir.
- e) *Özel Amaçlı Sorgu İmkânı (Ad Hoc Query Facility)*: VTS yüksek düzeyli, etkili, uygulamadan bağımsız bir sorgu özelliğine sahip olmalıdır. Bu sorgu dili olmak zorunda değildir, yerine bir çeşit grafik ara yüzü de olabilir [5].

2.3. NESNEYE YÖNELİK VERİTABANI MODELİ

2.3.1. NYVTYS Standartları

NYVTYS'lerini tanımlamak için 1989, 1990 ve 1995 yıllarında sırasıyla üç tane bildiri yayınlanmıştır [5,17–18]. Bu üç bildiri farklı enstitüler tarafından yazıldığından standartlar birbirinin devamı şeklinde değildir ve NYVTYS'leri için değişik tanımlamalardan oluşmaktadır. Bu arada 1991 tarihinde pazardaki önde gelen şirketler tarafından belirli standartlar geliştirmek amacıyla Object Data Management Group (ODMG) kurulmuştur.

İlişkisel model çerçevesi standartlarla belirlenmiş bir model olmasına rağmen; ODMG standartlarında nesneye yönelik veri modeli (NYVM), nesneye yönelik programlama mantığında inşa edilir. Diğer yandan bir sorgu dili desteklenmeyip, dizayn programlama diline bağlı olarak değişir. Bu grup Nesne Sorgu Dili (Object Query Language - OQL) gibi standartlar oluşturmasına rağmen SQL'in tersine OQL; Amerikan Ulusal Standartlar Enstitüsü (American National Standards Institute - ANSI), ve Standardizasyon için Uluslararası Organizasyon (International Organization for Standardization - ISO) gibi standartlar tarafından kabul edilmemiştir.

En yeni ODMG standardı ODMG 3.0'dır. ODMG 3.0 veritabanı, nesne ve programlama dilleri üzerine bina edilerek uygulama taşınabilirliğini ve veri kaydını sağlamayı amaçlamıştır. Bu sayede programcılar herhangi bir programlama dilini kullanarak verilerini kaydetme imkânına sahiptirler.

2.3.2. NYVTYS'nin Avantajları ve Dezavantajları

NYVTYS'nin avantajlarının yanında birçok dezavantajı da vardır. Bunlar kısaca şu şekilde açıklanabilir [16,19–20]:

NYVTYS'nin Avantajları:

- a) *Zengin tip sistemi sağlar.* Veritabanında bulunan sistem tanımlı tiplerin dışında kullanıcı kendisi de veri tipi tanımlayabildiği için NYVTYS zengin tip sistemini desteklemektedir.
- b) *Daha az programlama çabası gerektirir.* Genişletilebilirlik, yeniden kullanım, kalıtım gibi Nesneye Yönelik Programlama (NYP) özellikleri sayesinde kullanıcı daha az çabayla daha çok yol kat etmiş olur.
- c) *Kompleks nesne ve ilişkileri modellemede daha iyidir.* NYVTYS'leri kompleks nesnelere kolayca modelleyebilmesi için NYP yapısına göre optimize edilmiştir. Bu sebeple nesnelere arasında dolaşması ve yüksek miktardaki kompleks veri koleksiyonlarıyla baş etmesi kolaydır.
- d) *Empedans Uyumsuzluğu (Impedance Mismatch) yoktur.* NYP ve İlişkisel Veritabanı beraber kullanılırken ortaya çıkan mantıksal ve teknik zorlukların toplamına empedans uyumsuzluğu denir. NYVTYS, kullanıcıya herhangi bir haritalama yapmadan verileri değişik veri yapılarına kaydetmeye izin verir. İlişkisel Veritabanında ise nesnelere tablolarla belirlenen modele göre haritalamak gerekmektedir. NYVTYS'lerinde empedans uyumsuzluğu olmaması özellikle kompleks veri açısından performans avantajı sağlamaktadır.

Çünkü ilişkisel veritabanında her işlem yapılmak istenildiğinde haritalama işlemi gerekmekte ve bu da performansı düşürmektedir.

- e) *Birincil anahtar kullanımı yoktur.* Nesnelere erişim ve sorgular her nesneye atanan bir nesne tanımlaması (Object ID) ile sağlandığından dolayı birincil anahtar veya yabancı anahtar gibi tanımlamalara ihtiyaç yoktur.
- f) *Varlıkların gösteriminde kolaylık sağlar.* Gerçek dünya veri modellemesine daha yakın olduğu için nesnelerin gösterimi daha anlaşılır bir yapıdadır.
- g) *Sınıf hiyerarşisi sağlar.* Alt sınıflarla kod paylaşımı, kodların tek bir kere tanımlanması ve ortak kullanım gibi özellikler kullanıcıya kolaylık sağlamaktadır.

NYVTYS'nin Dezavantajları:

- a) *Kabul edilmiş standart yoktur.* ODMG tarafından belirlenmiş bazı standartlar olmasına rağmen, ürünler tarafından tamamı uygulanmamaktadır. Örneğin OQL evrensel olarak kabul edilmiştir fakat farklı ürünlerde farklı sorgu dilleri kullanılmaktadır.
- b) *NYVTYS matematiksel altyapıya dayanmamaktadır.* İlişkisel veritabanının ilişkisel cebir ve bir grup teoriye dayanmasına karşılık NYV için böyle bir alt yapı söz konusu değildir.
- c) *Özel amaçlı sorgu eksikliği vardır.* Özel amaçlı sorgular özellikle son kullanıcıya hitap eden bir ara yüz sağlayan, karışık sorguların kolayca döndürülmesine izin veren bir eklentidir. SQL bu tip sorguları iyi bir şekilde desteklemesine rağmen NYV için henüz bu özellik tam anlamıyla söz konusu değildir.
- d) *Programlama dili bağımlılığı vardır.* Veritabanının desteklediği dillere göre kullanabileceği seçenekler sınırlıdır. Ortak bir programlama dili yoktur.

- e) *Evensel kabul edilmiş bir veri modeli yoktur.* Farklı programlama dillerinin kullanılması da model farklılıklarını daha görünür kılmaktadır.
- f) *NYVTYS şirketleri daha az duyulmuşlardır.* Çünkü ilişkisel veritabanı şirketleri kadar büyük değillerdir.
- g) *Programcılar, NYVTYS veya kullandığı teknoloji hakkında çok fazla bilgiye sahip değillerdir.* İnsanların bildikleri veritabanını kullanması daha kolaydır.

2.3.3. NYVTYS'nin Kullanım Alanları

Günümüzde kullanılmakta olan çok iyi ilişkisel veritabanı örnekleri vardır. Bununla birlikte NYVT'nin ilişkisel veritabanına göre çok daha uygun olduğu bazı uygulamalar da mevcuttur. Günümüzde NYVT'nin bazı kullanıldığı alanlar şöyledir:

- a) Finans,
- b) Devlet,
- c) Savunma,
- d) Sağlık,
- e) Teknoloji,
- f) Ulaşım,
- g) İletişim,
- h) Medya.

BMW, Boeing, Bosch, IBM, Intel, Ricoh ve Seagate gibi dünya markaları farklı uygulamalarda NYVT tercih etmektedirler [21]:

- a) Boeing savunma alanında P-8A Poseidon savaş uçağı modelinde program geliştiricilerinin verimliliği artarken organizasyonel maliyetleri düşürdüklerini beyan etmişlerdir.
- b) BOSCH Sigpack Systems AG, bir paketleme teknolojisi şirketi, Delta XR31 adlı otomatik paketleme robot sistemlerinde kullanmış ve NYVT'nin güvenli

olduğunu, kolay kullanıldığını ve yüksek sayıdaki verilerle çalışırken performansın çok iyi olduğunu belirtmişlerdir.

c) Seagate Mirra Sync and Share Personal Server mimarisinde kullanmışlar ve çok iyi performanslara ulaştıklarını ve müşterilerinin ürünü daha etkili kullandıklarını beyan etmişlerdir.

d) BMW gömülü araç kontrol elektroniklerinde kullanmış ve bu sayede kısa süreli başlangıç zamanlarına ulaştıklarını belirtmişlerdir.

NYVT kullanan firmalar ortak olarak performans artışından memnun olduklarını, bu tip bir veritabanının gömülü sistemlerde çok iyi sonuçlar verdiğini, zaman ve maliyet açısından kazançlı olduğunu, modelin ve kullanımın ilişkisel veritabanına göre daha kolay olduğunu belirtmişlerdir.

2.3.4. Nesne Sorgu Dili

ODMG Standartları, SQL’le benzer bazı özellikleri sağlayan ve SQL kadar yaygın kullanımlı bir sorgu dili oluşturabilmek için OQL standardını sunmuştur. OQL standartlarında İstanbul’da yaşayan işçileri bulmak için en basit sorgu “SELECT e FROM İsci e WHERE e.adres.sehir = " İstanbul ";" şeklinde yazılır.

Bu sorgu başlangıçta SQL sorgusu gibi görünse de aslında OQL’dir ve bazı farklılıklar göstermektedir. Birincisi sorguya cevap nesne olarak döndürülmektedir. İkinci olarak “where” deyiminin içeriği nesne referanslarını içeren bir yapıda nesnelere yönlendirme yapar.

OQL, “Insert” ve “Update” gibi tanımlamalara sahip değildir. SQL’in tersine sadece verileri çekmek için kullanılır. Nesnelere ilgili işlemler uygun metotlar çağırılarak yapılmaktadır.

2.4. NESNEYE YÖNELİK VERİ TABANLARINDA KUSURLU BİLGİ GÖSTERİMİ

NYVTYS geleneksel olarak bilinen ve kesin değerleri içeren belirli bir evren olarak modellenmektedir. Fakat gerçek dünya değerleri genellikle bulanık anlamlarla hatta eksik bilgilerle doludur. Eğer kaydedilen verinin bir parçası öznel değerlendirmeler veya yargılar gerektiriyorsa bulanık veri mantığı ortaya çıkmaktadır. Önemli bilgilerin kaybı sonucunu doğurabileceği için bu tip veriler göz ardı edilemez ya da kesin değerlere uyması beklenemez. Gerçek dünya verilerini, özelliklerini kaybetmeyecek şekilde gösteren veri modellerine ihtiyaç duyulmaktadır [22].

Veritabanlarında kusurlu verileri göstermenin çeşitli yolları vardır. Fakat bunların en temel üç tanesi; kesin olmama, bulanık olma ve belirsizlik şu şekilde açıklanabilir [23–24]:

Kesin olmama (Imprecision): Eğer bir bilgi muhtemel değerlerden oluşan bir grubu işaret ediyorsa ve gerçek değer bu bir grup değerden biriye bu bilgi kesin değildir. Örneğin “bir kişinin yaşı 30–35 yaş aralığındadır” veya “X kişinin yaşı 40’tan büyüktür” tanımı varsa bu bir kesin olmayan değerdir. Kesin olmayan değerlerin uç noktaları “kesin” değer veya “null” değerlerdir. Çünkü “kesin” değerde veri kesin bir noktaya işaret eder, “null” değerde ise hiçbir veri olmadığına işaret edilir.

Bulanık Olma (Vagueness): Bulanık olma da kesin olmayan gibidir fakat genelde dilsel olarak ifade edilir. “X kişinin yaşı gençtir.” ifadesi bu terime örnek gösterilebilir.

Belirsizlik (Uncertainty): Bazı zamanlarda gerçek dünya verileri hakkındaki bilgiler bir kesinlik arz etmez. Belirsizlik nitelik değerinin doğruluk derecesiyle ilgilidir. İfade edilen cümle hatalı değildir bununla birlikte ifade %100 doğru değildir. Mesela “X kişinin yaşı 31 veya 32’dir”, kesin olmayan bir değer ifade ederken; “X kişinin yaşı muhtemelen 32’dir”, belirsizlik ifade eder.

BÖLÜM 3

BULANIK NESNEYE YÖNELİK VERİTABANI YÖNETİM SİSTEMİ

NYVTYS'lerinde tamamlanmamış veri ile ilgili birçok yaklaşım vardır bununla birlikte birçok çözüm çeşidi de sunulmuştur. Bunlardan birisi kesin olmayan ve belirsiz verilerin değişik aşamalarda açıkça gösterilmesine izin veren bulanık kümeler tabanlı genişletilmiş veri modelidir. Bu aşamalar;

- a) Nitelik değeri aşaması,
- b) Nesne ve sınıfları arasındaki ilişki aşaması,
- c) Sınıf hiyerarşisi,
- d) Bir niteliğin nesneye bağlantısı veya ilişkisidir.

3.1. BULANIK NESNEYE YÖNELİK VERİTABANI YAKLAŞIMLARI

Bulanık Nesneye Yönelik Veri Modeli (BNYVM), NYVM'ne esneklik kazandırılması ve genişletilmesiyle oluşturulan bir veri modelidir. Bu amaçla çeşitli çalışmalar yapılmıştır. Bazı temel yaklaşımlar aşağıdaki gibi açıklanabilir:

Nancy Van Gyseghem ve arkadaşları tarafından önerilen yaklaşım bulanık bilgiyi bulanık kümeler ve belirsiz verileri de genelleştirilmiş bulanık kümeler metoduyla modelleyerek birbirinden ayırmıştır [25].

Bir diğer çalışma ise Bordogna ve arkadaşlarının sunduğu graf tabanlı olasılık hesaplarına dayanan veri modelidir. Dilsel tanımlamaların kullanımı teklif edilmiş ve nesne ile sınıf arasındaki ilişkilerde birleştirilerek kullanılmıştır [26].

Dubois ve arkadaşları, olasılık teorisini kullanarak sınıf hiyerarşilerinde belirsizlik ve bulanık olmayı inceleyerek; hiyerarşik olarak nitelik, sınıf üst sınıf dâhil edilme dereceleri ve bulanık küme aralıkları üzerinde çalışmışlardır [27].

Benzerlik ilişkisini baz alarak yapılan George ve arkadaşlarının çalışmaları; veri aşamasındaki kesin olmamayı değerlendirerek nesne-sınıf ve sınıf-alt sınıf yapılarındaki belirsizliğe katkısını göz önünde bulundurmuşlardır [28]. Sonrasında bu çalışma Yazıcı ve George tarafından geliştirilerek NYVM’nde belirsiz verinin değişik tiplerinin daha doğru bir şekilde gösterimi sağlanmıştır [29].

3.2. BENZERLİK TABANLI NESNEYE YÖNELİK VERİ TABANI SİSTEMİNİN MANTIKSAL TASARIMI

Çalışmamızda referans alınan bu model Yazıcı ve George tarafından geliştirilmiştir ve benzerlik ilişkisini baz almaktadır. Bu sebeple bu bölümde ayrıntılı olarak yer verilecektir. Bu yaklaşımda D alanındaki x , y ve z nitelik değerleri için benzerlik ilişkisi şu şekildedir [30]:

- a) $s(x, x)=1$ (yansıma).
- b) $s(x, y)=s(y, x)$ (simetri).
- c) $s(x, z) \geq \max_{y \in D} \{ \min [s(x, y), s(y, z)] \}$ (geçişlilik).

Her bulanık nitelik için, bir bulanık alan ve benzerlik matrisi tanımlanır. Bulanık alan niteliğin sınıfından bağımsız olarak alabileceği değerler kümesidir ve dom ile gösterilir. Bir niteliğin “aralığı” ise bir sınıfın üyesinin bir nitelik için izin verilen değerler kümesidir. Genel olarak ifade etmek gerekirse aralık alanın alt kümesidir. Bir C sınıfının a_i adlı niteliği için aralık tanımı; $rng_C(a_i)$ ile gösterilir. Burada $a_i \in Attr(C) = \{a_1, a_2, \dots, a_n\}$ ve $Attr(C)$, C sınıfının nitelikleridir.

Benzer nesnelere nesne-sınıf ve sınıf-üst sınıf seviyesinde bir sınıf ve bulanıklık oluşturmak için birlikte gruplanır. Bulanıklık oluşturulan bir nesnenin sınıfıyla oluşturduğu ilişkide ortaya çıkar. Bir nesne, bir sınıfa belirli bir üyelik derecesiyle

aıttır. Nitelik deęerlerinin aralık ve iliřkilerine gre, C sınıfındaki o_j nesnesinin $Attr(C)$ nitelikleri baz alınarak yelik derecesi hesaplanır (Eřitlik 3.1).

$$\mu_C(o_j) = g \left[f \left(RLV(a_i, C), INC \left(\frac{rng_C(a_i)}{o_j(a_i)} \right) \right) \right] \quad (3.1)$$

Eřitlik 3.1’de; $RLV(a_i, C)$, a_i nitelięinin C sınıfıyla olan iliřkisini; $INC(rng_C(a_i)/o_j(a_i))$ C sınıfındaki a_i ’nin belirlenen aralıęındaki o_j ’nin nitelik deęerlerinin dhil olma derecelerini; f fonksiyonu, sınıftaki n tane nitelięin toplamını; g fonksiyonu nesne ve sınıf-st sınıf arasındaki var olan baęlantı tipini gsterir. $RLV(a_i, C)$ kullanıcı tarafından tanımlanabilir ya da eřitli řekillerde hesaplanabilir.

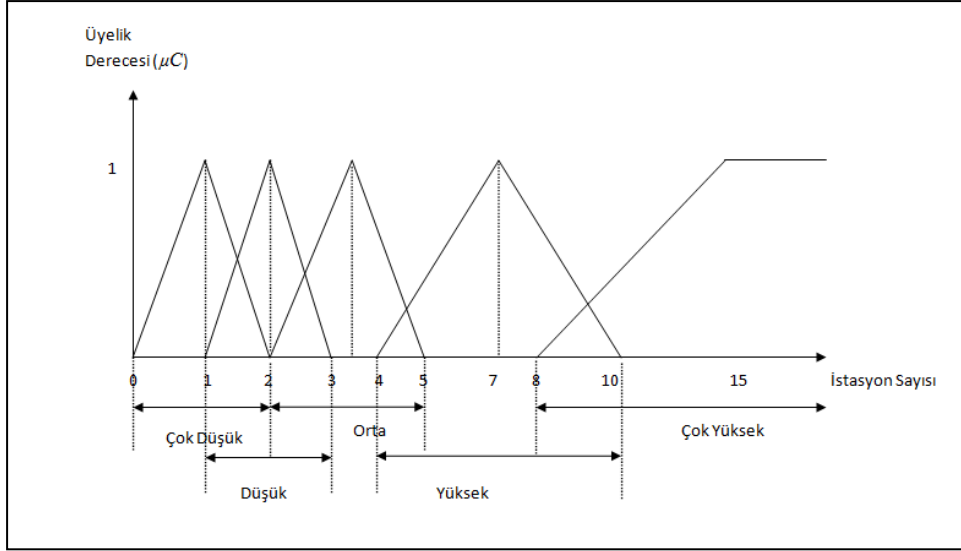
Veritabanında bulanıklık c dzeyde gerekleřtirilmektedir:

- a) Nitelik dzeyi,
- b) Nesne - sınıf dzeyi,
- c) Sınıf - st sınıf dzeyi.

3.2.1. Nitelik Dzeyinde Hesaplamalar

Nitelik dzeyinde bulanık nitelikler deęiřik belirsizlik řekillerinde ifade edilebilir. Eęer bir nitelik deęeri hakkında kesin deęer yerine tanımlayıcı bir nitelik varsa bu bir bulanık deęerdir. Bulanık nitelikler VE (AND), VEYA (OR), ZEL VEYA (XOR) mantıksal operatrlerinden birine sahip olan bir grup bulanık deęerle de ifade edilebilirler.

Nitelik ařamasında bulanık bir nitelik deęerinin kesin deęeri biliniyorsa veritabanına kesin deęerler kaydedilebilir. Bu modelde hem kesin hem de bulanık deęerlerin stesinden gelinebilir ve sorgular da yine kesin ve bulanık deęerlerle yapılabilir. Her iki veri tryle bařa ıkabilmek iin tanımlanmıř aralıklarla blmlenen yelik fonksiyonu grafięine gre belirsiz verilerin yerleri hesaplanır; buldukları aralıklara gre her nesneye nitelikler tanımlanır (řekil.3.1).



Şekil 3.1. Örnek üyelik fonksiyonu grafiği.

BNYVM’nde sorgu çalıştırırken eğer bulanık ve kesin değerler arasında karşılaştırma yapılması gerekirse; sistem kesin değerini bulanık terime olan üyelik derecesini otomatik olarak hesaplar. Eğer üyelik derecesi kullanıcı tarafından tanımlanan eşik değerinden büyükse veya eşik değerine eşitse o zaman nesne verilen durumu karşılar. Sorguya cevap olarak bu veri veritabanından çekilir.

Bu model çok değerli niteliklere izin vererek üyelik değerlerini gerçekçi bir şekilde hesaplar ve bunlar arasındaki ilişkiyi göstermek için mantıksal operatörleri kullanır.

Nitelik ve Sınıf Tanımlamaları:

BNYVM’nde nitelikler birden fazla değer alabilirler ve bu değerler VE, VEYA, ÖZEL VEYA mantıksal operatörleriyle birbirine bağlanırlar. Mantıksal operatörler şu şekilde gösterilirler:

a) VE:<.....>: VE mantıksal operatörü niteliğin birden fazla değer için kullanımını sağlar.

Firma.Bölüm=<İnsan Kaynakları,Muhasebe>

Firmanın bölümleri “İnsan Kaynakları” ve “Muhasebe”dir.

b) VEYA: {.....}: VEYA mantıksal operatörü bir sınıf için yapılabilen en belirsiz tanımdır.

Firma. Bölüm = { İnsan Kaynakları, Muhasebe }

Firmanın bölümleri “İnsan Kaynakları” veya “Muhasebe”dir.

c) ÖZEL VEYA: [.....] : ÖZEL VEYA mantıksal operatörü seçeneklerden birini seçmeyi sağlayarak tek bir niteliğe izin verir.

Firma. Bölüm = [İnsan Kaynakları, Muhasebe]

Firmanın bölümleri “İnsan Kaynakları” veya “Muhasebe” den herhangi biridir.

Her sınıfın tanımında, sahip oldukları bulanık niteliklerin her birinin sınıfla ilişkisini belirten ilişki kuralları ve bir aralık belirlenir. Bu yolla sınıfın “yaklaşık” bir tanımı verilmiş olur. Sınıfın niteliği aralıktan bağımsız olarak bulanık alandan herhangi bir değer alabilir. Bu modelde daha kesin bir sınıf tanımı verebilmek amacıyla mantıksal operatörler sınıf aralık tanımlarıyla ilişkilendirilirler. Mantıksal ilişkilere aralık tanımında karar verilir ve çok nitelikli nesnelere de bu kurala uyarlar. Örnek olarak A,B ve C alanlarından a,b,c niteliklerine sahip bir C sınıfı şu şekilde tanımlanır:

a) $\text{rng}_C(a) = \langle a1, a4, a6 \rangle$ $\text{dom}_C(a) = \{a1, a2, a3, \dots, a_n\}$

b) $\text{rng}_C(b) = \{b4, b6, b8\}$ $\text{dom}_C(b) = \{b1, b2, b3, \dots, b_n\}$

c) $\text{rng}_C(c) = [c1, c6]$ $\text{dom}_C(c) = \{c1, c2, c3, \dots, c_n\}$

İlişki ağırlıkları ise her nitelik için tanımlanır ve belirtilen sınıf için niteliğin aralık tanımının önemini belirtirler. Aşağıdaki örnekte sınıfı belirleyen nitelikler sırasına göre c,a ve b'dir:

$\text{RLV}(a)=1, \text{RLV}(b)=0.5, \text{RLV}(c)=1.8$

3.2.2. Nesne ve Sınıf Düzeyinde Hesaplamalar

Nesne-Sınıf düzeyi bir nesnenin sınıfa olan üyelik derecesine işaret eder. Bulanık sınıfları, kesin sınıflardan ayıran özellik bulanık sınıfların sınırlarının kesin olmamasıdır. Niteliklerin kesin olmaması sınıfın da kesin olmamasına yol açar. Bazı

nesneler sınıfı 1 üyelik derecesiyle tam üyelik gösterirken, bazı nesnelere 0 ile 1 arasında bir dereceyle üyelik gösterirler. Bununla birlikte yine de [0,1] arasında belirtilen bir dereceyle bu sınıfın üyesi olarak ele alınırlar. Kullanılan modelde bir sınıfta kullanılan niteliğin aralığı belirtilmişse de nesne ilişkili olduğu alandan istediği değeri alabilir. Bir nesnenin sınıfa olan üyelik derecesi ise niteliğin ilişki değeri, sınıf aralık değerleri ve nitelikler arasındaki benzerlik ilişkisiyle bulunur. Nesnenin nitelik değeri, sınıfın aralığına ne kadar yakınsa üyelik derecesi o kadar yüksek olur. o_j nesnesinin C sınıfına olan üyeliği Eşitlik 3.2’de gösterilmiştir.

$$\mu_c(o_j) = \frac{\sum INC(rng_c(a_i) * RLV(a_i, C))}{\sum RLV(a_i, C)} \quad (3.2)$$

Eşitlik 3.2’de $rng_c(a_i)$, a_i niteliğinin aralığını; $o_j(a_i)$, o_j nesnesinin a_i niteliğinin değerini, $INC(rng_c(a_i)/o_j(a_i))$ ise a_i niteliğinin değerinin aralığına olan dâhil olma derecesini gösterir.

Dâhil Olma Değerlerinin Hesaplanması:

Eğer bir nesnenin nitelik değeri “0” ise bütün mantıksal operatörler için dâhil olma değeri 0’dır; bunun dışında her bir mantıksal operatör için dâhil olma değerleri aşağıdaki gibi hesaplanır:

VE Mantıksal Operatörü

VE mantıksal operatöründe nitelik birden fazla değer alır ve bütün değerler eş zamanlı olarak var olurlar. Eşitlik 3.3’te VE işlemi görülmektedir:

$$INC(rng(a_i) / (o_j(a_i)) = \text{Min}[\text{Min}[\text{Max}(\mu_s(x, y))], \text{Min}[\text{Max}(\mu_s(z, w))]] \quad (3.3)$$

$$x \in rng(a_i), y \in o_j(a_i), z \in o_j(a_i), w \in rng(a_i)$$

Çizelge 3.1. Boy niteliği için benzerlik matrisi.

Boy	Çok Kısa	Kısa	Orta	Uzun	Çok Uzun
Çok Kısa	1.0	0.9	0.7	0.2	0.2
Kısa	0.9	1.0	0.7	0.2	0.2
Orta	0.7	0.7	1.0	0.2	0.2
Uzun	0.2	0.2	0.2	1.0	0.5
Çok Uzun	0.2	0.2	0.2	0.5	1.0

$rng(boy) = \langle \text{çok kısa, kısa} \rangle$

$n1.boy = \langle \text{kısa, orta} \rangle$

$n2.boy = \langle \text{çok kısa} \rangle$

$n1$ ve $n2$ nesnelерinin verilen özellikleri ve sınıf aralığına göre Çizelge 3.1'e bağlı olarak aşağıdaki hesaplamalar yapılır:

$$INC\left(\frac{rng_{kısıf}(boy)}{n1(boy)}\right) = Min\left[Min\left[Max(\mu_S(kısa, çok\ kısa), \mu_S(kısa, kısa)), Max(\mu_S(orta, çok\ kısa), \mu_S(orta, kısa))\right], Min\left[Max(\mu_S(\text{çok kısa}, kısa), \mu_S(\text{çok kısa}, orta)), Max(\mu_S(kısa, kısa), \mu_S(kısa, orta))\right]\right] = Min\left[Min\left[Max((0.9, 1.0)), Max(0.7, 0.7)\right], Min\left[Max((0.9, 0.7)), Max((1.0, 0.7))\right]\right] = 0.7$$

$$INC\left(\frac{rng_{kısıf}(boy)}{n2(boy)}\right) = 0.9$$

VEYA Mantıksal Operatörü

VEYA mantıksal operatöründe nitelik birden fazla değer alır ve bütün değerler eş zamanlı olarak var olabilirler. VEYA mantıksal operatörü VE'ye göre daha az kesindir. Nitelikler arasındaki benzerlikler azaldıkça, dâhil olma değeri de azalır, çünkü nitelik değerleri arasındaki benzerlik arttıkça belirsizlik azalır. Bu özellik nesneleri, daha yakın ve anlamlı nitelik değerleri almak için zorlar. Eşitlik 3.4'te görülen eşik değeri (Threshold), nesne niteliğinin elemanları arasındaki benzerliğin minimum düzeyini gösterir.

$$INC(rng_C(a_i) / o_j(a_i)) = Min[Max(\mu_s(x, z), Threshold(o_j(a_i)))] \quad (3.4)$$

$$x \in o_j(a_i), y \in rng_C(a_i), Threshold(o_j(a_i)) = Min[\mu_s(x, z)], x, z \in o_j(a_i)$$

n1 ve n2 nesnelerrinin verilen özellikleri ve sınıf aralığına göre Eşitlik 3.4'e ve Çizelge 3.1'e bağılı olarak aşağıdaki hesaplamalar yapılır:

$$rng(\text{boy}) = \{ \text{çok kısa, kısa} \}$$

$$n1.\text{boy} = \{ \text{kısa, orta} \}$$

$$n2.\text{boy} = \{ \text{çok kısa} \}$$

$$INC(rng_{kışl}(boy) / n1(boy)) = Min \left[\begin{array}{l} Max \left(\begin{array}{l} \mu S(kısa, \text{çok kısa}), \mu S(kısa, kısa) \\ \mu S(orta, \text{çok kısa}), \mu S(orta, kısa) \end{array} \right), \\ Threshold(\mu S(\text{çok kısa}, kısa)) \end{array} \right]$$

$$= Min [Max(0.9, 1.0, 0.7, 0.7), 0.7] = 0.7$$

$$INC(rng_{kışl}(boy) / n2(boy)) = 1$$

ÖZEL VEYA Mantıksal Operatörü

ÖZEL VEYA mantıksal operatörü aralıktaki sadece bir elemanın doğru olmasını sağlar. Yani aynı anda sadece tek bir eleman seçilebilir.

$$INC \left(\frac{rng_C(a_i)}{o_j(a_i)} \right) = Avg[Max(\mu_s(x, z))], x \in o_j(a_i), y \in rng_C(a_i) \quad (3.5)$$

n1 ve n2 nesnelerrinin verilen özellikleri ve sınıf aralığına göre Eşitlik 3.5'e ve Çizelge 3.1'e bağılı olarak aşağıdaki hesaplamalar yapılır:

$$rng(\text{boy}) = [\text{çok kısa, kısa}]$$

$$n1.\text{boy} = [\text{kısa, orta}]$$

$$n2.\text{boy} = [\text{çok kısa}]$$

$$INC(rng_{kışl}(boy) / n1(boy)) = Avg [Max (\mu S(kısa, çok kısa) , \mu S(kısa, kısa)) , Max (\mu S(orta, çok kısa) , \mu S(orta, kısa))] = Avg [1.0, 0.7] = 0.85$$

$$INC(rng_{kışl}(boy) / n2(boy)) = 1$$

3.2.3. Sınıf ve Altsınıf Düzeyinde Hesaplamalar

Sınıf ve Altsınıf düzeyinde yapılan hesaplamalar bir sınıfın bir üst sınıfına olan üyelik derecesini göstermektedir. Eşitlik 3.6 ve 3.7’de olası durumlar gösterilmektedir:

$$a) \text{ Durum } rng_C(a_i) = 0 : INC = 0 \quad (3.6)$$

$$b) \text{ Durum } INC = Min(Max[s(x, y)]), x \in rng_C(a_i), y \in rng_C(a_i) \quad (3.7)$$

Sınıf tanımında “Kedigiller” sınıfı için aralık ve ilişkiler belirlenmektedir:

$$Range_{KEDİGİLLER}(hız) = \{ orta, yüksek \}$$

$$Relevance(hız, KEDİGİLLER) = 1.5$$

$$Range_{KEDİGİLLER}(koku alma) = \{ orta, yüksek \}$$

$$relevance(koku alma, KEDİGİLLER) = 0.9$$

Sınıf tanımında “Kedigiller” sınıfının “Çita” üyesi için aralık belirlenmektedir:

$$range_{ÇITA}(hız) = \{ yüksek \}$$

$$range_{ÇITA}(koku alma) = \{ düşük, orta \}$$

Çizelge 3.2 ve Çizelge 3.3’e göre “Çita” üyesinin hız ve koku alma nitelikleri için “Kedigiller” sınıfına dâhil olma değerleri hesaplanmaktadır:

Çizelge 3.2. Hız niteliği için benzerlik matrisi.

Hız	Düşük	Orta	Yüksek
Düşük	1.0	0.6	0.2
Orta	0.6	1.0	0.4
Yüksek	0.2	0.4	1.0

$$INC(rng_{KEDİGİLLER}(hız) / rng_{ÇİTA}(hız)) = \\ = Min [Max(\mu_s(orta, yüksek)), Max(\mu_s(yüksek, yüksek))] = Min(0.4, 1.0) = 0.4$$

Çizelge 3.3. Koku alma niteliği için benzerlik matrisi.

Koku Alma	Düşük	Orta	Yüksek
Düşük	1.0	0.6	0.2
Orta	0.6	1.0	0.4
Yüksek	0.2	0.4	1.0

$$INC(rng_{KEDİGİLLER}(koku alma) / rng_{ÇİTA}(koku alma)) = Min [Max(\mu_s(orta, düşük), \mu_s(orta, orta)), Max(\mu_s(yüksek, düşük), \mu_s(yüksek, orta))] = Min(1, 0.4) = 0.4$$

3.2 No'lu eşitlik kullanılarak alt sınıfın sınıfa olan üyelik derecesi bulunmaktadır:

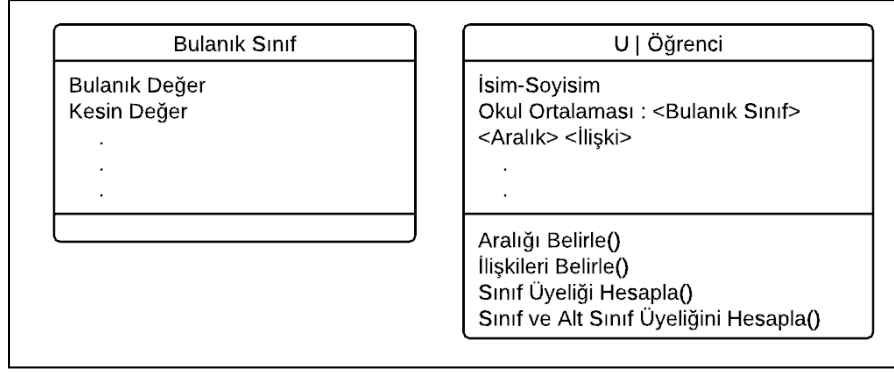
$$\mu_{KEDİGİLLER}(çita) = [INC(rng_{KEDİGİLLER}(hız) / rng_{ÇİTA}(hız)) * RLV(hız, KEDİGİLLER) + INC(rng_{KEDİGİLLER}(koku alma) / rng_{ÇİTA}(koku alma)) * RLV(koku alma, KEDİGİLLER)] = (0.4 * 1.5 + 0.4 * 0.9) / (1.5 + 0.9) = 0.4$$

NYVM'nde bir sınıfı tanımlamak için ilişkiler, nitelikler ve metotlar kullanılır. BNYVT'nda ise bu özelliklerin yanında sınıf için aralık ve ilişki tanımı da yapılmaktadır.

3.3. SINIF HİYERARŞİSİNİN UML DİYAGRAMLARDA GÖSTERİMİ

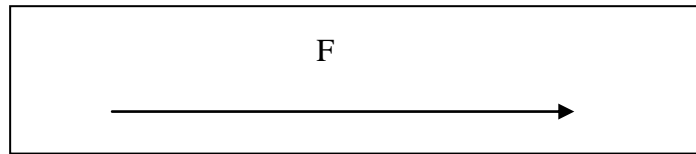
Nesneye yönelik sistemleri modellemek için kullanılan UML diyagramları verileri modellemeyi kolaylaştırmak için geliştirilmiştir. Fakat belirsiz ve kesin olmayan verilerin gösteriminde sadece UML diyagramları kullanmak yeterli değildir ve bunun için diyagramlar farklı metotlarla genişletilerek kullanılmaktadır [31–34].

Yazıcı'nın çalışmalarına göre UML diyagramlarında belirsizlik nitelik, nesne ve sınıf, sınıf ve üst sınıf düzeylerinde olmak üzere üç aşamada ele alınmaktadır [29].



Şekil 3.2. Nitelik, sınıf ve alt sınıf UML diyagram gösterimi.

Şekil 3.2’de gösterildiği gibi belirsizlik tipini belirtmek için bulanık bir sınıf oluşturulmaktadır. Öğrenci sınıfını tanımlarken bu bulanık sınıf, aralık ve ilişki belirlenerek okul ortalaması niteliği bulanık bir değer olarak atanır. Öğrenci sınıfının bulanık değerler içerdiğini göstermek amacıyla sınıf adının yanına “U” işareti konulmaktadır. Nesne ve sınıf ilişkisi bir bulanıklık içeriyorsa “[U]” işareti sınıf isminin yanına yazılarak bu ilişki biçimi tanımlanmış olur. Sınıf ve alt sınıf düzeyinde bir bulanıklık durumunda sınıflar arası ilişkilerin üzerine “F” yazılarak bulanıklık durumu gösterilir (Şekil 3.3).



Şekil 3.3. Bulanık kalıtım ilişkisini gösteren UML gösterimi.

BÖLÜM 4

ATÖLYE YÖNETİM SİSTEMİ

Firmalar, atölye yönetim sistemi kapsamında hangi işlerin, hangi sırayla, ne zaman, hangi maliyetle yapılacağını bilerek belirsizliğin oluşturduğu kapasite düşüşlerini ve dar boğazları ortadan kaldırıp üretim performanslarını artırabilmektedirler.

Bu amaçla önceki yapılan çalışmalarda atölye yönetim sistemlerinin;

- a) Üretim planı,
- b) Çalışması,
- c) Çalışma saatleri,
- d) Materyalleri,
- e) Üretim durumunu yürütmek gibi konular ele alınmıştır.

Bununla birlikte mevcut bulunan atölye yapısının ve veritabanının birbiriyle entegrasyonlarından bahsedilmiş; nesneye yönelik yaklaşımın sistemin bağımsız davranış sergilemesini ve katmanlı bir sistem için ayrı bir davranış entegre etmesini sağladığı belirtilmiş; akıllı imalat sistemlerinin hazırlık zamanlarını, maliyetleri ve karar verme süreçlerini azalttığı konuları üzerinde durulmuştur.

Çalışma kapsamında atölye sistemi ayrıntılı olarak incelenmiş ve bu incelemeler göz önünde bulundurularak sistem tasarımı yapılmıştır.

4.1. ATÖLYE YAPISI

Mevcut atölye yapısı istasyonlardan oluşmaktadır ve her istasyon bir iş grubunu temsil etmektedir. Bu iş grupları genelde makine endeksli olarak tanımlanmakla

birlikte, ofis gibi atölye içerisinde yer alan birimler de iş odaklı olarak tanımlanabilmektedir.

Bu çalışmada atölye maliyetleri beş ana kalemden ele alınarak değerlendirilmiştir:

- a) *Malzeme*: Malzemeler hammadde, yarı mamul veya hazır malzeme olarak ayrılmaktadır. Malzemelerin birçoğu hammadde olarak alınmakta ve atölyede işlenmektedir. Alınan malzemelerin bir kısmı hazır olarak alınıp tasarlanan sistemlerde kullanılırken, diğer kısmı ise yarı mamul olarak alınarak işlemlerde kullanılmaktadır.
- b) *İşçilik*: İstasyonlarda çalışan çeşitli uzmanlar bulunmakla birlikte, işçiler zaman zaman yer değiştirebilmektedirler. Bu kişilerin normal ve fazla mesai olarak yaptığı çalışmalar maliyete yansımaktadır.
- c) *Dış Hizmet*: Atölye dâhilinde işlerin tamamının yapılamaması sebebiyle farklı firmalardan hizmet alımı yapılmaktadır.
- d) *Sarf Malzemeleri*: Atölyede proje kapsamında kullanılacak olan sarf malzemelerin alımı da gerçekleştirilmektedir.
- e) *Ekstra Maliyetler*: Gerçekleştirilecek olan projenin banka, nakliye gibi ek maliyetleri bu başlıkta değerlendirilmektedir.

Müşterinin talebi doğrultusunda gerçekleştirilecek olan iş için ilk olarak ön projelendirilme yapılmaktadır. Bu aşamada işin maliyeti hesaplanarak ve genel planları yapılarak müşteriye bir teklif sunulur. Eğer müşteri bu teklifi kabul ederse teklifin projelendirilmesi için müşteriyle görüşmeler yapılarak, yapılan iş detaylı olarak incelenir. Projelendirme aşamasında kâğıt üzerinde veya gerekli bilgisayar programları kullanılarak proje çizimleri yapılır, projede hedeflenen işin nasıl, hangi sırayla yapılacağına karar verilir sonrasında kullanılacak istasyonlar ve kaynaklar belirlenir. Proje kapsamında kullanılacak olan malzeme, harcanan işçilik süresi, alınacak olan dış hizmet ve sarf malzemeler, yapılabilecek ekstra maliyetlerle

birlikte; atölye altyapısı ve müşteri talebi göz önünde bulundurularak bütçe daha geniş bir şekilde değerlendirilir ve proje alınan kararlara göre hayata geçirilir.

Projenin yapımı devam ederken ilerleme adım adım izlenir, planlar ve ihtiyaçlar projenin işleyişine göre düzenlenir. Yapılan parça veya ekipmanlar sürekli kontrol edilerek projeye uygun olup olmadığı sorgulanır ve gerekirse uygun olmayan kısımlar yenilenir.

Projelerin adım adım takip edilmesi amacıyla her istasyonda yapılan işlerin veri dökümü o işe yönelik formlarda tutularak kayda geçirilmektedir. Bütün bu işlemlerin sistematik bir şekilde gerçekleştirilebilmesi için bütüncül bir yaklaşımla oluşturulacak bir sisteme ihtiyaç duymaktadır. Atölye yönetim sisteminin tasarımı ile bu verilerin etkili bir şekilde kullanımı ve saklanması, bununla birlikte sistemin iyileştirilmesi amaçlanmaktadır.

4.2. SİSTEM ANALİZİ

Yapılan sistem analizleri sonucu mevcut atölye yönetim sistemlerinde;

- a) Bilgi entegrasyonu eksikliği,
- b) Veriye dayalı maliyet hesabı yapılamaması,
- c) Verimlilik hesabı yapılamaması,
 - a) İşçilik ve makine verimliliğinin hesaplanamaması,
 - b) İşin ne kadar sürede yapılmakta olduğunun bilinmemesi,
 - c) Dış hizmet ve atölye imkânları açısından verimlilik kıyaslaması yapılamaması,
 - d) Teklif verme aşamasında parametrik değil tahmini maliyetler sunulması,
- d) Teklif verme sürecinin çok uzun olması gibi sorunların olduğu görülmüştür.

Dolayısıyla imalatta yer alacak tüm verilerin analizinin yapılarak iş maliyetlerinin ortaya konulması ve karar destek sisteminin oluşturulması bu projenin başlıca amacını oluşturmuştur.

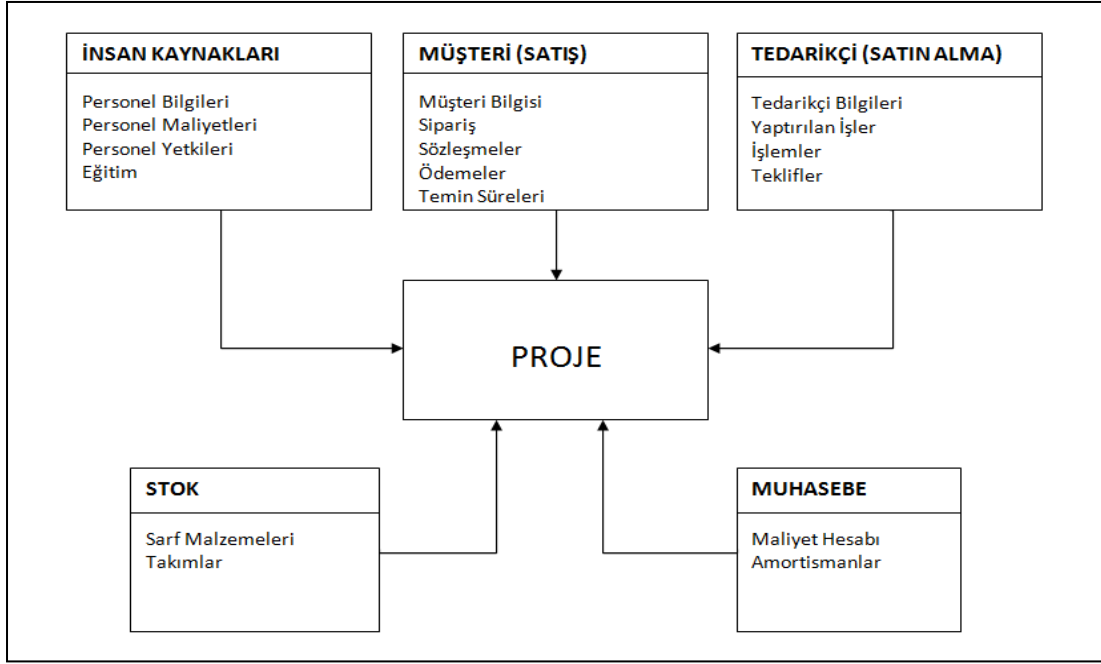
Ek açıklamalar A örneğinde görülmektedir ki mevcut durumda bir firmada maliyetler, bir Excel tablosunda tutularak hesaplanmaktadır ve bu veriler de ancak proje gerçekleştirildikten sonra kayda geçirilebilmektedir. Müşterilerden gelen herhangi bir talebe karşı çok sayıdaki projeye ait verilerin incelenmesi ve parametrik olarak teklif amaçlı kullanılması mümkün değildir. Bu yöntem etkin bir yöntem olmamasının yanında verilerin analiz edilmesine de olanak sağlamamaktadır. İşçilik ve makine ile ilgili herhangi bir veri bulunmadığından verimlilikle ilgili bir hesaplama yapılabilmesi mümkün olmamaktadır.

Bu sebeple üretim sektöründe yer alan işletmelerin hızlı bir şekilde, sağlıklı verilere dayalı maliyetleri takip edebileceği ve yine teklif sunabileceği bir ortam hazırlanması hedeflenmektedir. Bu sistem küçük ve orta büyüklükteki işletmelerin %99'unu oluşturduğu ve büyük bir ekonomik katkı sağladığı ülkemizde, makine imalat atölyelerinin kalitesini artırarak zamandan tasarruf etmek ve müşteriye hızlı hizmet sunmak yoluyla atölye yönetimini kolaylaştıracaktır.

4.3. SİSTEM MİMARİSİ

Analizi yapılan sistem ve NYVTYS'nin sağladığı avantajlar göz önünde bulundurularak genel çerçevesi belirtilen sistemin temel mimarisi Şekil 4.1'deki gibi tasarlanmıştır. Atölye Yönetim Sisteminin aşağıdaki modüllerden oluşması düşünülmektedir:

- a) İnsan Kaynakları,
- b) Muhasebe,
- c) Proje,
- d) Satın alma,
- e) Müşteri,
- f) Stok.

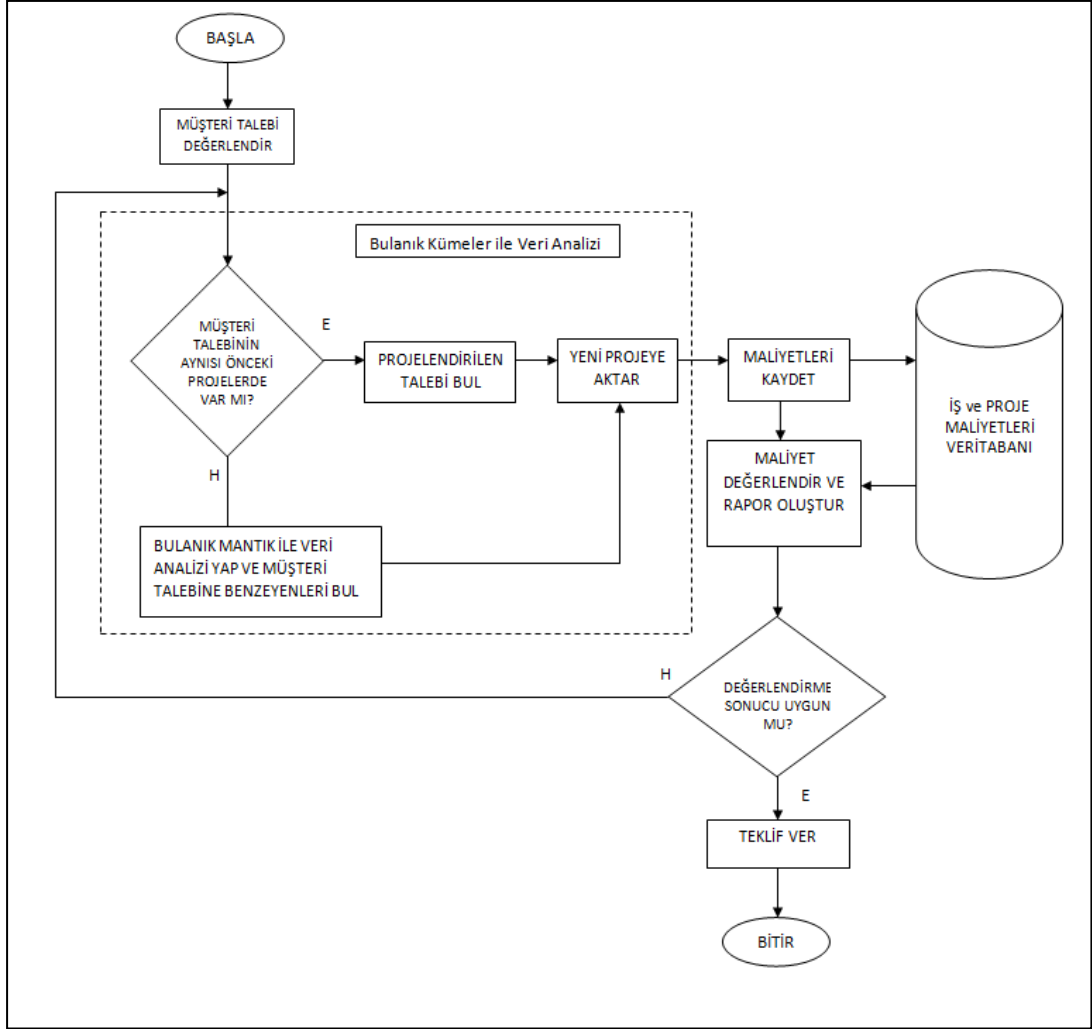


Şekil 4.1. Projenin temel modülleri.

Bununla birlikte sistemin önemli bölümünü “Proje” kısmı oluşturmaktadır. Proje kısmı iki aşamadan oluşmaktadır:

- İmalat aşaması: Önceden yapılan üretimlerin veritabanında tutulması
- Teklif verme aşaması: Mevcut verilerin ele alınarak daha sonraki projelere yapay zekâ kullanılarak teklif verilmesi

Şekil 4.2’de görülmektedir ki öncelikle talep edilen projenin veritabanında olup olmadığına bakılacaktır. Eğer aynısı, daha büyüğü veya küçüğü yapılacaksa maliyetler bu duruma göre hesaplanacak ve teklif verilecektir. Fakat projenin aynısı yoksa ve kabaca bir teklif verilmek isteniyorsa bulanık mantık ile hızlıca bir teklif verilebilir ya da talep edilen projenin ekipmanlarına bakılarak daha ayrıntılı bir maliyet çıkarılabilir. Bununla birlikte aynı mantık ekipman ve proje arasında da geçerli olacaktır. Detaylı incelemede ekipman daha önceden yapılmışsa büyüklüğü hesap edilerek teklife eklenecek olmayanlar için ya bulanık mantık uygulanacak veya ekipmanın içerdiği parçalara bakılarak daha detaylı bir inceleme yapılacaktır ve döngü bu şekilde devam edecektir. (Şekil 4.3)

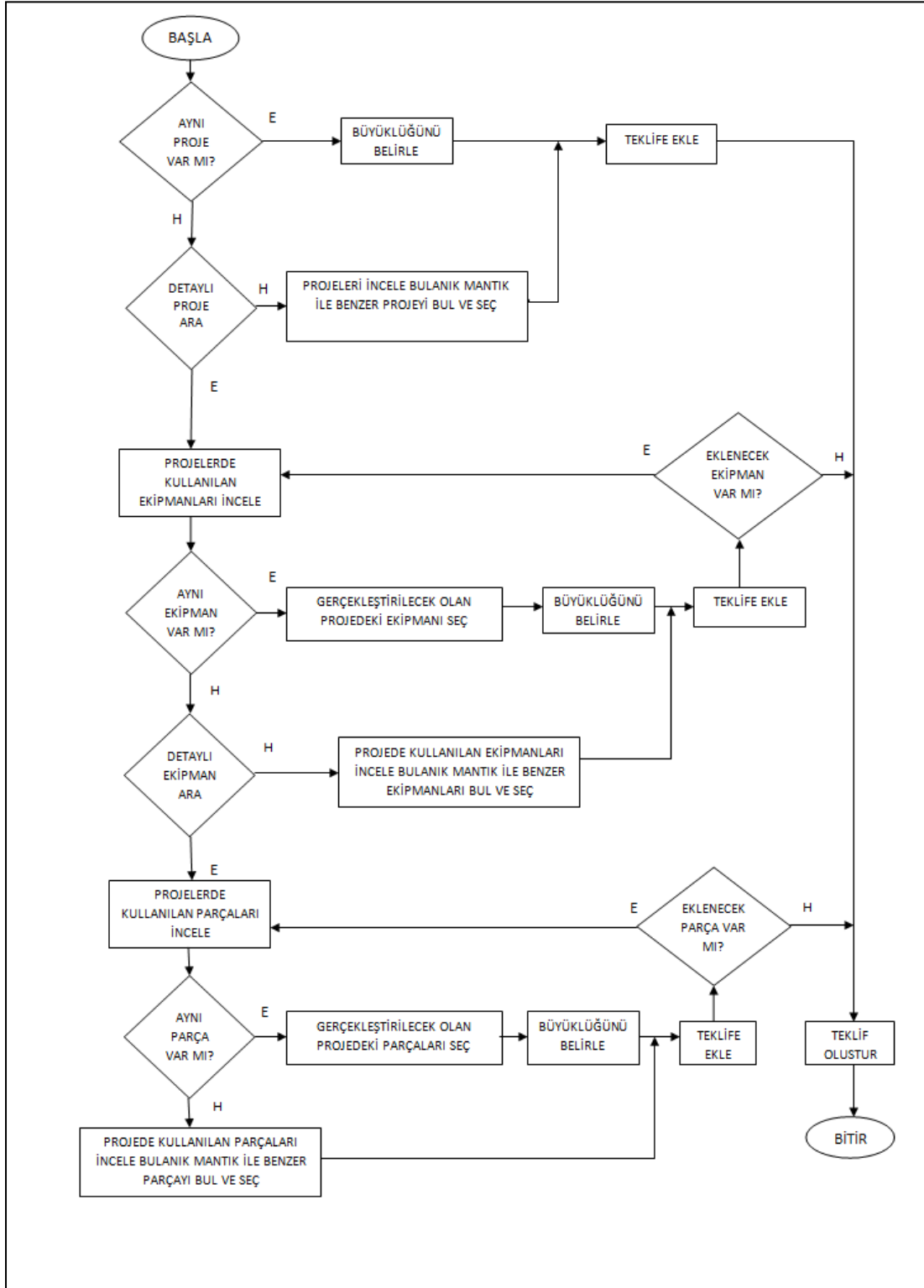


Şekil 4.2. Projenin maliyet değerlendirme ve teklif verme aşamasını gösteren akış diyagramı.

Projeye bağlı olarak bulanık olması gereken nitelikler belirlenmiştir ve bu nitelikleri seçerken özellikle projenin maliyetini ortaya koyan değerler olmasına dikkat edilmiştir.

Sistemde projenin maliyetini belirleyen bulanık özellikler;

- İstasyon sayısı,
- Malzeme büyüklüğü,
- Malzeme üzerindeki çalışma süreleridir.

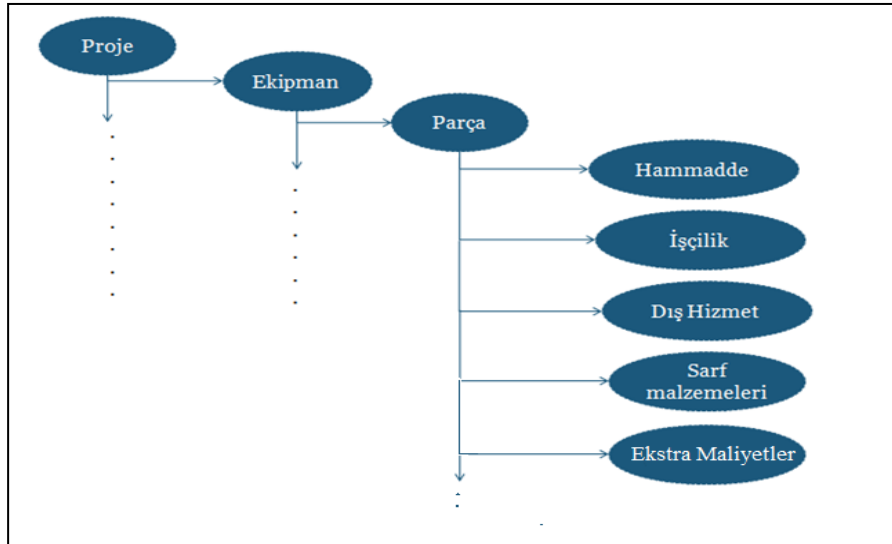


Şekil 4.3. Bulanık küme ile veri analizi aşamasını gösteren akış diyagramı.

Bu nitelikler projelerde maliyeti belirleyen en etkin parametreler olduğundan ve kesin değerler içermediğinden dolayı bulanık olarak tanımlanmışlardır. Bir malzeme

üzerindeki çalışma saati; kişinin yetkinliği, normal saat veya mesai saati farkı, kişinin daha önce bu işi yapmış olması gibi sebeplere bağlı olarak değişebilmektedir. Mekanik işlemleri tanımlayan istasyonlar; aynı parça için her seferinde farklı sayılarda kullanılabilir. Aynı istasyonun birden fazla olması ve farklı verimlilikte çalışması bir parça üzerindeki işlem süresinden, kişi sayısına kadar değişiklik oluşturulabilmektedir. Örneğin, eski olan istasyonlarda çıkabilecek olan arızalar parça süresini etkileyen nedenlerden birisi olabilir. Malzeme büyüklüğü ise projenin tipine göre alternatif malzemelerle gerçekleştirilebileceğinden maliyeti etkileyen özelliklerden birisi olarak seçilmiştir. Bu özelliklerin kesin değerler içermemesi bulanık veritabanı seçilmesinde etkili olan sebeplerdir.

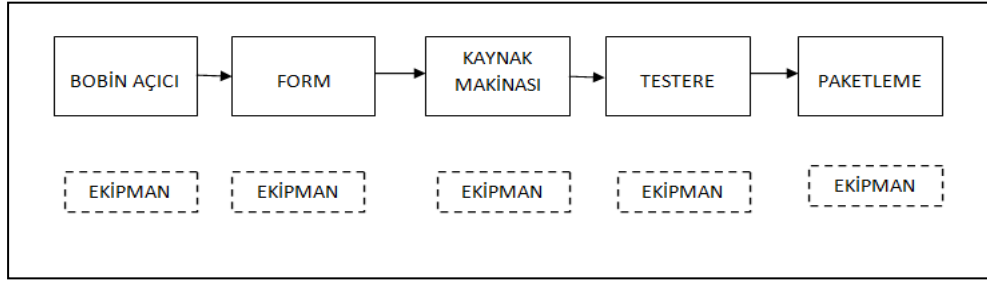
Nesneye yönelik veritabanında yaptığımız sisteme örnek bir proje modülü Şekil 4.4.'te görülmektedir. Proje modülü kırıncımlı bir yapıdan oluşturulmuştur. Nesneye yönelik veritabanının hiyerarşik yapısı bu yapının gerçekleştirilmesi açısından büyük bir kolaylık sağlamıştır. İçten dışa doğru parçalar ekipmanları, ekipmanlar ise projeyi oluşturmuş ve maliyetler toplanarak projenin toplam maliyeti elde edilmiştir.



Şekil 4.4. Veritabanının proje modülü.

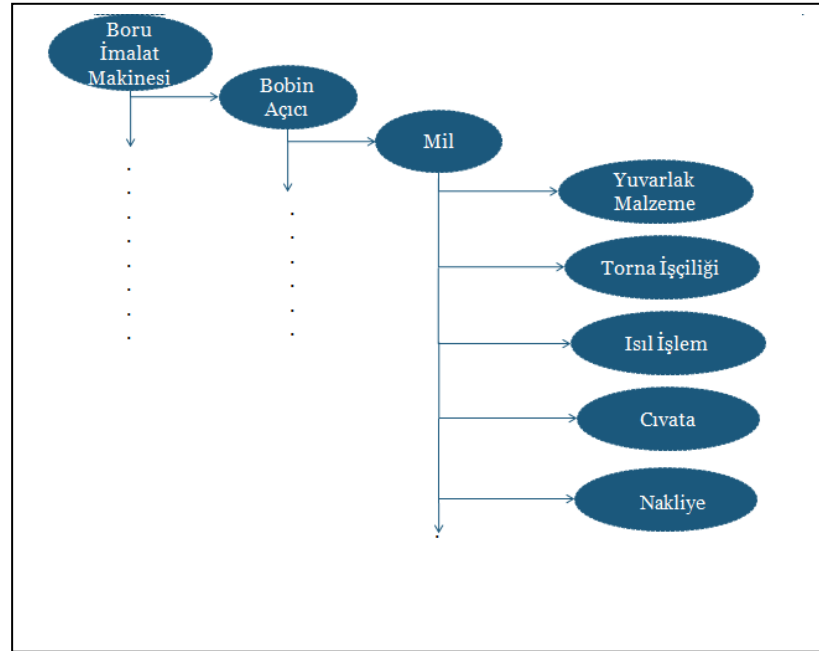
Örnek Proje - Boru İmalat Makinesi

Bir müşteriden boru imalat makinesi talebi geldiğinde bu makineyi oluşturan her ekipman parçalardan başlayarak hammadde, işçilik, dış hizmet, sarf malzeme maliyetleri sisteme kaydedilecek, her ekipmana ait olan parçaların maliyetleri toplanarak bunlar ekipman maliyetlerini oluşturacak, ekipmanların maliyeti toplanarak projenin maliyeti ortaya çıkacaktır (Şekil 4.5 ve Şekil 4.6).



Şekil 4.5. Boru imalat makinesinin ekipmanları.

Tasarlanan veritabanı sayesinde her yeni gelen proje sisteme kaydedilmektedir. Bu kayıt işlemi planlanan ve gerçekleşen değerlerin ayrı olarak kaydedilmesine imkân sağlamaktadır.



Şekil 4.6. Boru imalat makinesinin maliyet şeması.

4.4. NESNEYE YÖNELİK VERİTABANI MODELİ

Projenin altyapısını oluşturan veritabanı tasarımı üzerine çalışmalar yapılmış ve uygulanacak sistemin modeli çıkartılmıştır. Veritabanı tasarlanırken şu sıra izlenmiştir:

- a) Atölye yönetim sisteminin analizinin yapılması ve sistem ihtiyaçlarının belirlenmesi,
- b) Atölye yönetim sisteminin mimarisinin tasarlanması,
- c) NYVT ve BNYVT üzerinde çalışmalar yapılması,
- d) Veritabanı modelinin çıkarılması,
- e) Uygulamada gerekli olan sınıfların oluşturularak aralarındaki ilişkilerin tanımlanması,
- f) Örnek bir atölyenin sistem analizi yapılarak belirlenmiş olan yapıya uygun olarak veritabanı sisteminin oluşturulması.

4.4.1. Kullanılan Veritabanı

Bu çalışmada yaygın olarak kullanılan NYVTYS'lerinden biri olan db4o kullanılmıştır. Db4o, db4objects, Inc. tarafından üretilen, .NET ve Java platformlarında kullanılabilen saf NYVTYS'lerinden biridir. Db4o bir sunucu sistemi yerine kütüphane olarak geliştirilmiş ki bu da onun küçük bir bellek kaplamasını sağlar. Bununla birlikte ürün web sitesinin sunduğu açıklamalı notlar, forum, program indirme destekleri bu ürünün seçilmesinde etkili olan özelliklerden birkaçıdır.

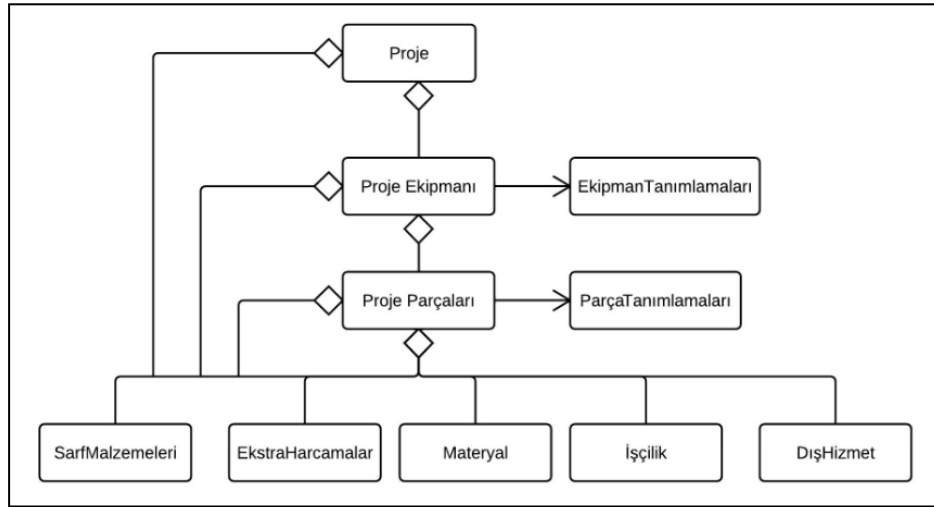
İlişkisel veritabanları yaygın olarak kullanılıyor olsa da, db4o gibi NYVT onların kullanımına karşı bir çözüm olarak değil, yetersiz olduğu durumlarda kullanılacak bir seçenek olarak sunulmaktadır ve “tek beden herkese uymaz” mantığını öne sürer. Klasik veritabanlarında olduğu gibi bütünlük, tutarlılık, izolasyon ve devamlılık karakteristiklerini desteklemekle birlikte, sınıf tanımlamaları ve nesne ilişkileri veritabanı şemasını oluşturmak için yeterlidir, herhangi bir

haritalamaya ihtiyaç duymamaktadır. Bir uygulamayı gerçekleştirmek için NYP dili yeterli olmaktadır.

Db4o, Query by Example (QBE), Language Integrated Query (LINQ), Native Queries (NQ) ve Simple Object Database Access Query (SODA) gibi değişik sorgulama dillerini desteklemektedir. Hatta .NET platformları için LINQ tavsiye edilmektedir. Bununla birlikte Object Manager Enterprise modülü ara yüz aracılığıyla bazı basit çek ve bırak fonksiyonlarıyla sorgular oluşturulmasına ve bu pencere üzerinden sorgu sonuçlarının incelenmesine izin verir.

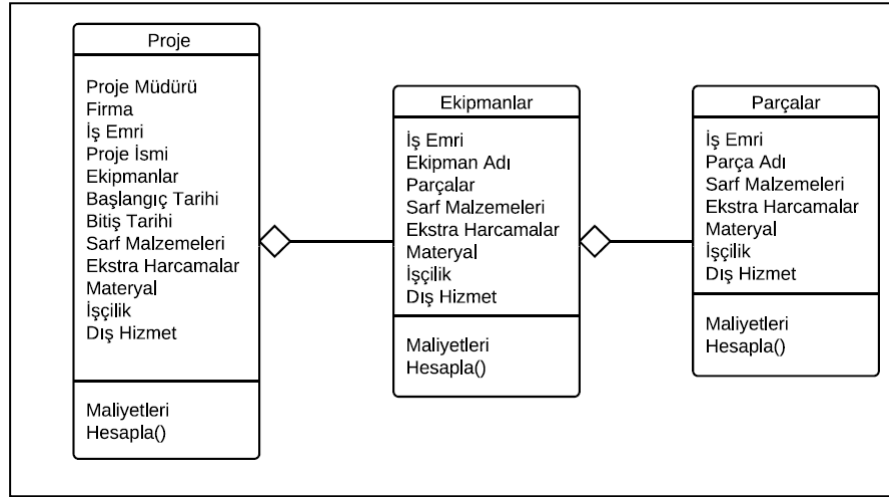
4.4.2. UML Diyagramları

Veritabanı modelini daha iyi sunmak ve görebilmek amacıyla UML diyagramları kullanılmıştır. UML diyagramları sayesinde çalışmanın veritabanı modeli daha iyi tanımlanmış, görüntülenmiş ve oluşturulmuştur.



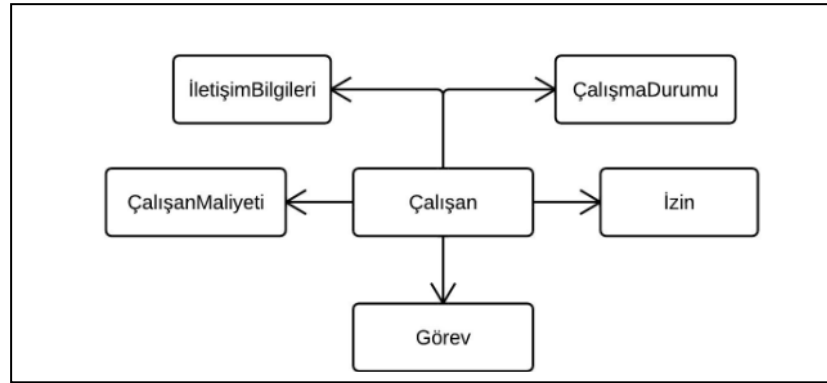
Şekil 4.7. Proje modülü.

Şekil 4.7’de proje, ekipman ve parçalar ve bunların projelerle ilişkilendirilmesi ve kaydedilmesi için gerekli bilgiler yer almaktadır. Ayrıca parçadan itibaren projeye kadar maliyet hesaplaması da yapılacak ve bilgiler kaydedilecektir. Şekil 4.8’de ise proje modülünün detayı görülmektedir.



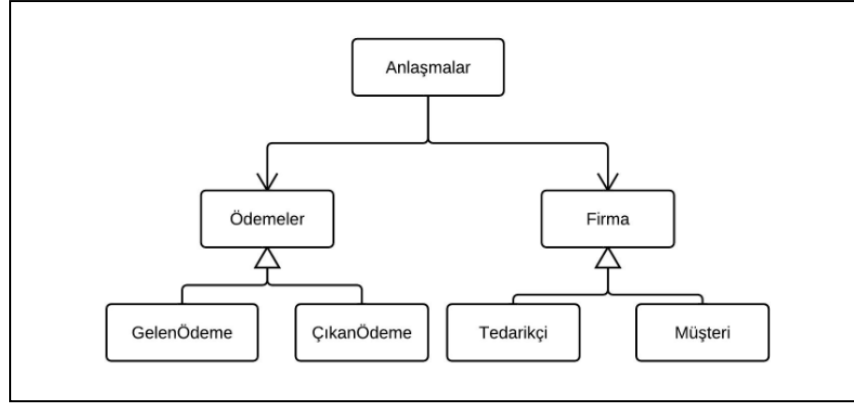
Şekil 4.8. Proje modülü detayı.

Şekil 4.9.'da UML diyagramları vasıtasıyla İnsan Kaynakları bölümünün tasarımı görülmektedir. Bu bölüm firmada görev alan ve daha önceden görev almış olan her seviyeden çalışanın bilgilerini tutmak amacıyla planlanmıştır.



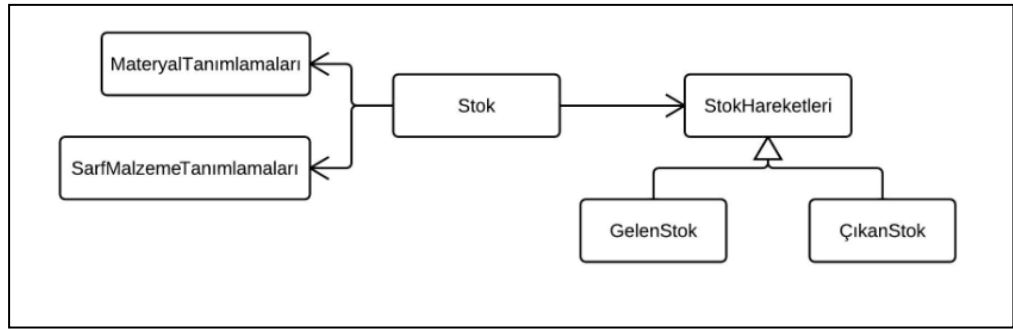
Şekil 4.9. İnsan kaynakları modülü.

Şekil 4.10'da firmanın Satış ve Satın Alma Bölümleri ile ilgili diyagram görülmektedir. Veritabanının bu diyagrama göre tasarlanan kısmında firma ile ilgili bilgiler kaydedilecek, firmalarla yapılacak sözleşmelerle ilgili bilgiler yer alacak ve ödeme gibi sözleşme şartları bu bölümde bulunacaktır.



Şekil 4.10. Satış ve satın alma modülü.

Şekil 4.11’de görülen Stok Bölümü firmaya alınan malzemelerle ilgileri bilgileri ve maliyetleri tutacak, bununla birlikte atölye ile stok bölümü arasında yapılan malzeme akışını kontrol edecektir.



Şekil 4.11. Stok modülü.

4.4.3. Veritabanı Sorguları

Db4o veritabanı QBE, LINQ, NQ ve SODA olmak üzere dört çeşit sorgu desteklemektedir:

a) Veritabanında kullanılan temel kaydetme, silme ve güncelleme sorguları aşağıdaki gibidir:

```

public void Set(object obj)
{
    Db4oHttpModule.Client.Set(obj);
    Db4oHttpModule.Client.Commit();
}
  
```

```

public void Store(object obj)
{
    Db4oHttpModule.Client.Store(obj);
    Db4oHttpModule.Client.Commit();
}

public void Set(object obj)
{
    Db4oHttpModule.Client.Set(obj);
    Db4oHttpModule.Client.Commit();
}

```

- b) QBE sorguları hızlı ve öğrenmek için ideal bir sorgu dilidir. Basit ve kullanımı kolaydır. Mantıksal operatör gerektirmeyen sorgular için idealdir.

```

Object[] result = db.Get(typeof(Project));

IList<ProjectParts> result =
Db4oHttpModule.Client.Query<ProjectParts>(typeof(ProjectParts)
);

```

- c) NQ'nin temeli programlama dillerine dayanır ve bir sorguyu ifade etmenin en iyi yöntemi o programlama dilinin kendisini kullanmaktır tezini savunur. NQ parametre olarak döndüreceği sınıfı alarak bir metot oluşturur ve bu metodun sonucunun “doğru” olduğu nesnelere listelenir.

```

IList<Project> project =Db4oHttpModule.Client.Query<Project>
(delegate(Project objProject)
{
    return objProject.AppExternalServiceCost>100;
});

```

- d) Visual Studio 2008'le birlikte gelen LINQ özelliği; değişik türde bilgileri çekmek için kodların içinde kullanılan SQL gibi bir grup tanımlanmış kurallardır. Db4o, .NET versiyon 3.5 için tam LINQ desteği vermektedir ve LINQ sayesinde nesneye yönelik sorgular yapılabilir. Aşağıdaki sorguda görüldüğü üzere db4o veri kaynağı olarak kullanılmaktadır gerisi ise LINQ sorgu sözdizimidir.

```

var varPart = (from ProjectParts p in Db4oHttpModule.Client
where p.ObjectId.Equals(guidPart)
select p).Single();

```

- e) SODA db4o'nun düşük seviyeli bir sorgulama ara yüzüdür. Birçok uygulama için NQ ve LINQ daha iyi çözümlerdir.

```
IQuery query = db.Query();  
query.Constrain(typeof(Project));  
IObjectSet result = query.Execute();
```

4.5. BULANIK NESNEYE YÖNELİK VERİTABANI MODELİ

BNYVM, NYVM'ne bulanık özellikler eklenerek genişletilmesiyle oluşturulmaktadır. Çalışmamızda nesnenin niteliğine, bulanık özellik eklenmesi sağlanarak bu model elde edilmiştir. Bu sayede nesnelere kesin değerlerle birlikte bulanık değerler de alabilmektedir.

Örneğin, istasyon sayısının klasik NYVT'ında;

Nesne.İstasyonSayısı=3;

Şeklinde gösterilirken; bunun BNYVT'ında kullanımı;

Nesne.İstasyonSayısı=Düşük,Orta;

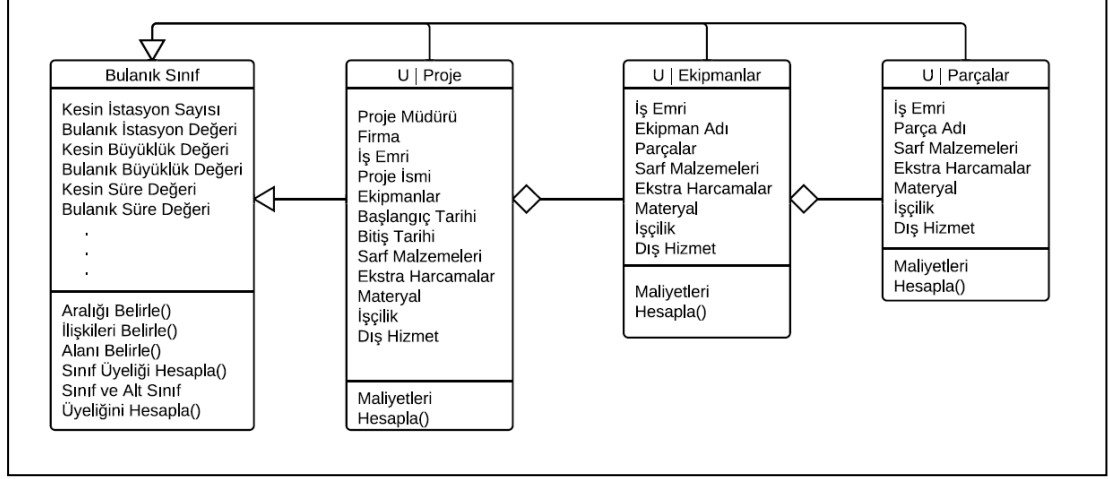
şeklinde dir. Bununla birlikte bulanık olarak sorgu yapma imkânı da sağlamaktadır. Örneğin NYVT'ında “*üç farklı istasyon kullanılmış olan projeleri göster*” şeklinde bir sorgu yazılabilirken; BNYVT'ında “*istasyon sayısının düşük olduğu projeleri göster*” şeklinde kesin bir değer içermeyen sorgular da çalıştırılabilir.

4.5.1. Genişletilmiş UML Diyagramları

BNYVT modelini gösterebilmek için UML diyagramları genişletilerek bulanık değerleri göstermesi sağlanmıştır. Çalışmada ortak bulanık özellikler belirlenerek bulanık bir sınıf oluşturulmuş ve kalıtım yoluyla gerekli sınıfların bu özellikleri kullanması sağlanmıştır.

Şekil 4.12'de görülmektedir ki bulanık sınıf çalışmamızda bulanık olarak belirlenen nitelikler olan; istasyon sayısı, büyüklük ve süre nitelikleri hem bulanık değerler hem de kesin değerler içermektedir. Bu sayede belirlenen niteliklerle ilgili bulanık ve kesin değerler için hem kayıt hem de sorgu imkânı tanınmaktadır. Proje, ekipmanlar

ve parçalar sınıfları da bu özellikleri ve bu sınıfın metotlarını kalıtım yoluyla kullanmaktadır.



Şekil 4.12. Bulanık nitelik içeren sınıflar.

4.5.2. Bulanık Değerlerin Mantıksal Kullanımı

Veritabanındaki bulanık değerlerin sorgulanması sistemin “Teklif” ara yüzünde gerçekleşmektedir. Müşteriye teklif verilmek istendiğinde talebe en yakın proje, ekipman veya parçayı bulmak amacıyla bulanık sorgu kullanılmaktadır.

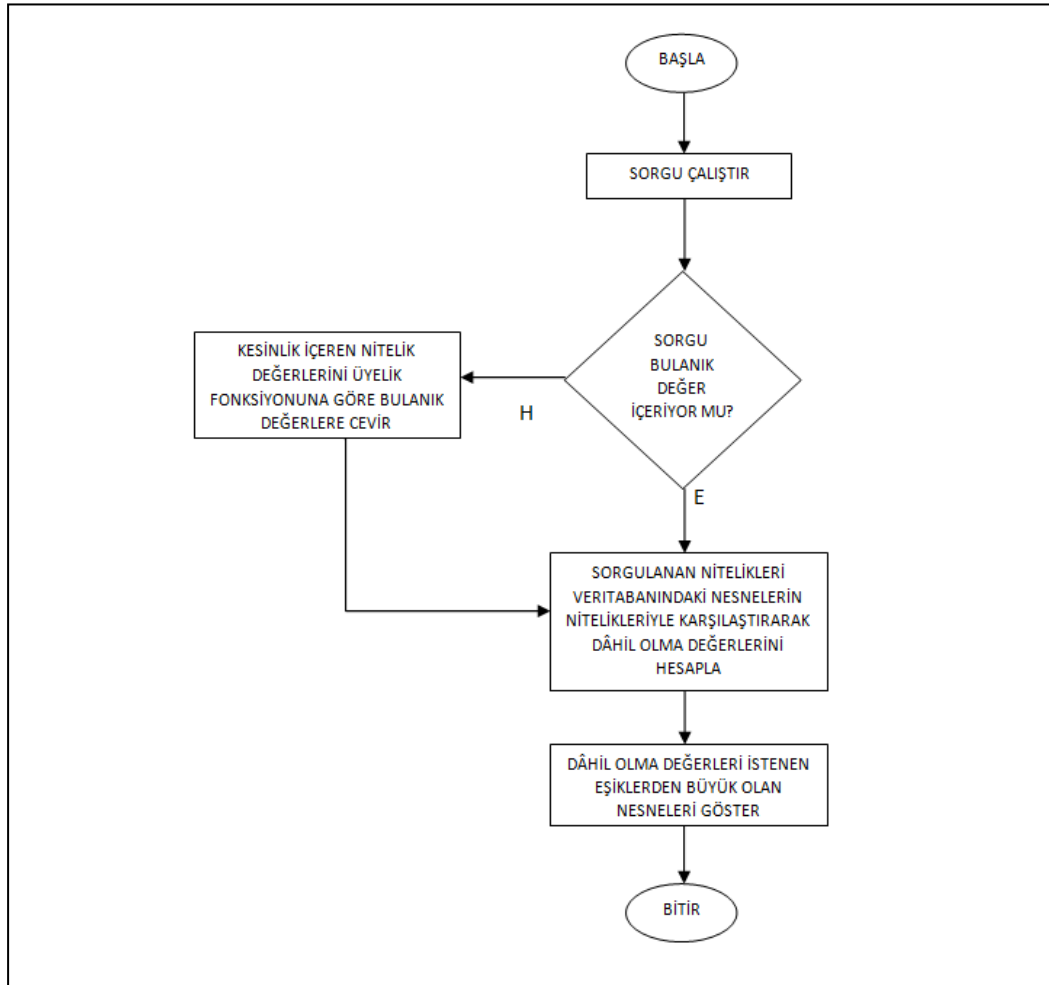
Örnek olarak talep edilen projede bulanık özellikler olan; parçanın büyüklüğü, istasyon sayısı ve toplam süresinin ne kadar olacağı ListBox’lardan seçilerek veritabanındaki projede kullanılan parçalarla karşılaştırılır (Şekil 4.13). İstenen eşğin üzerindeki parçalar ekranda gösterilir.

Burada “ShowPart” butonuna tıkladığında Şekil 4.14’teki akış diyagramında görülen sıra izlenmekte ve sonuç olarak Şekil 4.15’teki sonuçlar gösterilmektedir.

FUZZY PART QUERY PAGE

<input checked="" type="radio"/> FuzzySize VeryLow ▲ Low ▲ Medium ▲ High ▼	<input checked="" type="radio"/> FuzzyStationNo VeryLow ▲ Low ▲ Medium ▲ High ▼	<input checked="" type="radio"/> FuzzyWeek VeryLow ▲ Low ▲ Medium ▲ High ▼
<input type="radio"/> CrispSize <input type="text"/>	<input type="radio"/> CrispStationNo <input type="text"/>	<input type="radio"/> CrispWeek <input type="text"/>
Threshold <input type="text" value="0,2"/>	Threshold <input type="text" value="0,2"/>	Threshold <input type="text" value="0,7"/>
<input type="button" value="ShowPart"/>		

Şekil 4.13. Ara yüzde bulanık özelliklere sahip parçaların sorgulanması.



Şekil 4.14. Bir sorgunun çalıştırılması akış diyagramı.

JobOrder	PartName	SizeIncVal	StationIncVal	WeekIncVal	
120100-03-03	Delik İşleme Seyyar Freze(Modül 2) X Eksen İşleri (AG-01)	0,6	1	0,8	ShowPartInformation AddPart
120100-02-01	Küçük Seyyar Freze(Modül 1) X Eksen İşleri(KD-11)	0,8	1	0,8	ShowPartInformation AddPart
120100-01-01	Büyük Seyyar Freze(Modül 1) X Eksen İşleri(KD-11)	0,2	0,5	0,8	ShowPartInformation AddPart

Şekil 4.15. Ara yüzde bulanık özelliklere sahip parçaların gösterilmesi.

$$DelikİşlemeSeyyarFreze.Büyüklik = \{ düşük, orta \}$$

$$DelikİşlemeSeyyarFreze.İstasyon Sayısı = \{ yüksek \}$$

$$DelikİşlemeSeyyarFreze.Süre = \{ çok düşük \}$$

ListBox'dan büyüklük niteliği için “Çok düşük - Düşük” aralığı sorgulanmıştır. Dâhil olma değerleri Çizelge 4.1'e göre hesaplanır:

Çizelge 4.1. Büyüklük için benzerlik matrisi.

Büyüklik	Çok Düşük	Düşük	Orta	Yüksek	Çok Yüksek
Çok Düşük	1.0	0.8	0.6	0.2	0.1
Düşük	0.8	1.0	0.6	0.2	0.1
Orta	0.6	0.6	1.0	0.2	0.1
Yüksek	0.2	0.2	0.2	1.0	0.5
Çok Yüksek	0.1	0.1	0.1	0.5	1.0

$$INC(rng_{PARÇA}(Büyüklik) / DelikİşlemeSeyyarFreze(Büyüklik)) = Min [Max (\mu_S(\text{çok düşük}, düşük), \mu_S(\text{çok düşük}, orta), \mu_S(\text{düşük}, düşük), \mu_S(\text{düşük}, orta)), Threshold(\mu_S(\text{düşük}, orta))] = Min [Max (0.8, 0.6, 1.0, 0.6), 0.6] = 0.6$$

$$INC(rng_{PARÇA}(Büyüklik) / KüçükSeyyarFreze(Büyüklik)) = 0.8$$

$$INC(rng_{PARÇA}(İstasyon Sayısı) / BüyükSeyyarFreze(İstasyon Sayısı)) = 0.8$$

ListBox'dan süre niteliği için “Orta - Yüksek” aralığı sorgulanmıştır. Dâhil olma değerleri Çizelge 4.2'ye göre hesaplanır:

Çizelge 4.2. Süre için benzerlik matrisi.

Süre	Çok Düşük	Düşük	Orta	Yüksek	Çok Yüksek
Çok Düşük	1.0	0.8	0.6	0.2	0.1
Düşük	0.8	1.0	0.6	0.2	0.1
Orta	0.6	0.6	1.0	0.2	0.1
Yüksek	0.2	0.2	0.2	1.0	0.5
Çok Yüksek	0.1	0.1	0.1	0.5	1.0

$$INC(rng_{PARÇA}(Süre) / Delik\dot{I}şlemeSeyyarFreze(Süre)) = Max [\mu_s(orta, yüksek), \mu_s(yüksek, yüksek)] = (0.2, 1.0) = 1$$

$$INC(rng_{PARÇA}(Süre) / KüçükSeyyarFreze (Süre)) = 1$$

$$INC(rng_{PARÇA}(Süre) / BüyükSeyyarFreze (Süre)) = 0.5$$

ListBox'dan istasyon sayısı niteliği için “Çok Düşük” aralığı sorgulanmıştır. Dâhil olma değerleri Çizelge 4.3'e göre hesaplanır:

Çizelge 4.3. İstasyon sayısı için benzerlik matrisi.

İstasyon Sayısı	Çok Düşük	Düşük	Orta	Yüksek	Çok Yüksek
Çok Düşük	1.0	0.8	0.6	0.2	0.1
Düşük	0.8	1.0	0.6	0.2	0.1
Orta	0.6	0.6	1.0	0.2	0.1
Yüksek	0.2	0.2	0.2	1.0	0.5
Çok Yüksek	0.1	0.1	0.1	0.5	1.0

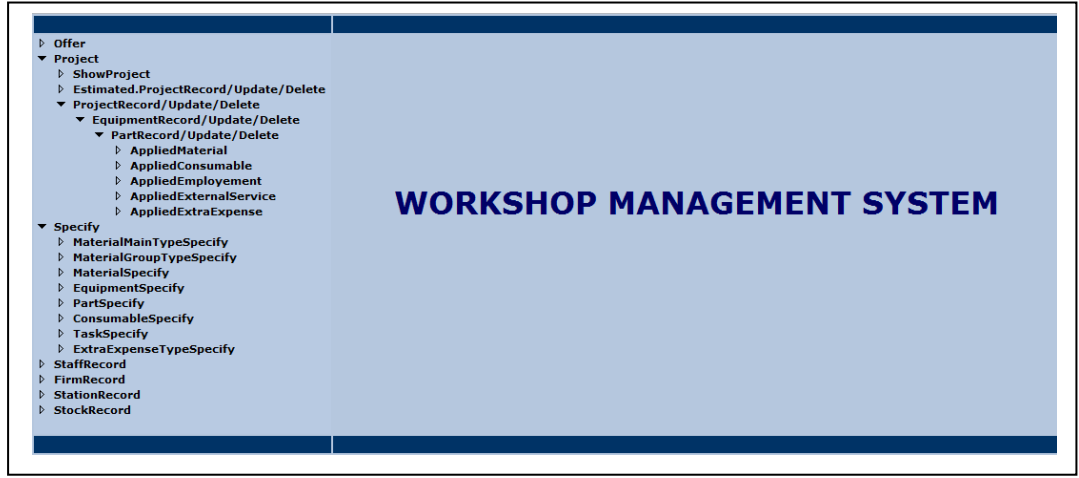
$$INC(rng_{PARÇA}(İstasyon Sayısı) / Delik\dot{I}şlemeSeyyarFreze (İstasyon Sayısı)) = \mu_s(düşük, çok düşük) = 0.8$$

$$INC(rng_{PARÇA}(İstasyon Sayısı) / KüçükSeyyarFreze (İstasyon Sayısı)) = 0.8$$

$$INC(rng_{PARÇA}(İstasyon Sayısı) / BüyükSeyyarFreze (İstasyon Sayısı)) = 0.8$$

4.6. SİSTEM ARAYÜZÜ

Kullanıcıya uzaktan erişim sağlamak ve kolaylık açısından web ara yüz kullanılması uygun görülmüştür ve bu amaçla ASP.NET kullanılmıştır. Sistemin ihtiyaçlarıyla bağlantılı olarak veri kaydetmek, güncellemek ve silmek için sayfalar bulunmakla birlikte; projeleri detaylı inceleyebilmek amacıyla da sayfalar oluşturulmuştur. Şekil 4.16'da proje giriş sayfası görülmektedir.



Şekil 4.16. Proje giriş sayfası.

Şekil 4.17.'de görülmektedir ki bir DropDownList'ten proje seçilmekte ve içerdiği ekipman ve parçalar Şekil 4.18.'deki gibi incelenmektedir.



Şekil 4.17. İncelemek üzere proje seçme.

Şekil 4.18’de görülen ekran parça ve ekipmanlar hangi işlemlerden geçtiyse ayrıntılı olarak inceleme imkânı vermekte, malzemeler, işçilik, dış hizmetler ve ekstra maliyetlerin görülmesini sağlamaktadır.

MaterialCode	MaterialMainType	UnitPrice	Quantity	Cost
ÇL80X500	RawMaterial	2,12	157	332,84
ÇL80X130	RawMaterial	27,57	1,4	38,6
M12 x55	ReadyMadeMaterial	0,56	2	1,12
M12 x65	ReadyMadeMaterial	0,71	5	3,55
M12 x140	ReadyMadeMaterial	4,22	26	109,72
ÇL35X1000	RawMaterial	0,07	9	0,63
TotalCost				486,46
PipeCost				0
PlaqueCost				372,07
RotundateCost				0
ReadyMadeMaterial				114,39

Şekil 4.18. Proje detayı inceleme.

BÖLÜM 5

DEĞERLENDİRME

Çalışma kapsamında atölyelerde gerçekleştirilen projelerin maliyet analizini sağlayan proje tabanlı bir sistem oluşturulmuştur. Bu sistemde verilerin tamamı atölye tabanlıdır ve kullanıcılara veriye dayalı yorum yapma fırsatını verdiği için çeşitli avantajlar sağlamaktadır. Talep edilen projenin veritabanında bulunan önceki verilerle karşılaştırması ve kısa sürede müşteriye uygun bir teklif verilmesi sağlanmaktadır.

Bu amaçla geliştirme aşamasında UML diyagramları kullanılarak NYVM çıkartılmıştır ve UML kullanımının model oluşturma, gösterme ve değiştirme aşamalarında büyük kolaylıklar sağladığı görülmüştür.

Çalışmanın temeli nesneye yönelik mantıkla paralel olduğundan NYVT'nin bu bağlamda amacı gerçekleştirmekte çok uygun olduğu görülmüştür. Ayrıca NYVT için ayrı bir şema oluşturma, normalleştirme veya haritalama gerekmediğinden modelin oluşturulması ilişkisel veritabanına göre daha kısa sürmüştür. Bununla birlikte NYVT'nin gerçek dünya verilerini göstermede çok daha başarılı olduğu ve az kod kullanıldığı gözlemlenmiştir.

Çalışmamızda ilişkisel veritabanında kullanılan sabit veri tipleri dışında liste, dizi, gömülü nesnelere gibi veri tipleri kullanıldığı için NYVTYS'nin hem zengin veri tipi hem de kompleks nesnelere özelliğinden faydalanılmıştır.

Sınıf ve nitelik hiyerarşileri sayesinde kompleks nesne ilişkilerinin gösterimi ve modellenmesinin kolaylaştığı görülmüş, yine aşırı yükleme, geçersiz kılma ve geç bağlanma gibi programlama dili özellikleri kompleksliği azaltmış ve performansı artırmıştır.

Nesneler hiyerarşik olarak modellenerek kalıtım özelliğiyle birlikte metot ve kodların yeniden kullanımını sağlamış ve kullanıcıya daha az programlama çabası gerektirmiştir.

Nesneye yönelik programlama ve ilişkisel veritabanı arasında ortaya çıkan haritalama ve normalizasyon gibi problemler ortadan kalkmıştır. Çünkü veritabanı bir dış servis olmanın dışında kalıcı bir program genişletmesi olarak bulunmaktadır.

Her tablo için birincil veya yabancı anahtar oluşturulmadığından dolayı programlama süresi kısalmıştır. Nesneler gerçek dünya verilerini daha iyi modellediğinden dolayı veritabanı tasarımı ve modellemesinin daha kolay oluşturulduğu görülmüştür

NYVT'nın avantajlarıyla birlikte bazı dezavantajları da görülmüştür. Her ne kadar NYVT kullanımı her geçen gün artan bir grafiğe sahip olsa da henüz ilişkisel veritabanları kadar kullanım alanına sahip değildir. Bu sebeple NYVT satıcılarının sağladığı kaynaklara veya yardımlara rağmen gerekli bilgiyi bulmak her zaman çok kolay olmamaktadır. Bununla birlikte bu sistemlerin yaygınlaşması ile yaşanan problemlerin önüne geçilmesi muhtemeldir.

Veritabanları için kabul edilmiş standart bir sorgu dili olmadığından dolayı veritabanına özgü sorgu dillerinin öğrenilmesi pek esnek bulunmamıştır.

Kullanılan veritabanında özel amaçlı sorgu eksikliği vardır. Dolayısıyla anlık sorgular oluşturulması çok sınırlı komutlar için gerçekleştirilebilmektedir. Kullanılan veritabanında bunu gerçekleştirmek için oluşturulan ara yüz çok iyi geliştirilmiş bir ara yüz değildir.

Programlama dili bağımlılığı olması veritabanını seçerken kısıtlama oluşturan bir sebeptir. Çünkü her veritabanı her programlama diliyle gerçekleştirilememektedir.

Veritabanının kullanım oranının az olması sebebiyle yaşanan problemlerle ilgili çözümlere veya veritabanı ile ilgili dokümanlara ulaşmak çok kolay olmamaktadır.

Çalışma kapsamında atölye yönetim sisteminde kullanılacak olan karar-destek sistemi için bulanık nesneye yönelik veritabanı sunulmuştur. Veritabanında kesin olmayan verilerle çalışmak amacıyla NYVM'ne bulanık özellikler eklenerek BNYVT oluşturulmuştur. Nesne kayıt ve sorgulamaları esnasında BNYVT'nın nitelik, nesne ve sınıf bulanık ilişkileri ele alınmıştır. Bu ilişkiler sayesinde nesnelerin sınıflarına olan üyelik değerleri hesaplanmıştır.

BNYVT insan düşünce ve idrakine daha uygun olduğundan dolayı verilerle çalışabilmek için daha doğal bir yol sağlamaktadır. Benzerlik matrisinin temel alındığı bu çalışmada daha geniş aralıkta ve daha zengin sınıf tanımlamaları yapabilmek için mantıksal operatörler kullanılmıştır. Kullanılan mantıksal operatörler veri modelinin gösterimini ve sorgulamaları zenginleştirerek, veritabanının özelliklerinin genişletilmesini sağlamıştır. Bu sayede gerçek dünya verileri daha iyi bir şekilde tanımlanmış ve ifade edilmiştir.

BNYVM kullanılarak veritabanındaki hiyerarşik yapının belirlenmesi dışında ayrıca nesnelere arasında benzerlik üzerinden ilişki kurulması sağlanmıştır. Bu ilişkiler sayesinde sorgularda dilsel ifadeler kullanılarak sisteme daha sağlam ve esnek bir altyapı oluşturulmuştur. Benzerlik matrisi ve bulanık kümelerin kullanımı kesin olmayan verilerin gösteriminde resmi bir matematiksel zemin oluşturmaktadır.

Kesin kümeyle çalışmaktan farklı olarak sorgu yapılan nesnelerin birbirlerine benzerlik oranı bulunarak karşılaştırılması sağlanmıştır. Normal sorgularda sorgu değeri kesin olarak sağlayan sonuçları döndürürken, bulanık veritabanında istenen değerlerle veritabanındaki değerlerin benzerlik oranları hesaplanarak sorgu sonucu bu oranlarla birlikte getirilmektedir.

Çalışmada örnek verilerle yapılan testlere göre projenin büyüklük, istasyon sayısı ve çalışma saatleri gibi üretim parametreleri bulanık olarak tanımlanmış ve gerektiğinde sorgulamalar da bulanık olarak yapılmıştır. Bu parametreler göz önünde bulundurularak talep edilen projenin daha önceden yapılan projelerle olan maliyet benzerlikleri bulanık olarak bulunmuştur. Veritabanındaki nesnelere hem kesin hem de bulanık değerler alabilmekte, kesin ve bulanık olarak sorgulanabilmektedir.

Bunun yanında sorgulara eklenen eşik değeri bulanık benzerliğin sınırlarını da belirlemektedir. Projeye benzemesi beklenen eski projelerin benzerlik oranları istenen eşiğin üzerinde olmalıdır. Bu da veritabanında yapılan işlemlerde esneklik sağlamaktadır. Test aşamasında kullanılan örnek veriler ışığında sistemin eşik değeri tanımına göre benzer ekipmanları bulabildiği gösterilmiştir. Sistemin performansı veritabanında yer alacak olan proje verilerinin artması ile doğru orantılı olarak artacaktır.

BNYVT ve NYVT, ilişkisel veya herhangi farklı bir tip veritabanına karşı çıkmış veritabanı tipleri değildir. Kompleks ve kesin olmayan verilerin analizi söz konusu olduğunda göz önünde bulundurulması gereken seçeneklerdir. Çalışmaya uygun veritabanı seçimi için projeler daha iyi analiz edilmeli ve buna göre bir karar verilmelidir.

Sonuç olarak, imalat yapan özellikle küçük ve orta büyüklükteki mekanik atölyelerin maliyete dönük olarak ihtiyaç duydukları bir yönetim sisteminin NYVTS tabanlı bir ortamda tasarımı gerçekleştirilmiştir. Bu sistem proje, ekipman, istasyon ve çalışan bazlı farklı analizlerin yapılabilmesine imkan sağlamaktadır. Buna ilave olarak sistemin sahip olduğu veriler ışığında da tekliflendirme yapabilme imkânını sağlamaktadır.

KAYNAKLAR

1. Bolat , B. ve Tacer, S., “Ayakkabı sektöründe yer alan orta ölçekli bir firmada atölye kontrol çalışmalarına yönelik bir uygulama”, *V. Ulusal Üretim Araştırmaları Sempozyumu*, İstanbul, 25-30 (2005).
2. Zhang, Q., “Object oriented database systems in manufacturing: selection and applications”, *Industrial Management&Data Systems*, 101 (3): 97-105 (2000).
3. Hsu, C. L. and Liu, Y.S., “The development of the software for the controller of an automatic machine workshop by object-oriented programming”, *Journal of Materials Processing Technology*, 101 (1-3): 231-237 (2000).
4. Ma, Z. M., “Engineering informatipn modeling in databases: needs and constructions”, *Industrial Management&Data Systems*, 105 (7): 900-918 (2005).
5. Atkinson, M., Bancilhon, F., DeWitt, D., Dittrich, K., Maier, D. and Zdonik, S., “The object-oriented database system manifesto”, *First International Conference on Deductive and Object-Oriented Databases*, Kyoto, 223-240 (1989).
6. Jahnke, J., Schafer, W. and Ziendorf, A., “A design environment for migrating relational to object oriented database systems”, *International Conference on Software Maintenance*, California,163-170 (1996).
7. Joseph, J.V., Thatte, S.M., Thompson, C.M. and Wells, D.L., “Object-oriented databases: design and implementation”, *Proceedings of the IEEE*, 79 (1): 42-64 (1991).
8. Stajano, F, “A gentle introduction to relational and object oriented databases”, *ORL Technical Report*, Cambridge, 1-37 (1995).
9. Erdemir, H, “Nesneye yönelik veritabanı tasarımı ve bir uygulama”, Yüksek Lisans Tezi, *Beykent Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 18-26 (2008).
10. Orak, İ. M., Yaman, B. and Guerrini, G., “Designing workshop management system supporting decision making with OODB”, *Procedia Technology*, 1: 19-23 (2012).
11. Bordogna, G., Pasi, G. and Lucarella, D., “A fuzzy object-oriented data model for managing vague and uncertain information”, *International Journal of Intelligent Systems*, 14 (7): 623-651 (1999)
12. Şimşek, İ., “Veritabanı yönetim sistemlerinde bulanık mantık ile veri işleme”, Yüksek Lisans Tezi, *Marmara Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 1-10 (2009).

13. Ma, Z. M. and Yan, L., “A literature overview of fuzzy database models”, *Journal Of Information Science And Engineering*, 24 (1): 189-202 (2008)
14. Pasi,G. and Yager, R. R., “Calculating attribute values using inheritance structures in fuzzy object-oriented data models”, *IEEE Transactions On Systems, Man, And Cybernetics-Part C: Applications And Reviews*, 29 (4): 556-565 (1999).
15. İnternet: Nişancı, G., “Object Oriented Databases: The Essentials”, www.bilkent.edu.tr/~nisanci/oodbms.doc (2012).
16. İnternet: Maier, D., “Object-Oriented Database Theory”, <http://www.csd.uoc.gr/~hy562/Papers/OODBMS.pdf> (2012).
17. Stonebraker, M., Rowe, L.A., Lindsay, B. G., Gray, J., Carey, M. J., Brodie, M. J., Bernstein, P. A. And Beech, D., “Third generation database system manifesto”, *Acm Sigmod Record* , 19 (3): 31-44 (1990).
18. İnternet: Darwen, H. and Date, C. J., “The Third Manifesto”, Technical report, <http://www.acm.org/sigmod/record/issues/9503/manifesto.ps>, (2012).
19. Alam, M. and Wasan, S. K., “Migration from relational database into object oriented database”, *Journal of Computer Science*, 2 (10): 781-784 (2006).
20. İnternet: Lin, C., “Object-Oriented Database Systems: A Survey” <http://users.soe.ucsc.edu/~lcx/courses/cmcs277/cmcs277-project.pdf> (2012).
21. İnternet: Db4o “Db4o Veritabanını Kullanan Firmalar” <http://www.db4o.com/about/customers/> (2012).
22. George, R., “Uncertainty management issues in the object-oriented database model”, Doktora Tezi, *Graduate School of Tulane University*, New Orleans, 30-34 (1992).
23. Ma, Z. M. and Yan, L., “A literature overview of fuzzy database models”, *Journal Of Information Science And Engineering*, 24:189-202 (2008).
24. İnternet: Motro, A., “Imprecision And Uncertainty In Database Systems” <http://cs.gmu.edu/~ami/research/publications/pdf/fuzdbms95.pdf> (2012)
25. Gyseghem, N. V., Caluwe, R. D. and Vandenberghe, R., “UFO : Uncertainty and fuzziness in an object-oriented model”, *Second IEEE Conference on Fuzzy Systems*, California, 2: 773-778 (1993).
26. Bordogna, G., Pasi, G. and Lucarella, D., “A fuzzy object-oriented data model for managing vague and uncertain information,” *International Journal of Intelligent Systems*, 14 (7): 623-651 (1999).

27. Dubois, D., Prade, H. and Rossazza, J. P., “Vagueness, typicality, and uncertainty in class hierarchies”, *International Journal of Intelligent Systems*, 6 (2): 167-183 (1991).
28. George, R., Srikanth, R., Petry, F. E. and B. P. Buckles, “Uncertainty management issues in the object-oriented data model”, *IEEE Transactions on Fuzzy Systems*, 4 (2): 179-192 (1996).
29. Yazıcı, A. and George, R., “Fuzzy Database Modeling 1st ed”, *Physica-Verlag*, New York, 135-175 (1999).
30. Zadeh, L. A., “Similarity relations and fuzzy orderings”, *Information Sciences*, 3 (2): 177-200 (1971).
31. Özgür, N. B., Koyuncu, M. ve Yazici, A., “An intelligent fuzzy object-oriented database framework for video database applications”, *Fuzzy Sets and Systems*, 16 (15): 2253–2274 (2009).
32. Ma, Z.M., Yan, L. and Zhang, F., “Modeling fuzzy information in UML class diagrams and object-oriented database models”, *Fuzzy Sets and Systems*, 186 (1): 26-46 (2012).
33. Zhang, F., Ma, Z.M., Yan, L. and Wang, Y., “A description logic approach for representing and reasoning on fuzzy object-oriented database models”, *Fuzzy Sets and Systems*, 186 (1): 1-25 (2012).
34. Özgür, N. B., “An intelligent fuzzy object-oriented database framework for video database applications”, Yüksek Lisans Tezi, *Ortadoğu Teknik Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 25-60 (2007).

EK AÇIKLAMALAR A.

ÖRNEK MALİYET HESAPLAMA TABLOSU

Mekanik Proje Değerlendirme		İş Sıra No		1				
Mekanik Proje Maliyet Tablosu		İşin Adı		Bakır Kalıp ve Soğutma Şasesi 190'lık Slaba Göre İmalat Resimlerinin Oluşturulması Toplam 8 adet				
		İş Emri No						
		Başlama Tarihi						
		Bitme Tarihi						
		Malzeme ve İşçilik			Fatura vb.			Genel Toplamı KDV'LI
Miktar (Kg, Adet, M. Vb.)	Birim Maliyeti (YTL/.....)	Ön Toplam Maliyet	Firması	Tarih	Fatura Miktarı KDV'SİZ			
Fiili Maliyetler						75.008	95.641	
1	Malzeme (Ham ve/veya Yarımamül)		11.910			54.214	68.267	
	Plaka		0					
	1.) 100x215x1020, 321 Kal.	8	950	11.400	EMRE PAS	26 Kas	11400	13.452
	2.) 8X162X356, 304 Kalite	8	19	228	EMRE PAS	26.Kas	228	269
	3.) 8*162*450, 304 Kalite	8	23,5	282	EMRE PAS	26.Kas	282	333
	Yuvarlak		0					
	Profil		0					
	Bronz ve vb. özel Malzemler		0					
	1.) Bakır Malzeme	8	-	0	SINOSTELL	13.Oca	42303,61	42.304
	Ön İşçilik (Dış Hizmet)		0					
			0					
2	Malzeme (Hazır Malzeme ve Özel Takımlar)		1.155				992	2.162
	Hazır Malzemler		0					
	1.) Silikon 8 mm Fitol	35	2,4	84	KARDEŞLER	13.Oca	84	99
	2.) Muhtelif Malzeme	-	-	387	PROMAK	15.Oca	328,1356	328
	3.) Muhtelif Malzemeler	-	-	684	PROMAK	21.Oca	579,7458	580
	İşe Özel Takımlar		0					
			0					
3	Atelye İşçiliği		3.480				3.480	6.960
	Çatma ve Kaynak vb.	6	120	720	TEKNOPLAN	-	720	720
	Talaşlı İmalat	15	120	1.800	TEKNOPLAN	-	1.800	1.800
	Mekanik Montaj	8	120	960	TEKNOPLAN	-	960	960
	Taşıma, Maniplasyon vb.				TEKNOPLAN	-		
4	Dış Hizmet						16.322	18.251
	Isıl İşlem Krom Kaplam							
	Talaşlı İmalat							
	1.) Soğutma Şasesi İmalatı	-	-	-	OFAR MAK	-	10000	11.800
	Talaşsız İmalat							
	Taşıma Ve Nakliye							
	1.) Malzeme Nakliyesi(Çin)	8	-	-	TNT	19.Oca	4375,5	4.376
	2.) Gümrük İşlemleri	8	-	-	PIRAMİT	14.Oca	300	354
					PIRAMİT		40	40
	3.) Gümrük İşlemleri	8	-	-	ÇELEBİ A.Ş.	21.Oca	296,6	350
	4.) Gümrük İşlemleri	8	-	-	TNT	17.Oca	165	165
	5.) Gümrük Vergisi	8	-	-	GÜMRÜK	19.Oca	879,24	879
	6.) Banka Masrafı	8	-	-	ALBARAKA	14.Oca	146,02	146
	7.) Malzeme Nak(Emre Pas)	-	-	-	YÖNTEM	10.Ara	120	142
5	İş Takip İşçiliği			0				0
	İş Hazırlama Teknisyeni ve ressam			0				
	Diğerler			0				
				0				

ÖZGEÇMİŞ

Beyza YAMAN 1986 yılında Karabük'te doğdu. İstanbul Köy Hizmetleri Anadolu Lisesi'nden mezun olarak ilk ve orta öğrenimini İstanbul'da tamamladı. 2005 yılında Marmara Üniversitesi Teknik Eğitim Fakültesi Elektronik-Bilgisayar Eğitimi Bölümü'nde öğrenime başlayıp 2009 yılında mezun oldu. Aynı yıl Karabük Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Departmanı'nda araştırma görevlisi olarak göreve başladı ve Bilgisayar Mühendisliği bölümüne yüksek lisans öğrencisi olarak kabul edildi. Halen aynı bölümde çalışmalarına devam etmektedir.

ADRES BİLGİLERİ

Adres: Karabük Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü
Oda No:215-E KARABÜK

Tel: 0 370 433 20 21 / 1144
Faks: 0 370 433 32 90
E-posta: beyzayaman@karabuk.edu.tr