

**DİSİPLİNLER ARASI ÜRÜN TASARIMI İÇİN
YAPAYZEKA TABANLI BİR SİSTEM
GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Yasemin BOY

DANIŞMAN

Yrd. Doç. Dr. Ahmet GAYRETLİ

BİLGİSAYAR ANABİLİM DALI

Mayıs 2009

Bu tez çalışması Tübitak 106M226 nolu proje ile desteklenmiştir.

AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

DİSİPLİNLER ARASI ÜRÜN TASARIMI İÇİN
YAPAYZEKA TABANLI BİR SİSTEM GELİŞTİRİLMESİ

Yasemin BOY

DANIŞMAN

Yrd. Doç. Dr. Ahmet GAYRETLİ

BİLGİSAYAR ANABİLİM DALI

Mayıs 2009

ONAY SAYFASI

Yrd.Doç.Dr. Ahmet GAYRETLİ danışmanlığında,
Yasemin BOY tarafından hazırlanan
DİSİPLİNLER ARASI ÜRÜN TASARIMI İÇİN
YAPAYZEKA TABANLI BİR SİSTEM GELİŞTİRİLMESİ
başlıklı bu çalışma lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri
uyarınca
15/06/2009
tarihinde aşağıdaki jüri tarafından
Bilgisayar Anabilim Dalında
Yüksek lisans tezi olarak oybirliği/oy çokluğu ile kabul edilmiştir.

| | Ünvanı, Adı, SOYADI | İmza |
|--------|------------------------------|------|
| Başkan | Doç. Dr. Yılmaz İÇAĞA | |
| Üye | Yrd. Doç. Dr. Ahmet GAYRETLİ | |
| Üye | Yrd. Doç. Dr. Uçman ERGÜN | |

Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
.../.../2009 tarih ve
..... sayılı kararıyla onaylanmıştır.

Doç. Dr. Zehra BOZKURT
Enstitü Müdürü

İÇİNDEKİLER

| | |
|--|-------------|
| ÖZET | ii |
| ABSTRACT | iii |
| TEŞEKKÜR | iv |
| SİMGELER VE KISALTMALAR DİZİNİ | v |
| ŞEKİLLER DİZİNİ | vi |
| RESİMLER DİZİNİ | vi |
| ÇİZELGELER DİZİNİ | viii |
| | |
| 1. GİRİŞ | 1 |
| | |
| 2. GENEL BİLGİLER | 2 |
| 2.1 Yapay Zeka | 2 |
| 2.1.1 Bilgi Tabanlı Yapay Zeka ve Uzman Sistemler | 3 |
| 2.2 Uzman Sistemler | 3 |
| 2.2.1 Uzman Sistemlerin Genel Özellikleri | 4 |
| 2.2.2 Uzman Sistemlerin Geleneksel Bilgisayar Programlarından Farkları | 5 |
| 2.2.3 Uzman Sistemlerin Avantajları | 6 |
| 2.2.4 Uzman Sistemlerin Dezavantajları | 7 |
| 2.2.5 Uzman Sistemlerin Kullanımına Uygun Alanlar | 9 |
| 2.2.6 Uzman Sistemlerin Rollerini | 9 |
| 2.2.7 Uzman Sistemlerin Faydaları | 11 |
| 2.2.8 Uzman Sistemlerin Uygulanmasındaki Kısıtlar | 12 |
| 2.2.9 Uzman Sistemlerin Yapısı | 13 |
| 2.2.10 Uzman Sistemlerin Geliştirilmesi | 14 |
| 2.2.10.1 Alan Seçimi | 14 |
| 2.2.10.2 Uzmanların Seçimi | 14 |
| 2.2.10.3 Bilginin Elde Edilmesi | 15 |
| 2.2.10.4 Program Geliştirilmesi | 16 |
| 2.2.11 Uzman Sistemlerin İnsan ve Makine Yönü | 17 |

| | |
|---|-----------|
| 2.2.11.1 İnsan Yönü | 17 |
| 2.2.11.2 Makine Yönü | 17 |
| 2.3 Elektromekanik Sektörüne Bakış | 18 |
| 2.4 Sistem Analizi | 22 |
| 2.4.1 Analiz Aşamasının Adımları | 24 |
| 2.4.1.1 Keşif | 24 |
| 2.4.1.2 Kullanıcı Analizi | 25 |
| 2.4.1.3 Eğitsel Analiz | 25 |
| 2.4.1.4 Kullanılma Amaçları | 25 |
| 2.4.1.5 Teknik Analiz | 26 |
| 2.4.2 Sistem Geliştirme Döngüsü | 26 |
| 2.4.2.1 Kavramsal Analiz Safhası | 27 |
| 2.4.2.2 Detay Analiz Safhası | 28 |
| 3. MATERYAL VE METOT | 33 |
| 3.1 Uzman Sistem İnsan Yönünün Uygulanması | 33 |
| 3.1.1 Yönetim | 33 |
| 3.1.2 Uzman | 33 |
| 3.1.3 Bilgi Mühendisi | 33 |
| 3.1.4 Kullanıcı | 34 |
| 3.2 Uzman Sistem Makine Yönünün Uygulanması | 35 |
| 3.2.1 Bilgi Tabanı | 35 |
| 3.2.2 Çıkarım Mekanizması | 35 |
| 3.2.3 Kullanıcı Arayüzü | 36 |
| 4. SİSTEM TASARIMI | 38 |
| 4.1 Tasarıma Giriş | 38 |
| 4.1.1 Uzman Sistemin Alt Bölümleri | 38 |
| 4.2 Anahtar Karakteristiklerin Belirlenmesi | 39 |
| 4.3 Veritabanının Oluşturulması | 40 |
| 4.3.1 Tabloların Belirlenmesi | 40 |
| 4.3.2 Veri Girişlerinin Yapılması | 43 |

| | |
|--|-----------|
| 5. SİSTEM YAZILIMI | 44 |
| 5.1 Algoritmanın oluşturulması | 44 |
| 5.2 Tabloların Algoritmaya Uyumu | 47 |
| 5.3 Programlama Dilinin Seçilmesi | 48 |
| 5.4 Yazılımın Geliştirilmesi | 49 |
| 5.4.1 Algoritmanın Yazılıma Dönüştürülmesi | 49 |
| 5.4.1.1 Modüllerin Belirlenmesi | 50 |
| 5.4.1.2 Mekanik Kısım | 51 |
| 5.4.1.3 Etkileşim Kısım | 52 |
| 5.5 Test Aşaması | 55 |
| 5.6 Sorunların Giderilmesi | 55 |
| | |
| 6. TARTIŞMA VE SONUÇ | 57 |
| | |
| EKLER | |
| 1. Dış Kapak | x |
| 2. İç Kapak Sayfası | xi |
| 3. Onay Sayfası | xii |
| 4. Kaynaklar Bölümü | xiii |
| 5. Özgeçmiş Sayfası | xiv |
| 6. Program Kodları | xv |

ÖZET

DİSİPLİNLER ARASI ÜRÜN TASARIMI İÇİN YAPAYZEKA TABANLI BİR SİSTEM GELİŞTİRİLMESİ

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü

Bu projenin amacı, disiplinler arası ürün tasarımında karşılaşılan sorunları belirlemek, bu sorunları irdeleyerek anahtar sorular elde etmek ve sorulara alınacak cevaplar doğrultusunda tasarımın bir modelinin yazılım ortamında oluşturulmasını sağlamaktır. Böylece, kullanıcı isteklerine optimum düzeyde uyan model üretim öncesinde belirlenebilecektir. Sistemin geliştirilmesinde model olarak elektromekanik bir robot kullanılmıştır. Robotun parçaları arasındaki etkileşim incelenerek elde edilen verilerden karakteristik özellikler belirlenmiş, bu doğrultuda tasarımı belirleyecek uygun sorular çıkartılmıştır. C# programlama diliyle yazılan uygulama programı sayesinde kullanıcıya sorulan bu sorulara alınan cevaplar doğrultusunda sistemin etkileşimleri ve anahtar özellikleri belirlenmektedir. Sonuçta elde edilen veriler öneri ve uyarı olarak kullanıcıya iletilmektedir.

Sistem başlangıçta belirlenen sınırlı sayıda soruya göre oluşturulmakla beraber yapılan geliştirmeler sonucunda farklı sorular eklenebilir hale getirilmiştir. Ayrıca, yalnızca model olarak seçilen robotla sınırlı kalınmayıp herhangi bir elektromekanik sistem tasarımında kullanılabilir şekilde geliştirme yapılması da mümkündür.

2009, 57 sayfa

Anahtar Kelimeler: Elektromekanik ürün, Sistem tasarımı, Disiplinler arası ürün tasarımı

ABSTRACT

AN ARTIFICIAL INTELLIGENCE SYSTEM FOR INTERDISCIPLINARY PRODUCT DESIGN

Afyon Kocatepe University,
Institute for the Natural and Applied Sciences

The aim of this project is to find out problems that may occur during desing of interdisciplinary products, get key questions by evaluating these problems and provide forming out a model of product in software environment via answers. So, the optimum model adapted to user's expectation can be derived before production. An electromechanical robot is used as model in system development phase. Characteristic properties are determined from data that was collected by evaluating attractions between parts of robot. These properties are used for appropriate questions which determine the design. Attractions in system and characteristic properties of system can be defined via user's answers of these questions by software written in C#. System outputs are submitted to user as warnings and suggestions.

The system is formed with limited questions, however developments are done to add different questions. Also, there is no restrirction to use only the robot as model, but it can be developed to implement any electromechanical system design.

2009, 57 sayfa

Keywords: Electromechanical product, System design, Interdisciplinary product design

TEŐEKKÜR

Tez alıőmam sűresince bilgi ve deneyimleri ile bana destek olan tez danıőmanım ve deęerli hocam Yrd. Do. Dr. Ahmet GAYRETLİ'ye teőekkűrlerimi sunarım.

Tez alıőmamda buldukları katkılardan dolayı Sn. Do. Dr. Mevlűt DOĐAN'a, Őęr. Gűr. Mahmut KANTAR'a, Őęr. Gűr. Mustafa NARTKAYA'ya, Arő. Grv. Őzgűr VERİM'e ve tűm bűlűm arkadaőlarım teőekkűrlerimi bor bilirim.

Yasemin BOY

AFYONKARAHİSAR, Mayıs 2009

SİMGELER VE KISALTMALAR DİZİNİ

1. Simgeler

2. Kısaltmalar

| | |
|------|-------------------------------|
| SDLC | System Development Life Cycle |
| JAD | Joint Application Development |
| I/O | Input Output |
| IS | Information Systems |
| S | Soru |
| C | Cevap |
| U | Uyarı |
| O | Öneri |
| Z | Seçmeli soru |
| P | Problem yok |

ŞEKİLLER DİZİNİ

| | <u>Sayfa No</u> |
|---|-----------------|
| Şekil 2.1 Uzman sistemler ve geleneksel bilgisayar programları arasındaki genel farklar | 6 |
| Şekil 2.2 Bir Uzman Sistemin Yapısı | 13 |
| Şekil 2.3 Uzman Sistemlerin İnsan Yönü | 17 |
| Şekil 2.4 Uzman Sistemlerin Makine Yönü | 18 |
| Şekil 2.5 Analiz Süreci Adımları | 31 |
| Şekil 3.1 Etkileşim Algoritması | 34 |
| Şekil 3.2 Mekanik Kısım Kuralları ve Sonuçları | 35 |
| Şekil 3.3 Etkileşim Kısmı Kural ve Sonuçları | 36 |
| Şekil 5.1 Akış Şeması | 44 |
| Şekil 5.2 Algoritma | 45 |
| Şekil 5.3 Etkileşim Algoritması Sonucu | 46 |

RESİMLER DİZİNİ

| | <u>Sayfa No</u> |
|--|-----------------|
| Resim 3.1 Soru Ekleme Arayüzü | 37 |
| Resim 4.1 Model Seçilen Robotun Parçaları | 40 |
| Resim 4.2 Motor Tablosu | 41 |
| Resim 4.3 Soru Tablosu | 42 |
| Resim 5.1 Etkileşim Tablosu | 48 |
| Resim 5.2 Program Giriş Sayfası | 50 |
| Resim 5.3 Mekanik Kısım | 51 |
| Resim 5.4 Mekanik Kısım Kriterlerin Belirlenmesi | 52 |
| Resim 5.5 Etkileşim Modülü | 53 |
| Resim 5.6 Etkileşim Modülü Soru Örneği | 54 |

ÇİZELGELER DİZİNİ

| | <u>Sayfa No</u> |
|--|-----------------|
| Çizelge 2.1 Elektromekanik Bir Ürünün Üretim Sürecinde Yaşanan Uyumsuzluk Problemi | 20 |

1. GİRİŞ

Farklı disiplinlerin bir arada uyumla çalışmasını gerektiren elektromekanik sistemlerin tasarlanması oldukça karmaşık algoritmalar gerektirmektedir. Her bir disiplin bazında ayrı ayrı yapılacak analiz, tasarım, uygulama ve test süreçleri tasarımcıların büyük zamanını almaktadır. Ayrıca ürünün analizden üretime kadar geçen bütün süreçlerinde çalışan değişik alanlardan uzmanlar uyum içinde çalışmak zorunda kalmakta, yönetsel açıklarda ise tasarımda en başa dönüldüğünden zaman kaybı artmaktadır. Başlangıç olarak projede kullanılan yöntemler olan yapay zeka, uzman sistemler, analiz süreci açıklanmış, daha sonra sistemin yazılım ve geliştirme süreci ele alınmıştır. Sayılarla hesap yapmaktan yerine kavramlarla muhakeme yapan yapay zeka kavramı ile bilgi tabanlı sistemindeki kuralları kullanarak belirli sorunları çözen bilgi tabanlı uzman sistemlerden faydalanılarak elektromekanik sektöründe kullanılacak, tasarım ve sorun çözmeye dayalı bir uzman sistem geliştirilmeye çalışılmıştır.

2. GENEL BİLGİLER

2.1 Yapay Zeka

Yapay Zeka terimi resmen ilk defa 1950'lerin ortasında kullanılmıştır. Yapay Zeka, başka türlü otomatikleştirilemeyecek zeka süreçlerini modellemek ve makineleştirmek için yöntemler sağlamaya gayret eder. Yapay Zeka alanında çalışmak; geleneksel bilgisayar yaklaşımlarına ilaveten; çoğunlukla sembolik, algoritmik olmayan problem çözme metotları ve bilgisayarların, sayılarla hesap yapmaktan ziyade kavramlarla muhakeme yapmak için kullanımına dayanır.

Yapay Zeka bir bilgisayar bilimi alanıdır. Görüşleri; psikoloji, bilmeye ve kavramaya ait bilim, hesaplamalı dilbilim, veri işleme, karar destek sistemleri ve hesaplamalı modelleme gibi çeşitli diğer disiplinlere yakından bağlıdır ve tüm bu disiplinleri kaynak olarak kullanır.

Yapay zeka, insanın düşünme yapısını anlamak ve bunun benzerini ortaya çıkaracak bilgisayar işlemlerini geliştirmeye çalışmak olarak da tanımlanabilir. Yani programlanmış bir bilgisayarın düşünme girişimidir. Daha geniş bir tanıma göre ise, yapay zeka, bilgi edinme, algılama, görme, düşünme ve karar verme gibi insan zekasına özgü kapasitelerle donatılmış bilgisayarlardır.

Geliştirilen programların gerçek sorunlarla karşılaştığında çok kötü bir performans göstermesinin ardındaki temel neden, bu programların yalnızca yapay bir şekilde çalışıp konu ile ilgili bilgileri kullanmamasıydı. Bu dönemin en ünlü programlarından Weizenbaum tarafından geliştirilen Eliza, karşısındaki ile sohbet edebiliyor gibi görünmesine karşın, yalnızca karşısındaki insanın cümleleri üzerinde bazı işlemler yapıyordu. İlk makine çevirisi çalışmaları sırasında benzeri yaklaşımlar kullanılıp çok gülünç çevirilerle karşılaşıncaya bu çalışmaların desteklenmesi durdurulmuştur.

Her sorunu çözecek genel amaçlı program yerine belirli bir uzmanlık alanındaki bilgiyle donatılmış programlar kullanma fikri, yapay zeka alanında yeniden bir canlanmaya yol açtı. Kısa sürede "Uzman Sistemler" adı verilen bir metodoloji gelişti. Fakat burada çok sık rastlanan tipik bir durum, bir otomobilin tamiri için önerilerde bulunan uzman sistem programının otomobilin ne işe yaradığından haberi olmamasıydı.

Bütün bu çalışmaların sonunda yapay zeka arařtırmacıları iki gruba ayrıldılar. Bir grup, insan gibi düşünen sistemler yapmak için çalışırken, diđer grup ise rasyonel karar verebilen sistemler üretmeyi amaçlamaktaydı.

2.1.1 Bilgi Tabanlı Yapay Zeka ve Uzman Sistemler

Bilgi tabanlı yapay zeka sistemi, belli bir uygulama alanına (bilgisayar onarımı gibi) ilişkin pratik çözüm veya yordamlama bilgilerinden (sezgi, yargı ve çıkarımlar) oluşmuş bir bilgi tabanına dayalı olarak çalışır. İnsanların kendilerine ait bilgi tabanı sistemindeki EĞER- 0 ZAMAN (IF-THEN) kurallarını kullanarak belirli sorunları çözme kabiliyeti, bu yapay zeka türüne ilham kaynağı olmuştur. Bilgi tabanlı sistemlerin en gelişmiş örneđi uzman sistemlerdir. Belli bir soruna ilişkin uzmanlık bilgileri bir uzman sistemin bilgi tabanına yerleştirildikten sonra kullanıcıların bu bilgiden yararlanmak amacıyla uzman sistemle kurduđu iletişim bir uzman şahısla kurulan iletişimin bir benzeri olacaktır. Sorun çözülene kadar kullanıcı ile bilgisayar tabanlı uzman sistem arasında karşılıklı soru-cevap türünde bir iletişim oluşur.

2.2 Uzman Sistemler

Günümüzün bilgisayarları muazzam işlem yapma kapasitesine sahip olmasına karşın, ne yazık ki öğrenme becerisine sahip değildir. Yapay zeka arařtırmalarının bir kolu olan bilgi tabanlı sistemler bu olguyu değiřtirmeye çalışmaktadır. Uygulamada, "uzman sistemler" ve "bilgi-tabanlı sistemler" aynı anlamda kullanılan terimlerdir. Teknik açıdan bakıldığında ise, uzman sistem bir bilgi-tabanlı sistemin en gelişmiş biçimidir. Bir uzman sistem sorulara cevap veren, açıklık getirmek için soru soran, tavsiyelerde bulunan ve karar verme sürecine yardımcı olan diyaloga açık bir sistemdir. Daha az gelişmiş bilgi-tabanlı sistemlere ise yardımcı sistemler denilmektedir. Yardımcı sistem, kullanıcının göreceli olarak basit nitelikteki kararları vermesine yardımcı olan bir sistemdir. Yardımcı sistemler nihai kullanıcının belirli bir sorunu çözmekten ziyade muhakeme sürecinde yapabileceđi bir hata olasılıđını azaltma amacını gütmektedir.

Uzman sistemler insan düşünce sürecini taklit etmeye çalışır, muhakeme edebilir, çıkarımda ve yargıda bulunabilir. Günümüzde uzman sistemler değişik bilim dallarında karar vermeye yardımcı olarak kullanılmaktadır. Örneđin, tıbbi teşhiste, petrol arařtırmasında, finansal planlamada, vergi hesaplamada, kimyasal analizde,

cerrahide, lokomotif onarımında, hava tahmininde, bilgisayar tamiratında, uydu onarımında, bilgisayar sistemlerinin tasarımında, nükleer santrallerin işletilmesinde, devlet yasalarını yorumlamada ve daha nice alanlarda etkin bir biçimde kullanılmaktadır.

Bir uzman sistem, uzmanlık ve deneyim kullanımına ihtiyaç duyan ve zor problemleri yüksek bir yeterlilik seviyesinde çözebilen, yalnızca belli bir alanla ilgili geniş tabanlı bir bilgiye sahip ileri bir bilgisayar programıdır. Bunu; teknik bilgiler, bilgi, sezgisel yöntemler ve bu tür problemleri çözmek için uzmanlar tarafından kullanılan problem çözme süreçlerini kullanarak başarır. Uzman sistemler, böylece; insan bilgisi, uzmanlığı ve deneyiminin bilgisayarlarda depolanmasını sağlar.

2.2.1 Uzman Sistemlerin Genel Özellikleri

Başarılı bir uzman sistemin dört temel özelliği vardır. Bunlar:

a. Önemli Performans Yeteneği

Uzman sistemlerin başarılı bir şekilde gerçekleştirilmesi ve kullanılması için en az, ilgili konudaki uzman kadar performans göstermesi gerekir. Aksi halde sistemi meydana getirmek gereksizdir. Hatta sistemin uzmandan “daha iyi” performans göstermesi istenecektir. Ancak uzman “kadar” performans gösteren sistemin geniş bir kullanıcı kitlesi arasında yaygınlaşmasında elde edilecek çeşitli faydalar bulunmaktadır.

b. Makul Cevap Süresi

Bu özellik bir önceki ile yakından ilgilidir. Sistem, sorularla cevapları sağlama sırasında makul olmayan bir zaman harcarsa kullanılmaz ve son kullanıcılar uzman tavsiyesi için uzmanın kendisine danışma uygulamasına dönerler. Standart bir bilgisayar sistemi dizayn edildiğinde cevap süreleri için kriter elde edilmelidir.

c. Değişebilirlik

Uzman her zaman değişir. Yeni gerçekler, teknikler ve metotlar keşfedilmekte, yeni gerçekler ve kurallar gün ışığına çıkmaktadır. Bir uzman sistem, uzmanların bilgilerini değiştirebilmeleri- ile benzer ölçüde esnekliği kullanıcılara sağlamak zorundadırlar.

d. Aktif Kullanıcı Arayüzü

Çok iyi bir bilgi tabanına sahip; ancak zayıf bir kullanıcı arayüzü ile sunulan bir sistem kullanılmayacaktır. Bununla birlikte küçük bir bilgi tabanı ve iyi bir arayüzden oluşan sistem çok faydalı olabilir.

2.2.2 Uzman Sistemlerin Geleneksel Bilgisayar Programlarından Farkları

Uzman sistemler, kendilerini geleneksel bilgisayar programlarından ayıran çeşitli kabiliyetlere ve özelliklere sahiptirler. Bir uzman sistemin temel hedefi, uzmanların sezgisel uzmanlıklarına sahip olmak ve bunu yaygınlaştırmaktır; geleneksel bir programın hedefi ise bir algoritma setini tamamlamaktır.

Uzman sistemler, geleneksel yaklaşımlarla ele alınmasının çok zor veya imkansız olduğu düşünülen durumlarda kullanılabilir. Ayrıca uzman sistemler, ihtiyaç duyulan bilginin eksik veya ulaşılabilen bilginin tutarsız olduğu durumları ele alma kabiliyetine sahiptir. Uzman sistem teknolojisindeki teknikler belirsiz durumları ve belirsiz verileri tahlil ederler ve bazı uzman sistemler yeni bilgilere ulaşıldığında yargılarını değiştirebilirler. Bunlara ilaveten, çoğu kez bir uzman sistem aldığı kararlara veya neden belirli bir operasyonu gerçekleştirdiğine ya da belirli bir bilgiyi sorduğuna açıklama da getirebilir.

Uzman sistemler öncelikle, sayısal hesaplamalardan çok kavramlar hakkında sembolik muhakemeye dayanırlar. Bu sistemler çoğunlukla işleme ait yaklaşımlardan çok açıklayıcı yaklaşımlar kullanılarak programlanırlar ve programlama teknikleri, program kontrolünün alan bilgisinden geniş ölçüde ayrılmasına müsaade eder. Program kontrolünden ayrılmış açıklayıcı bilginin kullanımı, çoğu kez uzman sistemlerin, geleneksel programlara göre daha esnek ve tekrar gözden geçirilip düzeltilme ve güncellenmesinin daha kolay olmasını sağlamaktadır.

Geleneksel bir programın bu yeteneklere sahip olması çok zordur. Şekil 2.1 uzman sistemlerle geleneksel bilgisayar programlarını karşılaştırmaktadır.

| Uzman Sistem | Geleneksel Program |
|--|------------------------------|
| Kararlar alır | Sonuçlar hesaplar |
| Sezgisel yöntemlere dayanır | Algoritmalara dayanır |
| Daha esnek | Daha az esnek |
| Belirsizliği ele alabilir | Belirsizliği ele alamaz |
| Kısmi bilgi, tutarsızlıklar ve kısmi kanaatlerle çalışabilir | Komple bilgiye ihtiyaç duyar |
| Sonuçların açıklamalarını sağlayabilir | Sonuçları açıklamasız verir |
| Sembolik muhakeme | Sayısal hesaplamalar |
| Öncelikle açıklayıcı | Öncelikle işleme ait |
| Kontrol ve bilgi ayrılmış | Kontrol ve bilgi iç içe |

Şekil 2.1 Uzman sistemler ve geleneksel bilgisayar programları arasındaki genel farklar

2.2.3 Uzman Sistemlerin Avantajları

- **Elde Edilebilirlik:** Eğer uzmanın bilgisi bir dosyada saklanırsa bu bilgiyi organizasyon içerisinde geniş bir şekilde dağıtmak mümkündür. İhtiyaç duyan her bölüm, bu dosyanın bir kopyasını alabilir ve bu kopyaya her zaman ulaşılabilir. İnsan uzmanların aksine bilgisayar programları, sinirlenmez, yorulmaz, tatile çıkmaz, hasta olmaz ve ölmezler. Bilgiler daima kullanıma hazırdır.
- **Tutarlılık :** En iyi uzman bile yanlışlar yapabilir ya da önemli bir noktayı unutabilir. İyi bir uzman sistemle yanlışlar daha az meydana gelecektir. Bir bilgisayar programı tutarlıdır. Doğru olarak programlanması durumunda daima doğru sonuçlar verecektir.
- **Geniş Kapsamlılık:** Birden fazla uzmanın ortak fikrini almak çok zordur ve bir grup uzmanı bir konuda tartışıp, ortak bir yargıya varmak üzere bir araya getirmek neredeyse imkânsızdır. Bir uzman yalnızca kendi bilgi ve deneyimlerini kullanabilir. Bir bilgisayar sistemi ile, bir uzman sistem birden fazla uzmanın bilgisini içerebilir. Böylece uzman sistem tarafından verilen karar,

en az diđer katılımcıların vereceđi karar kadar iyi olacaktır. Ayrıca uzman sistemler birbirleriyle görüř alıřveriřinde bulunabilirler ve çeřitli seęenekler sunabilirler.

2.2.4 Uzman Sistemlerin Dezavantajları

- **Alan Seęimi:** Uzman sistem uygulamalarının bazıları iyi ęalıřırken diđerleri ęalıřmaz.

Bu sebeple belirli bir alanda belirli bir teknoloji kullanma ya da kullanmama kararı bilgisayar tabanlı sistemlerde önemlidir. Bazı problemler uzman sistemler için fazla karmařık olabilir. Eđer uzmanlar hemfikir deđilse veya o alandaki uzman müsait deđilse bu durumda o alan uygun deđildir. Benzer řekilde ęözmesi ok uzun zaman alan, bir ok etkileřimin olduđu ya da uzaysal (spatial) iliřkilere ve prosedürlere ok fazla bađımlılıđın bulunduđu problemler de uzman sistemler için uygun deđildir.

- **Test Etme:** Uzman sistemleri test etmek, özellikle, basite tavsiye vermekten ok karar almaya yardımcı olmak için kullanılan sistemlerde kritik bir ihtiyatır. Bir uzman sistemin test edilmesinde pek ok sorunla karřılařılabilir. Programı geliřtirenler, sistemin nasıl davranması gerektiđi konusunda her zaman emin deđildirler ve bu yüzden sistemi tam anlamıyla test edemezler. Biz uzman sistem programının izleyeceđi yolu tanımlamak kolay deđildir. Bu durum ok fazla risk ieren uygulamalarda ciddi bir problemdir (örneđin tıbbi uygulamalar). Sistemin büyüklüđu arttıka kontrol ve bakım iřlemlerinin zorluđu da artar. Kullanıcı, sistem tarafından önerilen tavsiyenin en uygun tavsiye olduđundan emin olmak ve olasılık dahilindeki tüm yan etkileri hesaba katmak zorundadır. Sistemin dođruluđunun ve tamlıđının test edilmesi ok zordur. Bununla birlikte, nihayet bu bir sorumluluk sorunudur: Kim sorumlu? Uzman sistemi kullanan, yazan, bilgisi uzman sistemde kullanılan ya da bu sistemin kullanım hakkına sahip olan kiři mi?
- **Belirsizlik:** Gerek dünyada sadece dođru ve yanlıř gereklerle ilgilenmeyiz. Genellikle sadece belli bir derece emin oluruz (kađıtları evde bıraktıđımızda %80 emin oluruz gibi). İdeal olarak bir uzman sistem, belirsizliđin üstesinden

gelebilmelidir. Fakat, uzman sistemin sunduđu mantıksal muhakemenin kullanımı için geerli istatistiksel kurallar bulmak byk bir teorik sorundur.

- **Sınırlamalar:** Uzman sınırlarını bilir; bilgisayar tabanlı bir sistem eđer zellikle programlanmamıřsa bilemez. Bu bakımdan, uzman sistemler pek o kadar iyi sonu vermezler. Her zaman bir cevap retmeye yneliktirler. Bu durum problem olabilir ve uzman sistemlerin uzmanlara ikame etmek yerine yardımcı bir ara olarak kullanılmalarının daha uygun olacađının altı izilmelidir.
- **Kabul Edilebilirlik:** Uzman sistem teknolojisine giriř, firma ierisinde, firmanın organizasyonel yapısı iin nemli sonular dođurur. Herkes bir bilgisayara gvenmek ya da kullanmak istemeyebilir. Bazı insanlar bilgisayar kullanımına diren gsterirler ve uzmanlarla alıřmayı tercih ederler. Uzmanlar bile bazen uzman sistemler ile ilgili řphecidir. Hatta sistemler iyi alıřtıđı ve uzmanlarla hemfikir olduđu zamanlarda bile kendileriyle aynı tip muhakeme yntemini kullanmadıklarını dřndkleri iin uzman sistemlere gvenmezler. Bu faktrler uzman sistem kullanımı dřnldđnde gz nnde bulundurulmalıdır. Aksi halde sistem, firma iinde aktif bir muhalefet ile karřılařabilir.
- **Bilgi Edinme:** Uzman sistem geliřtirmek iin řimdiye kadar karřılařılan en ciddi sorun, bilgi edinmenin yol atıđı “dar bođaz (bottleneck)”dır. Uzmanın bilgisini saklama ve ortaya ıkarma prosesi ođunlukla uzun ve yavař bir prosestir.
- **Gncelleme:** Bilgilerin sık sık deđiřtiđi alanlar uzman sistemlerin geliřtirilmesine pek uygun deđildir. Uzman sistemin uzmanlıđını kaybetmemesi iin bilgi tabanını srekli gncellemesi gerekir. Bilgi tabanının gncellenmesi iin gerekli kořullar tesis edilmelidir.
- **Davranıř :** Uzman sistemlerin amacı, insan uzmanları taklit etmek olmasına rađmen bunu yapabilen ok az sistem vardır. Sistemle kullanıcı arasındaki diyaloglar genellikle program tarafından ynlendirilir ve genellikle tanımlamaları anlamak zor olabilir. Konsltasyonlar kullanıcının deđil bilgisayar programının gdmnde olma eđilimindedir ve kullanıcı cevabı elde edebilmek iin sık sık gereksiz aıklamalarla uđrařmak zorunda kalmaktadır.

2.2.5 Uzman Sistemlerin Kullanımına Uygun Alanlar

Uzman sistemler ancak gerekli olduğu zamanlarda kullanılmalıdır. Bazı işletmelerin bulunduğu koşullar, uzman sistemin kullanılmasını gerek maliyet, gerekse sağlanan faydanın önemsiz olması yüzünden haklı kılmayabilir. Uzman sistemlerden aşağıdaki koşullar altında yararlanılması tavsiye edilmektedir:

- İşin tekdüzeliğine bağlı olarak uzman sistemin sık sık kullanılmasına gereksinim duyuluyorsa ve kullanıcı sayısı uzman sistemin kullanılmasını maliyet boyutunda ekonomik kılacak kadar fazla sayıda ise,
- Karar verme durumu karmaşıksa (basit durumlar için basit bir bilgisayar programından da yararlanılabilir.)
- Karar verme mantığı bir kural hiyerarşisine dönüştürülebiliyorsa
- Gözden geçirmenin çok uzun zaman alacağı, çok sayıda mümkün kombinasyonun olduğu problemlerde,
- Çok miktarda dikkate değer veriyi anlamlandırmanın ya da bilgilerin geri çağırılması (retrieval) işlemlerinin yapılmasının gerektiği durumlarda.
- Uygulama öneri, sınıflama, teşhis, yorum, açıklama, çözüm yolu seçme, durumu değerlendirme ve tahmin etme üzerinde yoğunlaşıyorsa.

2.2.6 Uzman Sistemlerin Roller

Bir uzman sistem, aşağıda verilen olası uzman rollerinin bir veya birden fazlasını gerçekleştirebilir.

- **Otonom Uzman:** Bir uzman sistem bir durumu analiz eder, bir karar alır ve daha sonra doğrudan doğruya ona göre davranır veya bir insan faaliyetini kumanda eder. Tüm bu aşamalarda bir insan katılımı söz konusu değildir. Nadir veya olağandışı durumlara uygun bir şekilde tepki gösterir. Örneğin, otonom bir uzmanın rolünde, bir uzman sistem; alıcı aletleri okuyarak, sıcaklık ve basınç gibi kontrol edilebilir değişkenlerde gerekli düzeltmeleri yapar ve doğrudan doğruya bir kimyasal süreci başlatabilir. Veya bir teknisyeni, sistem tarafından hatalı olduğu belirlenmiş bir makine parçasını değiştirmeye yönlenebilir.
- **İnsan Katımlı Otonom Uzman:** Uzman sistem, otonom bir uzman gibi davranır fakat aldığı kararları, bu kararları dikkate almayacak veya değiştirebilecek olan bir

uzmana sunar. Bu rolde, bir uzman sistem, uygulamadan önce bir tasarımcı tarafından gözden geçirilecek olan, kompleks bir tasarımı formüle edebilir.

- **Uzman Danışman:** Bir uzman sistem, bir görevi yerine getiren kişi için, uzman danışmanların yaptığı gibi, özellikle zor veya olağandışı durumlar için tavsiyelerde bulunan, uzman seviyede bir danışmanlık sağlar.
- **Meslektaş:** Uzman sistem, kendi sahasındaki insanlara önerilerde bulunur. Her ne kadar insandan daha yüksek ustalık seviyesine sahip olamasa da, insanın yapmaya zaman bulamayacağı analizleri yapabilir veya insan, faaliyetin en muhtemel seçeneğini elde etmeye çalışırken; uzman sistem, çok sayıda karmaşık alternatifi araştırabilir. Örneğin, aksaklıkları saptayıp çözümleyen bir uzman sistem, hataların muhtemel olmayan sebeplerini inceleyebilir. Oysa, aksaklıkları saptayıp çözümleyen insanlar, bu muhtemel olmayan sebeplerden birinin gerçekleşmesi durumunda, analizler için gerekli olan zamanın zararı tazmin edilemediğinden bunlarla uğraşmazlar.
- **Zeki Yardımcı:** Uzman sistem, uygulayıcının zeki bir yardımcısı gibidir. Belli bir durum meydana geldiğinde, bir dizi makul öneride bulunabilir veya dikkate alınması gereken bir dizi madde belirleyebilir. Uygulayıcının ilgileneceği şekilde değişkenlerin belli kombinasyonlarını araştırarak verileri gözleyebilir. Örneğin, zeki bir yardımcı rolünde, bir uzman sistem, bir fabrikada alıcı aletleri gözleyebilir ve ölçülen değerlerin bazı kombinasyonlarının problem yaratacağını saptadığında bir gösterge paneli aracılığıyla işaret verebilir.
- **Düşük – Seviye Zeki Yardımcı:** Uzman sistem, uygulayıcının vaktini işin daha zor kısımlarına ayırabilmesini sağlamak amacıyla nispeten düşük seviyede ve zaman alan görevleri yerine getirir. Örnek olarak, bir uzman sistem, büyük miktarlardaki ham veriyi, uzmanın incelemesini kolaylaştıracak şekilde daha küçük miktarlarda kısmi işlenmiş veriye dönüştürerek karmaşık bir uzman analizinin ilk adımını gerçekleştirebilir.

2.2.7 Uzman Sistemlerin Faydaları

Uzman sistem teknolojisi; uzmanlığın korunmasının veya uzmanlığa ulaşım ve uzmanlığın dağılım alanının genişletilmesinin değerli olduğu durumlar için düşünülebilir. Uzman sistemler aşağıdaki durumlarda uygun (ve çoğu durumda başarılı olabilecek tek teknoloji) olabilir:

- Uzmanlığın nadir olduğu durumlarda uzmanlık sağlamak.
- Uzmanlığı elde etmenin pahalı olduğu durumlarda uzmanlık sağlamak.
- Uzmanlara ulaşulamayan zamanlarda uzmanlık sağlamak.
- Uzmanlara ulaşulamayan mekanlarda uzmanlık sağlamak.
- Bir uzman analizinin alacağı zamanı kısaltmak.
- Daha az deneyimli ve daha az usta personelin performans derecesini yükseltmek.
- Deneyimli ve usta personelin performans derecesini arttırmak, böylece en tepedeki uzmanın seviyesine daha yakın bir performans göstermelerini sağlamak.
- Uzmanlık analizlerini uzmanların zaman bulamadığı durumlara uygulamak (bir karar vermeden önce büyük miktarda veriyi analiz etmek veya herbir olasılığı –en olmayacak olanı bile- göz önünde bulundurmak gibi).
- Uzmanlığı; yeniden atama, emeklilik veya başka personel kaybı yoluyla kaybedebileceği durumlarda bilgiyi ele geçirmek ve saklamak.
- Uzmanlar uzmanlıklarının en tepesindeyken, ileride aynı yoğunlukta çalışmadıklarında mevcut hünelerinin bir kısmını kaybedebilecekleri ihtimalini düşünerek bilgiyi ele geçirmek ve saklamak.
- Uzmanlar hastalık veya tatil nedeniyle bulunmadıklarında uzmanlığa ulaşım sağlamak.
- Uzmanlar düşük performans seviyelerinde olduklarında uzmanlığa ulaşım sağlamak.
- Sürekli personel değişiminin sözkonusu olduğu durumlarda, yeni personelin sürekli olarak eğitilmesi ihtiyacını en aza indirmek veya ortadan kaldırmak.
- En iyi uzmanların bile zaman zaman yapabilecekleri, imla hataları ve diğer küçük hataları en aza indirmek veya ortadan kaldırmak.

- Otomatikleşmiş uzmanlığı, diğer bağlı görevlerin otomatikleştirilmesi için bir temel olarak kullanarak faydalı kılmak, böylece geniş bir görevin komple otomasyonuna izin vermek.
- Uzmanlığın, her seferinde; aynı tarzda, tarafsız ve devamlı olarak uygulanmasını sağlamak, böylece, örneğin şirket politikasını kararlı hale getirmek.
- En iyi uzmanların zamanını en zor problemler veya diğer önemli görevler için boşaltmak.
- Bir ürün veya servis olarak (veya bir ürün veya servisin parçası olarak) pazarlanabilecek bir şekilde uzmanlığı otomatikleştirmek.

Böylece uzman sistemler; performans ve kaliteyi yükseltmek, verimliliği arttırmak, kararlılık sağlamak, önemli bilgiyi korumak, insan kaynaklarının daha iyi kullanılmasını temin etmek ve yeni veya daha iyi ürünlerin pazarlanmasına imkan vermek için uzmanlık bilgisi sağlarlar.

2.2.8 Uzman Sistemlerin Uygulanmasındaki Kısıtlar

Her ne kadar uzman sistemler, bazı durumlarda en iyi uzmanlardan daha iyi performans gösterebilir ve her ne kadar uzman sistem teknolojisi, geleneksel programlama teknikleriyle çözümü aşırı derecede zor olan (eğer tamamen imkansız değilse) problemlerin çözümünde kullanılabilirse de, bu sistemler her derde deva olarak düşünülmemelidir. Hiçbir teknoloji bütün problemleri çözemez ve bu yüzden uzman sistemlerin kısıtları bilinmelidir.

Herhangi bir alanda, bir uzman sistemin önde gelen bir uzmanın komple uzmanlığına ve deneyimine sahip olması ve insan bilgisine dayanıyor olması nedeniyle, bu sistemlerin, uzmanların çözemediği problemleri çözmesi mümkün değildir (vaktin yetmediği ve bilgisayarın sahip olduğu kaynaklara ulaşamama nedeniyle çözülemeyen problemler hariç).

Bir uzman sistem geliştirmek sadece dar ve sınırları iyi belirlenmiş bir alanda mümkündür. Mevcut uzman sistemler çoğunlukla sınırlarında zayıf performans gösterirler ve bu sistemlerin çoğu bir durumun kendi faaliyet alanlarına girip girmediğini belirlemede güçlük çekerler. Mevcut ticari uzman sistemler önemli

öğrenme yeteneklerine sahip değillerdir ve sadece birkaçı, sezgisel yöntemler işe yaramadığında faaliyet alanının temel prensiplerinin kullanımına imkan tanırırlar.

2.2.9 Uzman Sistemlerin Yapısı

Uzman sistemlerin temel şeması Şekil 2.2’de gösterilmiştir. Uzman sistemi oluşturan ve bir bütün şeklinde çalışarak insan ve makine yönlerini düzenli biçimde birleştiren alt sistemler kısaca açıklanmıştır.



Şekil 2.2 Bir Uzman Sistemin Yapısı

Bilgi Tabanı, çözülmesi hedeflenen problem ile ilgili bilgiyi içerir. Bu kısım alan bilgisi ve hafıza olarak ikiye ayrılabilir. Alan bilgisi problem çözümünün yapılabilmesi için gerekli kuralları (rules) içerir, hafıza ise bilgi tabanındaki kuralların işleme girmesi için gerekli gerçeklerden oluşur (facts).

Sonuç Çıkarma Mekanizması, gerçeklere bakarak hangi kuralın sağlandığına karar verir ve hafızadaki gerçeklere göre bir adım sonra hangi kuralın işleme gireceğini belirler.

Açıklama, tüm US’ler yaptığı işi açıklamalıdır. Hangi kural ve gerçeklere dayanarak sonuca vardığını göstermelidir.

Ara Yüzey, kullanıcı ile programın iletişimini sağlar.

Uzman, bir US’in yazılabilmesi için en az bir o problem alanının uzmanı ve bir de programlamayı yapacak bilgi uzmanı olmalıdır.

Kullanıcı, genelde problem alanını bilen biridir, bilgi uzmanı olması şart değildir.

2.2.10 Uzman Sistemlerin Geliştirilmesi

Uzman sistem geliştirmede sabit kurallar yoktur, birden fazla yaklaşımın kullanılabilceği birçok durum vardır ve her durumda işe yarayan birkaç standart teknik vardır. Bu yüzden burada, bir uzman sistemin geliştirilmesinde takip edilecek adımlar tanımlanırken, her adımda, uygulamada başarılı olmuş, ispatlanmış teknikler üzerinde durulacaktır.

Uzman sistem geliştirmenin öncelikli hedefi, seçilen alandaki bir uzmandan, alan içindeki bir görev deneyiminden kazanmış olduğu bilgiyi elde etmek ve daha sonra bu bilgiyi bir uzman sistem programında kullanmaktır.

En temel haliyle, uzman sistem geliştirme dört temel adımdan oluşur:

- a. Alan Seçimi
- b. Uzmanların Seçimi
- c. Bilginin Elde Edilmesi
- d. Program Geliştirilmesi

2.2.10.1 Alan Seçimi

Uzman sistem için bir uygulama alanının seçimidir. Bir uzman sistem için uygun alanın seçiminde, takip edilebilecek birçok teknik ve teknik olmayan düşünce vardır.

2.2.10.2 Uzmanların Seçimi

Seçilen alanla ilgili bir veya birden fazla uzmanın seçilmesidir. Uzman sistemler tasarlanırken, planlayıcı mutlaka, uzmanların derin tecrübelerinden ve tekrarlanan gözlemlerinden elde edilen uzmanlık bilgileri üzerine odaklanmalıdır. Bir uzman sistem geliştirme projesi için bir alan uzmanı belirlerken hesaba katılması gereken birçok faktör vardır. Burada anahtar nokta, belirlenen amaçlar ve problemler için doğru tip bilgiye sahip bir uzman bulmaktır.

Uzman; bir görevi, eğitimini ve deneyimini kullanarak, ele geçirmek ve dağıtmanın yararlı olacağı seviyede bir ustalıkla gerçekleştiren kişidir. Uzman, Nobel Ödülü kazanmış bir fizikçi veya işinin ehli bir ofis elemanı olabilir. Her ne kadar genellikle en üst seviyedeki kişinin bilgisini ele geçirmeye çalışsak da, zaman zaman orta seviyedeki bir karar vericinin muhakemelerini bile elde etmek ve otomatikleştirmek

yararlı olabilir. Bu durumda, bu kişi bizim amaçlarımız için bir uzman olarak düşünülebilir.

Çok sayıda tanımlanmış alt problemleri ile birlikte geniş bir bilgi sistemi inşa ederken, en alttan tepe yönetime kadar, organizasyonun her aşamasında, sık sık bilgi toplamak uygundur. Burada fikir, problemin nasıl büyüdüğünün, organizasyona nasıl etki ettiğini ve çözümüne kimlerin katılması gerektiğinin analiz edilmesidir. Ayrıca, bilgi tabanına ek sağlamak için diğer firmalarla ilgili bilgilere sahip dış tecrübeli uzmanlar tercih edilebilir. İç uzmanlar konu hakkında sınırlı veya yüzeysel bilgiye sahipken, dış uzmanlar başlıca bilgi kaynağı haline gelir.

2.2.10.3 Bilginin Elde Edilmesi

Uzman sistem geliştirme sırasında alan bilgisinin –gerçekler, kurallar, sezgisel yöntemler, prosedürler- uzmanlardan elde edilme süreci, bilginin elde edilmesi olarak adlandırılır. Bir uzman sistem geliştirilirken, bilgi mühendisleri, dokümanlardan ve doğrudan doğruya uzmanlardan yola çıkarak, uzman sistemin çözmeye çalıştığı problemi uzmanların çözüme biçimlerini belirlemeye çalışır. Bilginin elde edilmesinin sonucu, uzman sistemin bilgisinin ayrıntılarıyla tanımlanmasıdır.

Bilginin elde edilmesi, çoğunlukla, bilgi mühendisleri ve alan uzmanları arasında yapılan ve bilgi mühendislerinin uzmanların bilgisini temin etmeye çalıştığı toplantılar yoluyla başılır. Bilgi mühendisleri için bir alan uzmanıyla görüşmenin genel bir yaklaşımı, uzmanın kullandığı prosedür ve sezgisel yöntemleri belirleyebilmek için, seçilen alana ait görevi nasıl yerine getirdiği ile ilgili sorular sormaktır. Bilgi mühendisleri uzmana belli durumlar (örneğin, gerçek bir alan probleminin verileri) sunar ve o durumda karar vermek için kullanılan sezgisel yöntemleri temin etmeye çalışır. Bilgi mühendisleri ve uzman, temin edilmiş olan alan bilgisini uzmanın performansıyla karşılaştırmak için test problemlerini kullanırlar. Sistemin bilgisindeki zayıflıklar bulunur, değiştirilir ve yeniden uzman değerlendirmesine sunulur. Bu döngü, uzman sistemin bilgisi istenilen noktaya gelinceye kadar devam eder.

Bilginin elde edilmesi, çoğunlukla zordur. Çoğu uzman uzmanlıklarını veya karar verme şekillerini kolayca tanımlayamazlar. Birçoğu, arzu edileceği şekilde düşüncelerini rahatça ifade edebilen ve konuşkan kişiler değillerdir. Bundan başka, uzmanlar sıklıkla, kullandıkları süreçleri, sezgisel yöntemleri ve uzmanlıklarını nasıl

kullanılır hale getirdiklerinin tam olarak farkına varmadan görevlerini yaparlar. Bir karar aldıklarında, gerçekte göz önünde bulundurmuş oldukları safhaların farkında olmayabilirler. Bunlar gibi faktörlerden dolayı, bilginin elde edilmesi sürecinde, mümkün olabildiğince tam ve doğru sezgisel uzman yöntemlerinin belirlenebilmesi için, çoğunlukla bilgi mühendislerinin uzun dönemler boyunca alan uzmanlarıyla etkileşim halinde olmaları gerekir.

2.2.10.4 Program Geliştirilmesi

Belirlenen teknikleri, bilgiyi ve sezgisel yöntemleri bir araya toplayan bir bilgisayar programının tasarımıdır. Bilginin elde edilmesi sürecinde sağlanan uzman bilgisini düzenlemek için bir uzman sistem programının geliştirilmesinde beş temel eleman vardır:

- a. Yazılım aracının seçilmesi
- b. Donanımın seçilmesi
- c. Bilginin test edilmesi
- d. Bilginin dönüştürülmesi
- e. Programın test edilmesi ve yeniden değerlendirilmesi

Uzman sistemlerin çoğu bir yazılım aracı veya kabuğu (shell) kullanılarak geliştirilirler. Uzman sistemin hangi bilgisayar donanımı üzerinde geliştirileceğini belirlemelidirler. Yazılım aracının ve donanımın seçilmesi işlemlerinin ikisi birlikte geliştirme çevresi olarak adlandırılabilir.

Bilginin temsil edilmesi, bilginin elde edilmesi sürecinde sağlanan alan bilgisini alma ve bu bilgiyi uzman sistem programında temsil etmek için kullanılacak olan yaklaşımı tarif etme işlemidir. Uzmanlık bilgisi, bilgisayarda, bir programlama dili kullanılarak temsil edilebilir.

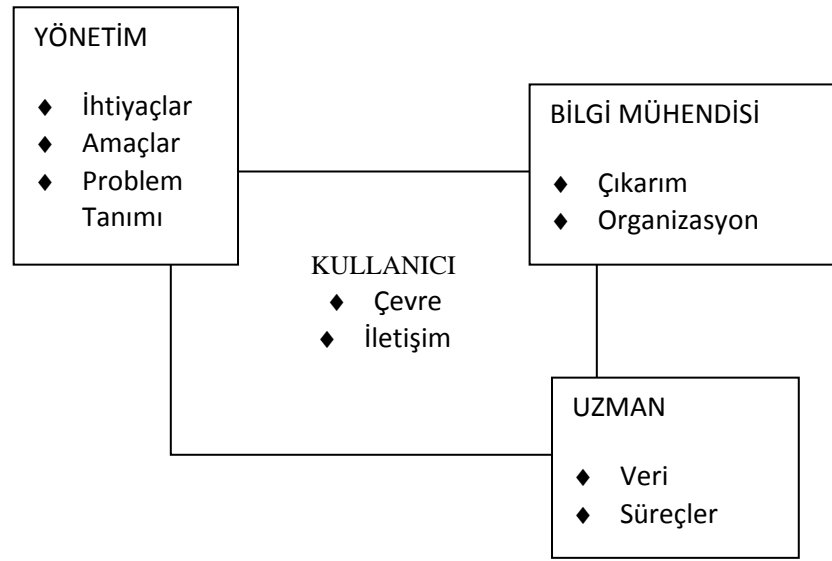
Bilginin dönüştürülmesi, bilginin elde edilme sürecinde sağlanan bilgiyi alma ve bilginin temsil edilmesini kapsayan yapıları ve paradigmaları kullanarak, bilgiyi bir operasyonel uzman sistem programına dönüştürme işlemidir.

Geliştirme sürecinde ve ayrıca son transfer veya yayılmadan önce, uzman sistem programı test edilmeli ve yeniden değerlendirilmelidir.

2.2.11 Uzman Sistemlerin İnsan ve Makine Yönü

2.2.11.1 İnsan Yönü

Şekil 2.2’de şemalaştırılan uzman sistemin insan yönü dört gruptan oluşur. Birinci grup, yönetimdir. Yönetim, uzman sistemin ihtiyacını belirlemek zorundadır ve bunu yaparken organizasyonun faydalarının sistemden türediğini farz etmelidir. İkinci grup, bilgi mühendisidir. Bilgi mühendisi, bilgi tabanındaki bilgileri uygun veri haline getirmeli ve bilgiyi sezgisel olarak organize etmelidir. Üçüncü grup kullanıcıdır. Kullanıcı, sistemin nasıl kullanılacağı, ne tip problemlerin çözüleceği ve ne tarz programların insan operatörüyle iletişim kuracağını hazırlar. Dördüncü grup ise, uzmandır. Sistem uzmana ihtiyaç duyar ya da uzman grupları gerçeklere dayalı bilgi formunda ve alan içindeki problemleri çözmeye kullanılan analitik metotlarla herhangi bir konu için bilgi tedarik ederler.

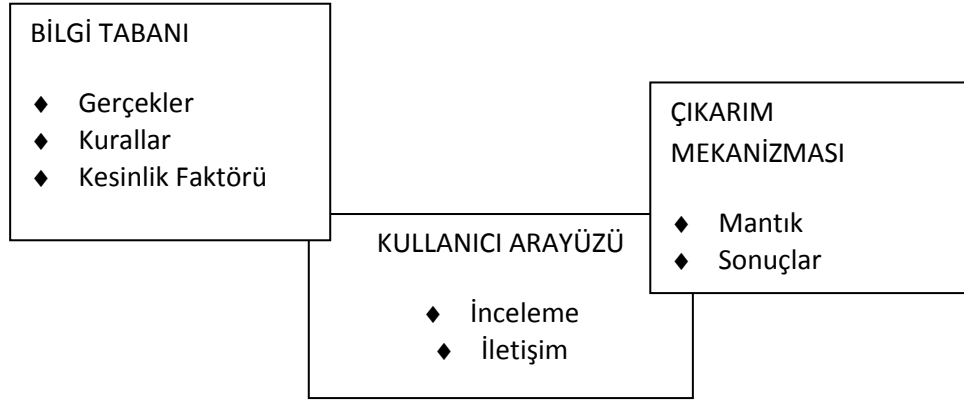


Şekil 2.3 Uzman Sistemlerin İnsan Yönü

2.2.11.2 Makine Yönü

Makine yönünde, gerçeklerin sembolik olarak temsil edildiği bilgi tabanının yazılım araçları vardır. Bütün uzman sistemlerde; bilgi tabanı, kullanıcı arayüzü ve çıkarım mekanizması bulunmaktadır. Bilgi tabanı gerçeklerin ifadesinden oluşur. Gerçekler arasındaki bağlantı sezgisel kurallarla belirlenir. Her kural ‘if’ önermesi ve

'then' sonucuna sahiptir. Önceki ve sonraki rastlantı sonucu olan olaylar tanımlanır. Gerçek iddiaların kesinlik derecesi her zaman mutlak değildir. Nicelik kesinliğinin faktörleri uzman sistemlerinin sonuçlarının doğruluğunu artırır. Bu ilgili ifadelerin güvenilirliği istatistik, olasılık veya tamamıyla kişisel fikirlere dayanır. Çıkarım mekanizması, sonuç çizimleri ve sonuçlar testinin kuralları arasında mantıksal ilişki kurar. Kullanıcı arayüzü, sistemi inceler ve yöneticiyle iletişim kurar. Uzman sistemlerin makine yönü Şekil 2.3 yardımıyla özetlenmiştir.



Şekil 2.4 Uzman Sistemlerin Makine Yönü

2.3 Elektromekanik Sektörüne Bakış

Bu projede uzman sistemlerden ve yapay zeka yaklaşımından yararlanılarak elektromekanik sektöründe kullanılacak, üreticinin ve son kullanıcının problem çözmesinde yardımcı olacak bir yazılım yapılacaktır. Bu sebeple, elektromekanik sektöründe neden bir uzman sisteme ihtiyaç duyulabileceği ve bu sektöründe karşılaşılan eşzamansızlık sorunları incelenmiştir. Parametrik Teknoloji Şirketinin internet sitesinde bulunan Elektromekanik ürün tasarımında karşılaşılan sorunlar başlıklı makalede şöyle açıklanmıştır.

Özellikle elektromekanik ürün tasarımında görev alan farklı alanlardaki uzmanların birbirleriyle uyum içinde çalışmalarının ne kadar önemli olduğu vurgulanmıştır. Tasarımda rol alan elektrik uzmanı, yazılım uzmanı ve mekanik uzmanının çalışmaları sırasında karşılaştıkları sıkıntılar aşağıdaki şekilde beş madde ile özetlenmiştir.

1-Mekanik uzmanının yaptığı değişikliklerin elektrik uzmanına zamanında iletilmemesi:

Mekanik uzmanı son şekli belirlenen elektronik kartın tasarımında, örneğin boyutunda, değişiklik yaptığı halde bu değişikliği uygun zamanda elektrik uzmanına bildirmemektedir. Elektrik uzmanı ise elektronik kartın üzerine yerleştirilecek parçaları yerleştirmiş ve karta son şeklini vermiştir. Bu ürünün boyutlarının uyumsuz olmasına yol açmaktadır.

2- Elektrik uzmanının mekanik uzmanından önce ilerlemesi:

Bu madde birinci maddenin tam tersi bir durumdur. Elektrik uzmanı elektronik kart üzerine yapılan yerleştirmede veya yerleştirilen parçalarda yaptığı değişiklikleri zamanında mekanik uzmanına bildirmez. Bu üretim sırasında gecikmeye, problemin çözümünün uzamasına ve maliyetin artmasına neden olabilmektedir.

3- Üretilen ürünün yazılım ile uyuşmaması:

Bu önlenmesi çok kolay konu firmaların en çok karşılaştığı sorundur. Yazılımın ürüne uymaması diğer bir kaotik, zaman isteyen ve yazılım uzmanı, test uzmanı ve elektrik uzmanın bir araya gelerek uzun süre çalışmalarına neden olmaktadır.

4- Tasarım yönetimi hataları:

Tasarımda rol alan bütün uzmanlar kendilerine özgü bir yapıda çalıştıklarında uyumsuzluklarla karşılaşılması çok olağandır. Proje yöneticisi de bütün bu disiplinlerin neler gerektirdiğini bilemeyebilir.

5- Yazılımın geliştirilmesi:

Yazılım çoğu zaman ürün geliştirmede projenin yönetiminde yapılan genel bir değişikliğin dışında bırakılmaktadır. Bu yüzden böyle bir değişiklikte yönetim karamsarlığa kapılır ve yazılımdaki uyumsuzluğun nelere mal olabileceği hesaplanır.

Elektromekanik ürünlerin üretiminde disiplinler arasında karşılaşılan sorunlar Çizelge 2.1 yardımıyla özetlenmiştir.

| | |
|--------------------------|--|
| 16 hafta | Makine mühendisi elektriksel kartın başlangıç şeklini tanımlar, kısıtlamalarını ve bağlayıcı gibi kritik parçaların kart üzerindeki yerlerini belirler. Daha sonra bu bilgileri elektriksel tasarım grubuna gönderir. |
| 12 hafta | Elektrik mühendisi mekanik bölüm bilgilerini alır, bileşenleri düzenler ve izleri belirler. |
| 8-10 hafta | Makine mühendisi tasarıma devam eder. |
| 6 hafta | Elektrik mühendisi kart üzerinde simülasyonu gerçekleştirir ve sinyal hatalarını tespit eder. Bu hataları düzeltmenin tek yolu hatlar için biraz daha yol açmaktır. Fakat, elektrik mühendisi bu değişikliği makine mühendisine bildirmez. |
| 3 hafta | Aynı zamanda, makine mühendisi de kart üzerine anten ve bataryayı yerleştirmede sıkıntı yaşar ve kartın alanını genişletmeye karar verir. Bu değişiklik makine mühendisine göre önemsiz görüldüğünden elektrik mühendisine haber verme gereği duymaz. |
| 2 hafta | Tasarım son kez gözden geçirilir. Proje yöneticisi makine ve elektrik mühendisinden tasarımların son halini ister. Tasarımların birbiriyle uyumsuz olduğu anlaşılır. |
| 1 hafta | Kartın temel tasarımı toplam miktarın ayrılması için kart üreticisine önceden verilmiştir fakat bu ölçü şimdi geçersizdir. |
| Süre bitimi ve devamında | Mekanik, elektriksel ve yazılım kısımlarının uyumunu sağlamak için gösterilen çabalar uzun toplantılar şeklinde devam eder. Üretim sırasında ve üretimin gecikmesi sonucu müşterilerle yaşanan sorunlar maddi ve manevi yönden kayıplara yol açmaktadır. |

Çizelge 2.1 Elektromekanik Bir Ürünün Üretim Sürecinde Yaşanan Uyumsuzluk Problemi (Anonymous. 2006. Overcoming the Top Five Challenges in Electromechanical Product Development. Parametric Technology Corporation (PTC).)

Bu sorunlara getirilebilecek çözümler ise çoğunlukla benzerdir.

1- Mekanik ve elektriksel tasarımının bütünleşmesi:

Mekanik ve elektriksel tasarımın uyumlu ilerlemesinin iki kritik noktası vardır. Birincisi her iki alanda da yapılan değişiklikler birbirini etkileyebileceğinden anında diğer alana bildirilmelidir. Hatta uyarılar otomatik olmalıdır. Elektriksel alanda yapılan işlemler diğer bölümlerce de izlenebilir durumda olmalıdır ki diğer alanları etkileyen değişikliklere anında müdahale yapılabilmelidir. İkinci önemli nokta ise tasarımın herhangi bir aşaması ile son aşamasının kıyaslanabileceği bir arayüzün program içinde bulunmasıdır. Birçok firma için disiplinler arası etkileşimli tasarımı denetlemeyi sağlayan bu sistem büyük bir avantajdır.

2- Büyük resmin organizasyonu:

Mekanik ve elektriksel kısımların uyum problemlerinde aşağıdaki temel yetkinliklere sahip bir çözüm gerekmektedir.

Tasarım bilgisi yönetimi: Tasarım araçları ve elektronik alanın yaptığı çalışmalar hakkında bilgi içeren kapsamlı bir çözüm gerekmektedir. Bu aynı zamanda değişikliklerin kolay kontrolünü de içermelidir.

İşlem akışı yönetimi: Mekanik ve elektriksel alanların birlikteliğinde en büyük ikilemlerden biri de iki disiplin arasındaki iş akışlarının yönetimidir. Bu sorunun çözülmesi için tasarım bilgilerine ve projenin son durumuna anında ulaşımın olması, ayrıca tasarım değişikliklerinin otomatik olarak bütün birimlere ulaştırılmasının sağlanması gerekmektedir.

Araç entegrasyonu: Proje yönetimi normalize edilmiş araçlardan bağımsız bir şekilde, herkes tarafından ulaşılabilirken aynı zamanda da bütün mekanik ve elektriksel araçları destekler formatta olmalıdır.

3- Yazılım açıklarının engellenmesi:

Yazılımın proje yönetimine dahil edilmesinde çoğunlukla uyuşmazlıklar çıkmaktadır, çünkü yazılım proje ekibinin dışında bir ekiple, genellikle dış kaynaklı yapılmaktadır. Dolayısıyla, yazılım geliştirme sürecini proje yönetimine dahil etmek zor bir süreçtir. Yazılımın sisteme entegrasyonunda süreci irdelemek gerekirse; elektrik mühendisi tasarımını tamamlar ve yazılım eklenip üretimi yapılmak üzere proje yöneticisine teslim eder. Fakat yazılım eklendiğinde ürünün çalışmadığı görülmektedir. Uzun incelemeler sonucunda, elektrik mühendisi tarafından tasarlanan ürünün yazılımın

son sürümüne göre dizayn edilmediği anlaşılmaktadır. Ayrıca, makine ve elektrik mühendisleri plan dahilinde birbirlerini takip edecek şekilde çalışmalarına rağmen, yazılım mühendisinin hangi noktada olduğu konusunda bilgileri bulunmamaktadır. Tasarladıkları ürün üzerinde yazılımı deneyememekte, paralel çalışmamaktadırlar. Oysa tasarlanan ürünün üretimine geçildiğinde yazılımın sisteme dahil edilmesi gerekmektedir.

Disiplinler arası ürün tasarımında karşılaşılan ekipler arası uyum probleminin çözümünde, yüksek yönetsel özelliklerin büyük önemi olduğu gibi, her bir uzmanın diğer ekiplerle güçlü iletişimini de gerektirmektedir. Alanlarda yapılan değişiklikler zamanında bildirilmeli, çözümler birlikte aranmalıdır. Üretim sürecine geçildiğinde ortak avantajlar değerlendirilmiş olup; optimum bir ürün hazırlanmalıdır.

Ürünün tasarım aşamasına geçilmeden önce üründen ne istendiğini, sistemin neleri yapacağını ve gereksinimlerin belirleneceği analiz kısmı ile elektromekanik sistemin oluşturulmasına başlanacaktır.

2.4 Sistem Analizi

Yazılım geliştirme sadece kodlamaktan ibaret değildir. Bir problemin çözümü olarak nitelediğimiz yazılımların ne yapacağını ve nasıl yapacağını belirlediğimiz yani problemi tanımladığımız aşama “Analiz” aşamasıdır. Yazdığımız kod ancak isteneni doğru bir biçimde yerine getiriyorsa başarılı bir yazılımdır. Bu nedenle öncelikle yazılımdan ne istendiğinin doğru bir biçimde tanımlanması gerekir. Yazılım geliştirmede kodlamaya başlamadan önceki aşamada ilk olarak genel bir kapsam çıkartılması gerekir. Bu kapsamla birlikte yazılımın üzerinde çalışacağı donanım bileşenlerini de kapsayacak şekilde bir sistem analizi yapılır. Ardından yazılım ve donanım gereksinimleri ana hatlarıyla belirlenince yazılım için proje planı yapılır, gerekiyorsa ekip oluşturulur ve sonrasında yazılım isterleri çözümlenmeye başlar.

Sistem analizi, hepsi bir araya geldiğinde anlamlı bir bütün oluşturan bir grup birbiriyle etkileşimli, bağlantılı veya bağımsız iş fonksiyonları, prosedürleri, aktiviteleri veya elemanlarını ayrılabilir en küçük iş parçacıklarına ayırarak tanımlamaktır. Sistem geliştirme sistemleri de diğer tüm işler gibi bir iskelet yapı altında çalışır. Sistem geliştirme de kullanılan bu çatıya SDLC (System Development Life Cycle) (Sistem Geliştirme Yaşam Döngüsü) adı verilir. Bu iskelet altında bulunan tüm işlem ve

aktiviteler, iyi tanımlanmış kapsamlı prosedürlerden oluşan metodolojiye göre gerçekleştirilir. Bu metodolojide gerçekleştirilen analiz Proje kararının verilmesi ile uygulama arasında olduğunu varsayabileceğimiz üç ana aşamadan oluşur. Bu üç ana aşama aşağıdaki genel işlevleri yürütür:

a. Analiz

b. İnceleme

c. Tasarım

Bu üç aşamada fonksiyonlar ve veriler incelenerek yeni yapının temel taşları belirlenir. Analizi yapacak olan kişiler şöyle sıralanabilir;

- Proje yöneticileri, işletme ve proje amaçlarını ortaya koyabilecek kişiler.
- Konu uzmanları, yapılan çalışmayı meslek edinen kişiler.
- Proje üzerine üretken fikirler üretebilecek kullanıcı odağındaki kişiler.

Bu projede analizi yapacak kişi bilgi mühendisi olacaktır. Uzmanlardan ve kullanıcılardan topladığı bilgileri uzman sistemin geliştirilmesinde kullanılacak yapıya dönüştürecektir. Analiz aşamasında bilgi mühendisinin sorabileceği sorulardan bazıları:

- Projenin tamamlanması için işletmenin ihtiyaçları nelerdir?
- Projenin amaçları nelerdir?
- Projenin ve çalışanların başarılı olup olmadığına nasıl karar verilecek?
- Bu başarı nasıl ölçülecek?
- Kullanıcı odağındaki kişiler kimler?
- Projenin rakipleri nelerdir?
- Rakiplerin eksiklikleri nelerdir?
- Proje elemanlarının gerçekten hepsine ihtiyaç var mıdır? Neden?
- Ürünün çözdüğü problemler nelerdir?
- Ürünün içermesi gereken özellikler nelerdir?
- İhtiyaçlar kullanıcılar mı yoksa başkaları mı belirleyecek?
- Kullanıcılar neden bu ürünü kullanmalılar?
- Projenin başarılı olduğuna kim karar verebilir? Beklentileri neler?
- Proje için ihtiyaçlar nelerdir?
- Ürünün kısıtlamaları nelerdir?

Bilgi mühendisleri projenin amaçlarını belirlemek için bu soruların ve olası diğer soruların cevaplarını alır.

2.4.1. Analiz Aşamasının Adımları

Bilgi mühendisleri analiz aşamasında aşağıdaki beş adımı uygularlar:

- Keşif
- Kullanıcı Analizi
- Eğitsel Analiz
- Kullanılma Amaçları
- Teknik Analiz

2.4.1.1 Keşif

Genellikle bilgi mühendisleri kullanıcılarına var olan rakip ürünler hakkında sorular sorarlar. Böylelikle bu süreç mühendis ve kullanıcı arasında bir mevzu olur. Bu aşamada var olan ürünler hakkında toplayabildikleri kadar bilgi toplarlar. Unutulmamalıdır ki fikri alınan her kullanıcının verdiği cevap eşsiz bir görüştür ve değerlendirilmelidir. Konu uzmanlarına sorulacak sorular uzmanlar tarafından araştırılır. Bilgi mühendisleri piyasadaki ürünleri ve diğer bütün kaynakları detaylı olarak inceledikten sonra konu uzmanlarına sorulacak soruları belirlerler. Bu soruların özenle seçilmesi gerekir çünkü konu uzmanları genellikle yoğun kişilerdir. Kendi iş yoğunluklarının dışında onları fazla sıkmadan, alınabilecek bütün bilgilerin etkili alınması için sorulacak sorular çok önemlidir.

Bu aşamanın sonunda konu uzmanları kullanıcılara incelenen ürünlerin listesini sunarlar. Kullanıcılardan bu listenin onaylanması ve eksikliklerin belirtilmesi istenir.

Keşif aşamasının faydaları:

- Bu süreçte proje için hayati bilgiler toplanır.
- Rakip ürünler hakkında bilgi sahibi olunur.
- Bilgi mühendislerinin bilgi toplamasına hız katar.
- Konu uzmanlarından istenilen bilginin daha kısa sürede alınmasını sağlar.
- Bilgiler kontrollü toplandığı için yanlış bilgi riski azaltılır.
- İleriki aşamalarda, geri dönülmesini engelleyerek zaman kazandırır.
- Projenin ilk kalite kontrolünün yapılmasını sağlar

2.4.1.2 Kullanıcı Analizi

Bilgi mühendisleri analiz aşamasında kullanıcıları bir grup olarak inceler. Bu aşamada uzmanlar kullanıcıların bilgilerini, alışkanlıklarını ve kullandıkları ürünleri inceler. Kullanıcıların alışkanlıkları nelerdir ve kullandıkları ürünlerle neler yapabilmektedirler?

Bu analizin amacı kullanıcıların gerçek ihtiyaçlarının tespit edilmesidir. Eğer kullanıcı analizi yapılmazsa kullanıcılar hakkında varsayımlarda bulunmaktan başka bir seçeneğiniz kalmaz. Bazı varsayımlar beklenmeyen sonuçların alınmasına yol açabilir.

Bu aşama sayesinde kullanıcıların alışkanlıkları ve kullandıkları programların detayları belirlenir. Bu detaylar sayesinde eksiklikler daha iyi tespit edilip hayata geçilebilir. Önceden de bahsedildiği gibi varsayımlarla yola çıkmak projeyi en başından başarısızlığa sürükleyebilir. Sorulabilecek sorular:

- Kullanıcı odağındaki kişiler kimlerdir?
- Kullanıcıların hazırda kullandıkları ürünlerin özellikleri nelerdir?
- Kullanıcıların beklentileri nelerdir?
- Kullanıcıların teknik araç-gereçleri ürünü kullanmak için yeterli mi?
- Kullanıcıların sahip olduğu internet bağlantısının özellikleri önemli mi?

2.4.1.3 Eğitsel Analiz

Eğitsel analiz de kullanıcının ürünleri etkili bir şekilde zorlanmadan kullanabilmelerini amaçlamaktadır. Bu aşamada kullanıcıların ürün hakkında bilmeleri gerekenler belirlenir. Kullanıcıların zaten bildiği şeylerin tekrar üzerinden geçmeye gerek yoktur. Bu yüzden bu bilgilerin ürün içerisinde yer almasına gerek kalmaz. Ayrıca bu aşamada kullanıcıların ihtiyacı olmayan bilgiler de üründen çıkartılır. Sonuç olarak kullanıcılar sadece bilmeleri gereken yeterli bilgileri ile ürünü etkili kullanabilmeleri amaçlanır. Ürün tasarımı ve açıklama bölümleri buna göre düzenlenir.

2.4.1.4 Kullanılma Amaçları

Unutulmamalıdır ki her aşaması ile tamamlanmış bir proje etkili olarak kullanılmıyorsa ürünün başarılı olduğu söylenilemez. Bir ürünün tam anlamıyla tamamlanmış olması için müşterilerin o ürünü etkili olarak kullanabilmesi gerekir. Kullanıcılar proje çalışanların belirlediği düzeyde ürünü amacına göre

kullanabiliyorlarsa bu analiz doğru yapılmış denebilir. Eğer ürün içerisinde kullanıcılar doğru yönlendirilmezlerse yolunu kaybetmiş bir turistten hiçbir farkları olmayacaktır.

2.4.1.5 Teknik Analiz

Bu aşamada projenin tamamlanması için gerekli teknik ihtiyaçlar belirlenir.

- Minimum gereksinimler – bilgisayar, internet bağlantısı, yazılımlar, vs.
- Server gereksinimleri – amaçlanan kullanıcı sayısını kaldırabilecek bir server
- Kullanıcıların gereksinimleri – flash plug in, IE7, vs.

2.4.2 Sistem Geliştirme Döngüsü (System Development Life Cycle)

Bir sistemin ortaya çıkışında özellikle yazılımda kullanılan değişik metodolojiler bulunmaktadır. Bu metodolojilerin kullanım amaçları sistemin sürekli yaşayan ve gelişen bir yapıda oluşturulabilmesidir. Bu yapının gelişmesi için en kritik nokta yazılımın analiz ve tasarım olarak adlandırılan kısmıdır ki zaten bu da yazılımın tüm geliştirme kısmını kapsar. SDLC olarak adlandırdığımız bu çevrim aşağıdaki safhalardan oluşur:

a. Kavramsal Analiz Safhası (Conceptual Design Phase)

Kavramsal analiz esnasında işletmenin ihtiyaçları belirlenir ve sistem ile kapsanacak kavramsal uygulamalar belirlenir. Genel olarak bir proje planı oluşturulur, tanımlar ve çözümler belirlenir.

b. Detay Analiz Safhası (Detailed Analysis Phase)

Bu safhada sistem geliştiricileri ve kullanıcılar, kavram analizde tanımlanmış olan sistemin fonksiyonel ve mimari gereklerine karar verir. Bu gerekleri karşılamak için tasarlanması gereken sistemin altyapısını planlar ve sistem geliştirme için hazırlık yaparlar.

c. Geliştirme Safhası (Development Phase)

Geliştirme safhasında sistem geliştiriciler bir önceki safhada oluşturulan detay analize uygun olarak sistemin kodlamasını ve test prosedürlerini gerçekleştirirler. Kuruluş için hazırlık yaparlar.

d. Kuruluş Safhası (Implementation Phase)

Kuruluş safhasında sistem geliştiriciler kullanıcılara kullanıcı eğitimlerini verirler ve son kullanıcılar ile birlikte sistemi test ederler. Sistemin kavram analizde ön görülen kapsam dahilinde tüm ihtiyaçları karşıladığına dair hemfikir olduğunda sistem geliştiriciler sistemin nihai kuruluşunu yapar ve sistem devreye alınır.

e. Kuruluş Sonrası Destek Safhası (Post Implementation Support Phase)

Bu safhada yöneticiler, kullanıcılar ve sistem geliştiriciler sistemi sürekli gözlem altında tutarlar. Önceki safhalarda belirlenen istek ve ihtiyaçlara uygunluğu sürekli ölçülür. Eğer gerekli olursa sistemin bütünsel yapısını bozmayacak küçük değişiklik talepleri bu safhada gerçekleşir. Sistemin gelişimi ve kullanılabilir ömrü tamamı ile bu safhanın sürekliliği ve etkinliğine bağlıdır.

2.4.2.1 Kavramsal Analiz Safhası

Bu safhada uygun sistem çözümü, sistemin temel amaçları ortaya konur ve genel bir proje planı oluşturulur. Sistem kapsamına giren iş akışları analiz edilir ve ihtiyaçlar ve hedefler belirlenir.

Genel Proje planı, projenin kapsamını, kurallarını, etki sınırlarını, ana hatlarını ve kısıtlarını yansıtır. Proje planı etkili bir detay analiz için önemlidir. Kesin ve net ifade edilmiş proje kapsamı detay analiz safhasında ihtiyaç ve çözümlerin doğru algılanabilmesini sağlar.

Kavramsal analiz safhası hedefler ve öngörülen sistem ile arasındaki köprüyü sağlar. Kavram analiz safhasında aşağıdaki adımlar sırası ile uygulanır:

a. Proje gerekleri tanımlanır

- Sistemin iyileştireceği düşünülen iş alanının mevcut durumu gözden geçirilir.
- İş alanının veri ihtiyaçları araştırılır.
- İş alanının hedefleri ve bilişim teknolojileri stratejisi gözden geçirilir.

b. Proje hedefleri tanımlanır.

- Proje hedefleri organizasyonun hedefleri ile uyuşacak şekilde düzenlenir.
- Sistemin organizasyon üzerinde yapacağı etki tanımlanır.

c. Uygun sistem çözümüne karar verilir.

- Alternatif çözümler araştırılır. (paket çözümler ~ özel çözümler)

- Genel olarak önerilen sistem için ilave donanım, yazılım ve iletişim altyapı ihtiyaçları belirlenir.
- d. Fizibilite çalışması yapılır.
- e. Kar- Maliyet çalışması yapılır.
- f. Genel Proje Planı hazırlanır.
- Sistem geliştirme alternatifleri incelenir.
 - Uygun SDLC dokümantasyon metodolojileri araştırılır.
 - Genel proje bütçesi ve takvimi hazırlanır.
 - Bir proje yöneticisi ve proje ekibi tanımlanır.
- g. Projenin sunumu ve onay alınması
- Proje bütçesi için onay alınır.
 - Projenin işletme bünyesinde mi, bir çözüm ortağı ile mi gerçekleştirileceğine karar verilir.

2.4.2.2 Detay Analiz Safhası

Detay analiz aşaması yazılımın altyapı ve mimari gerekliliklerini tanımlamak ve bu gerekleri karşılayacak şekilde sistem tasarımını oluşturmayı kapsar. Sistem geliştiriciler sistemi geliştirmeye başlamadan önce mutlaka ve mutlaka kritik noktaları tanımlamalı ve dokümante etmelidirler. Projenin büyüklüğüne göre detaylara girildikçe geliştirme ve test aşamalarındaki aksaklık ihtimalleri azaltılır. Detay analiz safhasında aşağıdaki adımlar sırasıyla uygulanır.

1. Kullanıcı ihtiyaçları belirlenir

1.1. İş akışı ve organizasyon yapısı çıkarılır

- Planlanan yeni prosesler varsa dokümante edilir.
- Planlanan organizasyon değişiklikleri dokümante edilir.
- Planlanan yeni organizasyon yapısı ile yeni proseslerin iletişimleri dokümante edilir.
- Organizasyon bilgi akış planı ve varsa proses re-engineering planları dokümante edilir.
- Üst yönetimle birlikte gözden geçirilir.

1.2. Kullanıcı ihtiyaçları tanımlanır

- Kullanıcılara yaklaşım ve iletişim tarzı belirlenir.

- Karşılıklı görüşmeler, toplantılar yürütülür.
- Mevcut endüstri/sistem trendleri ve benzer organizasyonlar gözden geçirilir.
- Mevcut sistem ve planlanan sistemler arası analizler yapılır.

1.3. Mevcut sistem tasarımı elde edilir.

2. Kalite ihtiyaçları belirlenir

2.1. Kalite ihtiyaçları tanımlanır

• Tasarım, bütçe ve iş takvimine etkilerini dikkate alınarak kalite kriterleri belirlenir.

- Performans
- Güvenilirlik
- Kullanışlılık
- Esneklik
- Bu kriterler için yönetimin onayı alınır.

2.2. Hedefleri ve ölçüm kriterleri belirlenir.

- Ölçüm kriterleri belirlenir.
- Ölçüm işlemleri belirlenir.

2.3. Ölçüm değerlendirmeleri belirlenir.

- Gelişimin ölçülmesi sağlanır.

2.4. Planlanan hedefler belirlenir.

- Yönetim ve kullanıcılarla gerçekçi gelişim seviyeleri belirlenir.

3. Sistem donanım ve yazılımı seçilir.

- Değişik ürünler ve üreticiler araştırılır.
- Kapasite ihtiyaçları belirlenir.
- Yazılım versiyonlarına karar verilir.

3.1. Tasarım gerçekleştirilir.

3.2. Test gerçekleştirilir.

3.3. Performans kıyaslaması yapılır.

4. İhtiyaç analizi ve spesifikasyonu

4.1. Olay modeli yaratılır.

- Proses akışları ve olaylara göre sistemin dışarıdan işleyişi modellenir.
- Kullanıcılarla olay modeli değerlendirilir.

5. Proses Tasarımı

5.1. İletişim tasarımı tamamlanır.

- Üst düzey akışlar tasarlanır.
- Kullanıcı ara yüz standartları dokümente edilir.
- Uyumluluk test edilir.

5.2. Pencere ve ekranlar tasarlanır.

- Veri erişimi ve veri alışverişi planlanır.
- Davranış modellemesi yapılır.
- Hata ve iletişim mesajları belirlenir.
- Kullanıcılarla gözden geçirilir.

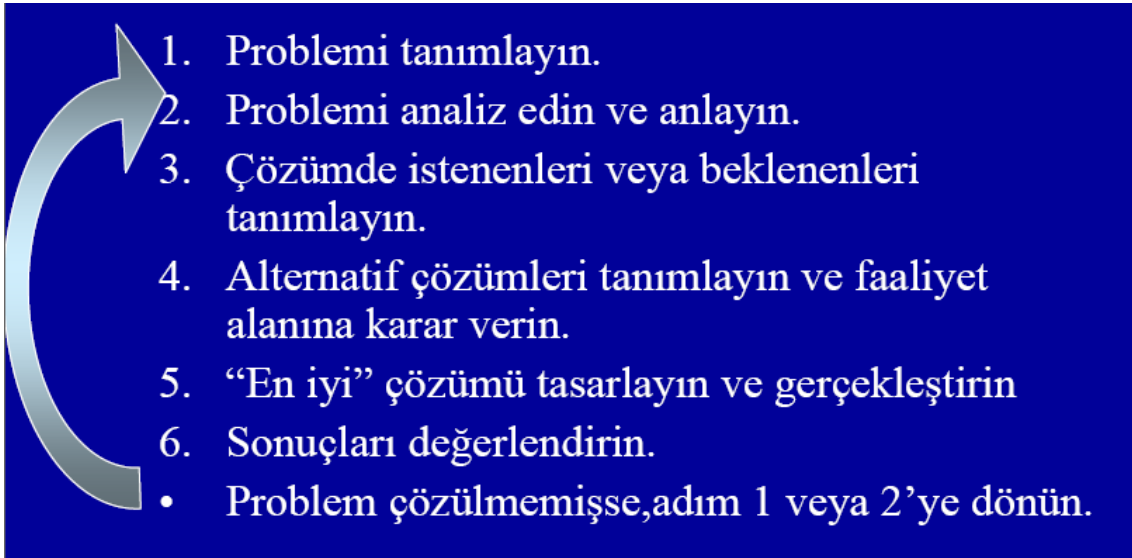
5.3. Rapor ve dokümanlar tasarlanır.

5.4. İş akışı tanımlanır.

- Tasarım işleri gözden geçirilir ve iş akışı görevleri organize edilir.
- Yeni iş akış şeması çıkarılır.
- Kullanıcı dokümantasyonu ana hatları belirlenir.

Disiplinler arası ürün tasarımı için geliştirdiğimiz sistem, farklı sistemlerin etkileşimini içerdiği için uzun bir analiz süreci gerektirmektedir. Elektromekanik bir sistemde bulunan mekanik, elektriksel ve yazılım kısımları için üç ayrı alanda analiz yapılmaktadır. Bu analizler paralel ilerleyebileceği gibi çeşitli dönemlerde bir arada ve uzmanların birlikte çalışmalarıyla da şekillenmektedir. Elektromekanik ürün tasarımında klasik yaklaşım olarak mekanik kısım analiziyle başlanmaktadır. Tasarım ürünün dış özellikleri, boyutları ve elektrik kısmın üzerine oturtulacağı zeminden başladığından analiz sürecinde bu bölümden başlamaktadır. Mekanik analiz ürünün fiziksel özellikleri üzerinden değerlendirilmektedir. Bunun yanında, mekanik kısım üzerine oturtulacak elektronik devrelerden ve beslemelerden oluşan elektriksel kısımda her bir parçanın ayrı ayrı ele alınarak sistemin gerektirdiği özellikleri taşıyıp taşımadıkları incelenecek, daha sonra bu parçaların birbirleri üzerinde olan etkileri ve etkileşimlerinin sistemi ne yönde etkilediğine bakılacaktır. Bu süreçte önemli olan nokta, her bir alanın sistem gereksinimlerine cevap vermesinin yanı sıra birbirleriyle uyum içinde çalışarak ürünü oluşturmalarını sağlamaktır. Elektriksel kısım devre

parçaları mekanik kısma uyum sağlayacak özelliklerde seçilecek, iki alanın uyumu sağlandıktan sonra son olarak bu iki sistemin birleşerek, ortak olarak çalışmasını sağlayacak yazılım kısmının incelenmesine geçilecektir. Mekanik ve elektriksel kısım analizleri kısmi olarak paralel ilerlese de, yazılım analizinin başlaması için sistemin ana yapısının oluşturulması gerekmektedir. Öyle ki, sistemin çalışma yapısına uygun olarak belirlenecek yazılım dili gerekli fonksiyonların sağlanabilirliği açısından önemli olacaktır. Örneğin nesne tabanlı bir dil kullanımı bazı sistemlerde kullanım kolaylığı sağlarken bazı sistemlerde gereksiz veri yığını haline gelebilmektedir. Yazılım dilinin seçimi analiz aşamasında belirlenmesi gereken en önemli noktadır. Dil seçildikten sonra başlanacak programda dilin daha sonradan değiştirilmesi programın baştan yazılması anlamına geleceğinden bu konuda gerekli hassasiyet gösterilmelidir.



Şekil 2.5 Analiz Süreci Adımları (İnt. Kay. 5)

Şekil 2.5’te sistem analizi adımları gösterilmiştir. Tasarlamakta olduğumuz sistem, başlangıçta model olarak belirlenen robot üzerinden gerçekleştirilmiştir. Robotu oluşturan mekanik sistem ile hareket etmesini sağlayan elektronik sistem, ayrıca elektronik sistemin mekanik sistemi etkilemesini sağlayan yazılım kısmı kimi zaman ayrı ayrı, kimi zaman ise birlikte ele alınmıştır.

Robotun mekanik kısmında parçalar birbirinden ayrılarak her bir parçanın mekanik özellikleri ve bağlantı noktaları belirlenmiştir. Birbiriyle etkileşim içinde olan bu parçalarda sistem için önemli olan özellikler tespit edilmeye çalışılmıştır. Örneğin parçaların birbiriyle uyumlu olması için ağırlık ve boyutları önemsenmiştir. Analizi

derinleştirmek adına, parçaların etkileşimini kolaylıkla gösteren bilgisayar destekli üç boyutlu modellemeler yapılmıştır. Bu modellemelerde Solidworks programı kullanılmıştır. Böylece robotun mekanik kısmı canlandırılmıştır. Mekanik kısım analiz sürecinin en kısa olduğu bölüm olmuştur. Başlangıç modellemesi yapıldıktan sonra elektriksel kısma paralel olarak düzenlemeler yapılmışsa da bu, geniş bir inceleme gerektirmemektedir. Yalnız, sistemin bütün olarak çalıştığı düşünülürse, her bir alanda yapılacak değişikliğin diğer alanları da etkileyeceği unutulmamalıdır.

Mekanik kısmın hareketini sağlayan elektronik kısmında ise elektronik kart, pil gibi devre parçalarının özellikleri incelenmiştir. Devre parçalarının ürün üzerinde yerleştirilmesine uygun fiziksel özellikler incelendiği gibi, ayrıca burada mekanik kısımdan farklı olarak parçaların birbirleriyle etkileşimi üzerinde durulmuştur. Örneğin pilin sağlaması gereken güce karşılık verdiği ısının elektronik karta etkisi önemli bir anahtar karakteristiktir. Bu yüzden farklı güçlerde pillere karşılık farklı elektronik kartlar seçilerek tasarlanacak sistem çeşitlendirilmiştir. Elektrik devrelerinde parçalar arası etkileşimlerin sistemin çalışabilirliğine etkisi büyüktür. Devre elemanlarından herhangi birinde çıkan bir arıza sistemin doğrudan durmasına neden olacaktır. Dolayısıyla elektriksel analiz hassasiyetle yapılmalı, her bir parça bütün özellikleriyle ele alınırken birlikte çalışabilirlikleri de mutlaka kontrol edilmelidir.

Yukarıda bahsedilen analizler neticesinde sistemin tasarım kısmı oluşturulmaya çalışılmıştır. Tasarım analizi tamamlandıktan sonra mekanik ve elektriksel kısmı bir arada çalıştıracak yazılımın analiz süreci başlamaktadır. Yazılım analizi doğru yazılım dilini seçmekle başlar. Sistemin gerektirdiği özelliklere cevap verebilecek bir program için en kullanışlı ve fonksiyonel dil seçimi yapılır. Daha sonra sistem modüllere bölünerek incelenir. Modüllere bölmekteki amaç, her bir modül içinde değişkenlerin belirlenmesi, böylece işlevselliğin artırılmasıdır. Yazılım sistemde oluşabilecek problemlere karşı çözüm üretecek bir bölümü de kapsamaktadır. Bu bölüm için ayrıca etkileşimler ve karakteristik özellikler göz önünde bulundurularak belirleyici sorular oluşturulmuştur. Sorular kullanıcının sorunu tanımlamasına yardımcı olarak kullanıcıdan alınacak cevaplar doğrultusunda çözüm yolları, uyarı ve öneriler sunulmasını sağlamaktadır. Elektronik kısımdaki parçaların birbirleriyle olan etkileşimleri göz önüne alınarak hazırlanan sorular sisteme oluşturulan veritabanı ve kullanıcı ara yüzü vasıtasıyla dahil edilmiştir.

3. MATERYAL VE METOT

3.1 Uzman Sistem İnsan Yönünün Uygulanması

3.1.1 Yönetim

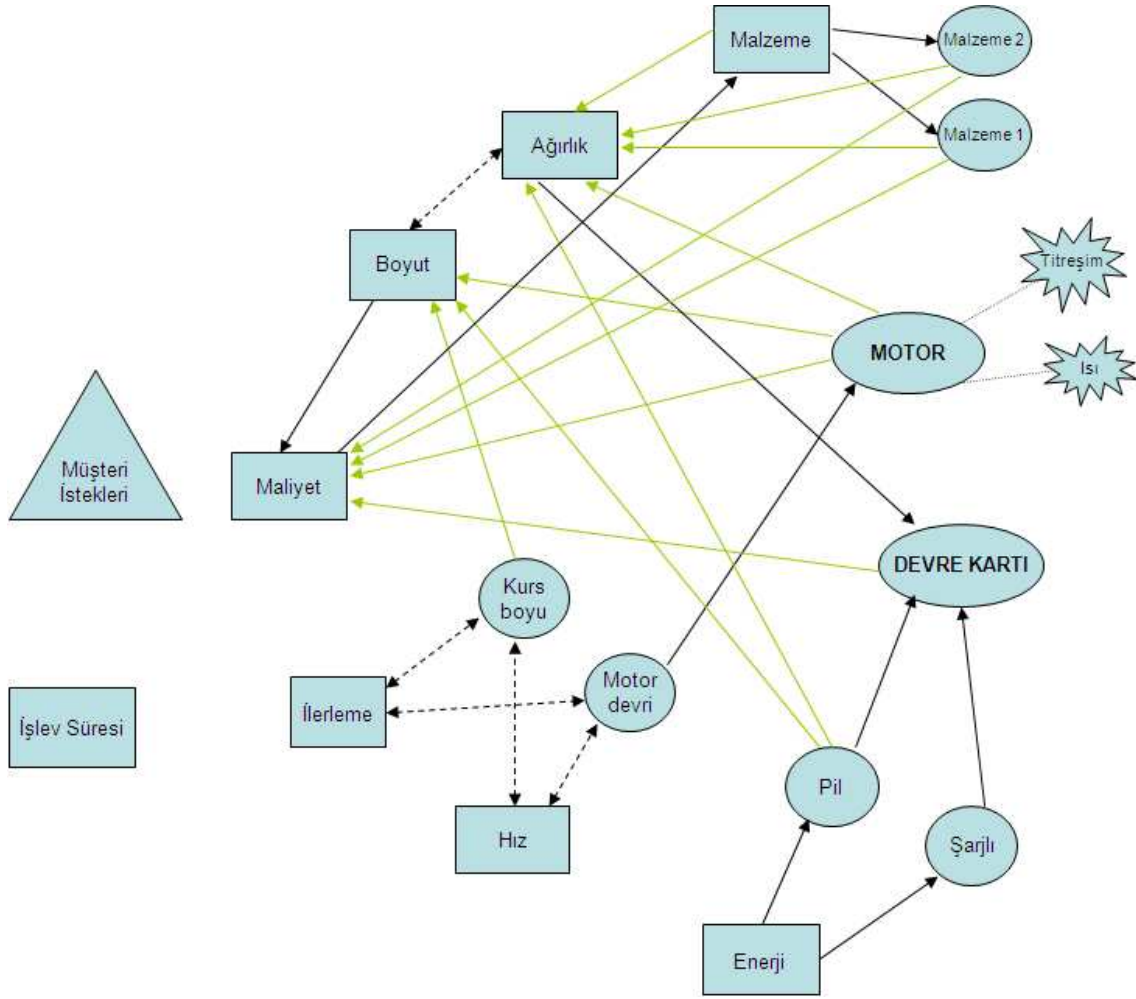
Yönetim bölümü üç adımda ortaya konulmaktadır. İlk adım ihtiyaçların belirlenmesidir. İhtiyaçlar özel bir çalışma sonucu belirlenebileceği gibi sektörde var olan ve zamanla fark edilen ihtiyaçları da kapsayabilmektedir. Bu projede ihtiyaçlar araştırmalar sonucu ortaya çıkmış daha sonra ihtiyaç analizi yöntemiyle belirginleştirilmiştir. Bu ihtiyaçlar doğrultusunda problem tanımlanmıştır ki bu ikinci adımdır. Tanımlanan problem üzerinden amaçlar tespit edilmiştir. Amaçların tek yönlü olmaktan ziyade çok yönlü olarak belirlenmesi ve bu doğrultuda bir sistem oluşturulması uzman sistemin değişik sektörlere uygulanabilirliğini, esnekliğini ve çözüm bulma gücünü arttıracaktır. Bu son adımda da kullanıcı, bilgi mühendisi ve uzmanlardan alınan veriler değerlendirilerek analiz edilmiştir.

3.1.2 Uzman

Uzman sistemlerin en önemli birimi olan uzmanlar verilerin belirlenmesi ve sistem algoritmasının kapsadığı süreçlerin organizasyonunda önemli etkiye sahiptir. Elektromekanik alanında yapılan bir uzman sistemde elektrik, mekanik ve bilgisayar alanlarından olmak üzere çeşitli uzmanlar seçilmiş, hem sistemin oluşturulmasında hem de son kullanıcı olarak bilgilerine başvurulmuştur. Uzmanlarla yapılan görüşmeler ile alınan bilgiler analiz edilerek amaçlar doğrultusunda değerlendirilmiş ve sisteme dahil edilmiştir.

3.1.3 Bilgi Mühendisi

Bilgi mühendisi uzmanlardan alınan verilerin analiz sürecine tabi tutar ve sisteme dahil edilecek çıkarımları yapar. Bu verilerin ışığında bir algoritma hazırlayarak uzman sistemi organize eder. Şekil 3.1'de sistem parçaları arasındaki etkileşimleri gösteren algoritma yer almaktadır.



Şekil 3.1 Etkileşim Algoritması

3.1.4 Kullanıcı

Uzman sistemin hazırlanma amacı belli bir alandaki kullanıcılara çözüm sunmaktır. Bu sebeple sistemin sunulacağı çevre analiz edilmelidir. Alan hakkında bilgi edinilerek son kullanıcı profili çıkartılabilecek ve buna göre sistem şekillendirilecektir. Örneğin, bilgisayar kullanmayı bilmeyen bir çevreye kullanıcı arayüzü karışık bir yazılım sunulamayacaktır. Bu projede elektromekanik sektörü ele alındığından yazılımda mekanik ve elektronik öğeler ön planda kullanılmıştır.

3.2 Uzman Sistem Makine Yönünün Uygulanması

3.2.1 Bilgi Tabanı

Uzman sistemin insan yönünden faydalanarak uzmanlardan alınana veriler işlenmiş ve sistemin algoritması hazırlanmıştır. Makine yönünde ise bu veriler gerçeklere ve kurallara dönüştürülmüş, kesin yargılara bağlanarak bir veritabanı ile sisteme dahil edilmiştir. Veritabanı oluşturulmasında, verilerin çok sayıda olmaması, aşırı güvenlik gerektirmemesi, sistemi ağırlaştırmaması gibi faktörler göz önünde bulundurulurken program olarak MS Access seçilmiştir. Bu sayede birçok insanın günlük hayatta kullandığı ve birçok bilgisayarda hali hazırda kurulu bulunabilecek bir program seçilmiştir.

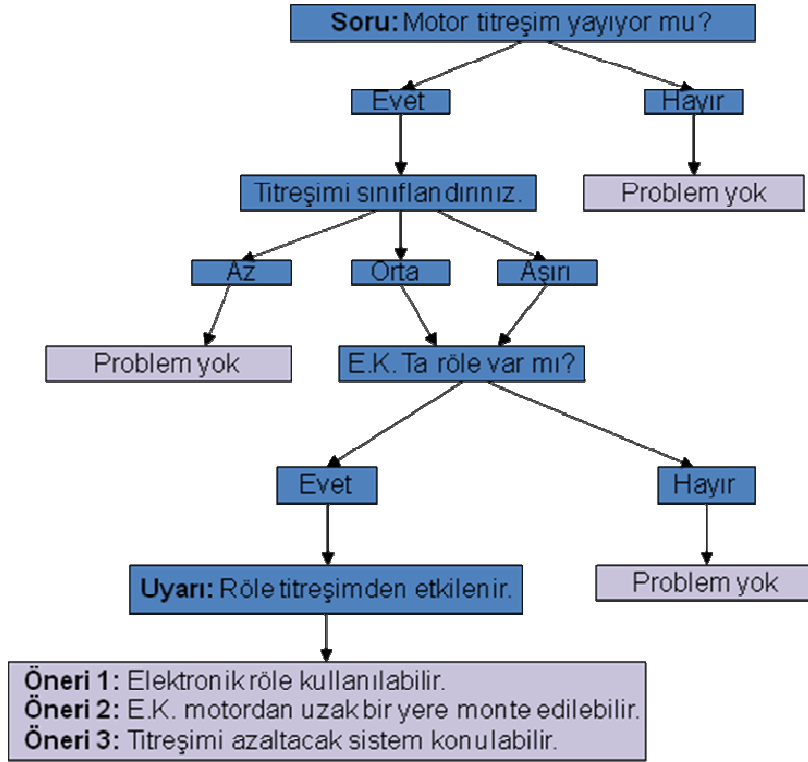
3.2.2 Çıkarım Mekanizması

Algoritma bu safhada geliştirilerek veritabanı ile kullanıcı arayüzü arasında köprü olacak mantık kuralları ve sonuçlara dönüştürülmüş ve yine bir tabloda saklanmıştır. Şekil 3.2’de uzman sistemin mekanik kısmını, Şekil 3.3’te ise etkileşim kısmını kapsayan kurallar ve sonuçlar algoritması gösterilmektedir.

HIZ: K-1: Girilen hız değeri sistemimizin max. ve min hız değeri arasında mı:
H-1: Evet ise; bu değeri sağlayacak motor/motorları seç ve AĞIRLIK seçeneğine yönlendir.
H-2: Hayır ise;
UYARI 1: Lütfen max.-min.. Değerleri arasında bir değer giriniz
çıkısın ve yeniden hız değeri girilecek ekrana dön.

BYT: Girilen x,y,z boyutu mekanizmamızın x,y,z boyutunu sağlıyor mu:
B-1: Evet ise; MALİYET seçeneğine yönlendir.
B-2: Hayır ise;
UYARI 3: a) İsteddiğiniz hız ve ağırlık özellikleri sağlayan motorun boyutu girdiğiniz boyutu sağlamıyor, motoru değiştirmek istermisiniz? (Eğer mekanizmanın boyutu motor yüzünden değişiyorsa)
B-3: Evet ise; Motor seçenekleri sunulur ve hız değeri girilen ekrana dönülür.
B-4: Hayır ise;
UYARI: Mekanizmanın boyutu min .. Değerini sağlıyor, boyut değerini yeniden giriniz
(boyut girilen ekrana dönülür)
b) Mekanizmanın boyutu istediğiniz değeri sağlamıyor, min X değerinde, bu boyutta olsun istermisiniz?
B-5: Evet ise; MALİYET seçeneğine yönlendir
B-6: Hayır ise;
UYARI: Lütfen boyut değerini başka bir değer girin (boyut değeri girilen ekrana dönülür)

Şekil 3.1 Mekanik Kısım Kuralları ve Sonuçları



Şekil 2.3 Etkileşim Kısmı Kural ve Sonuçları

3.2.3 Kullanıcı Arayüzü

Uzman sistemin hedef kitlesi olan kullanıcılara uygun bir arayüz tasarımı için, kullanıcılarla iletişim kurularak profil belirlenmiştir. Sistemin gereksinimlerini karşılayacak, kullanıcıya kolay kullanılabilir bir arayüz sunacak, veritabanı bağlantılarının rahatlıkla kullanılabilmesi ve farklı sektörler için yeniden geliştirilebilir bir istem sunabilecek bir yazılım dili seçilmesi amaçlanmıştır. Başlangıçta açık kaynak kodlu bir yazılım olan Delphi programlama dili uygun görülmüş ise de gerek fonksiyonlarının geniş yelpazesi gerekse kaynak bulma kolaylığı açısından .NET 2005 paketinde yer alan C# dili tercih edilmiştir. Şekil 3.4'te örnek olarak soru ekleme arayüzü gösterilmiştir.

admin

ANA MENÜ EKLE TEKRAR ÇIKIŞ

Soru Öneri Uyarı

SORU EKLE

Yeni soru metnini giriniz.

Önceki Soru Kodu :

Sonraki Soru Kodu :

Cevaplar : Çoktan Seçmeli Evet-Hayır

Cevap 1 : Cevap 2 :

Cevap 1den sonra : Cevap 2den sonra :

Cevap 3 : Cevap 4 :

Cevap 3ten sonra : Cevap 4ten sonra :

Resim 3.1 Soru Ekleme Arayüzü

4. SİSTEM TASARIMI

4.1 Tasarıma Giriş

Sistem tasarımı, uzun analiz sürecinden sonra elde edilen bilgilerin son kullanıcıya uygun hale getirilerek yazılım ortamında, kolay kullanılabilir, geliştirilebilir ve farklı sektörlere uygulanabilir bir formatta sunulmasını sağlayacak programın tasarlanması sürecidir. Bu süreçte, son kullanıcının istekleri yeniden değerlendirilecek, tabloların yazılım ile uyumu kontrol edilecek, arayüz tasarımı hazırlanacak, son kullanıcının programı test etmesi sağlanarak belirlenecek ikilemler tekrar ele alınacaktır.

Tasarım sürecinde analizden farklı olarak kullanıcının isteklerinden çok bu taleplerin sisteme uygulanabilirliği tartışılacaktır. Öyle ki, talepler doğrultusunda hazırlanacak arayüz yazılım dilinin işlevselliğine göre kullanıcının isteğini tam olarak karşılayamama riski taşımaktadır. Dolayısıyla, tasarım yazılımcının yapabilecekleri, kullanıcının talepleri ve yazılım dilinin uygulanabilirliği ile şekillenmektedir.

4.1.1 Uzman Sistemin Alt Bölümleri

Uzman sistemlerin alt birimleri Şekil 2.2’de gösterilmiştir. Bu projede elektromekanik sektörü için hazırlanan uzman sistemin alt birimleri bu şemaya göre incelenecek olursa;

Alan bilgisi, uzmanların konu hakkında sahip oldukları ve sistemin geliştirilmesinde yol gösterecek olan tecrübeleridir. Elektromekanik sektörü için elektrik, elektronik, mekanik ve bilgisayar alanlarındaki bu sisteme uygun her bilgi alan bilgisi olarak kullanılacaktır.

Konu uzmanı, sektör hakkında tecrübesi olan kişilerdir. Uzman sistemin veritabanının ve algoritmasının oluşturulmasında uzmandan alınan bilgiler işlenerek kullanılır. Uzman olarak elektrik, elektronik, mekanik ve bilgisayar alanlarından akademisyenlerden, elektrik alanında Yrd.Doç.Dr.Seydi V. ÜSTÜN, mekanik alanında Yrd.Doç.Dr.Ahmet GAYRETLİ, bilgisayar alanında ise Öğr. Gör. Mustafa NARTKAYA hocalarımızdan yardım alınmıştır.

Kullanıcı, sistemi tasarladığımız elektromekanik sektöründe faaliyet gösteren üreticiler kullanıcı grubu olarak tanımlanmıştır. Fakat projenin yapım sürecinde son kullanıcı olarak sistemi test edebilmek için mekanik bölüm öğrencileri seçilmiştir.

Bilgi tabanı, uzmanlardan ve kullanıcılardan toplanan veriler incelendikten sonra amaçlar ve kurallar doğrultusunda belirlenen algoritmalara göre şekillendirilmiştir. Veri tabanı sistem parçalarına ait verileri, bunların aralarındaki etkileşimleri ve etkileşimleri yöneten soruları kapsamaktadır.

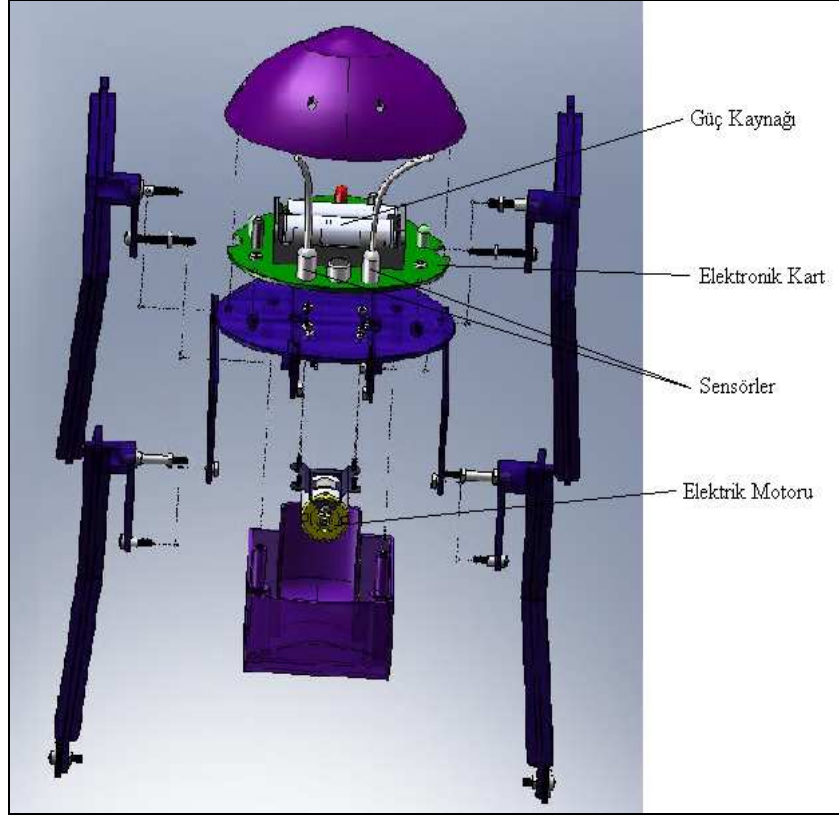
Hafıza (gerçekler), veritabanını şekillendirmek için uzmanlardan toplanan verileri etkileşimleri yönetecek ve arayüzle veritabanı arasında köprü görevi görecektir.

Arayüz, sistemin kullanıcıya sunulması, kullanıcıdan verilerin alınması ve bir sonuç geri verilmesini sağlamaktadır. Bu projede C# yazılım dili ile yapılan arayüzler kullanılmıştır. Arayüz veritabanı, son kullanıcı, kurallar, sonuç çıkarma mekanizması gibi bir çok alt birimin bir arada çalışmasını sağlamaktadır.

Sonuç çıkarma mekanizması, yazılımın arayüz aracılığıyla son kullanıcıdan aldığı giriş bilgilerini kurallar ve gerçekler doğrultusunda işleyerek kullanıcıya bir geri bildirim, bir sonuç vermesidir. Sonuç çıkarımında sistem veritabanında bulunan etkileşim ve soru tabloları kullanılmaktadır. Etkileşim algoritması tamamen etkileşim tablosuna aktarılmıştır. Böylece sorulan sorular alınarak bu yol takip edilir ve sonuca gidilir.

4.2 Anahtar Karakteristiklerin Belirlenmesi

Analiz aşamasında robotun mekanik ve elektronik kısımları incelenmiş her bölüm için farklı anahtar karakteristikler belirlenmiştir. Bu karakteristik özellikler doğrultusunda etkileşim tablolarını içeren bir veritabanı oluşturulmuştur. Veritabanı oluşturulurken robot parçalarının ortak özellikleri belirlenmiştir. Mekanik kısım için oluşturulan tabloda ağırlık, boyut gibi değişkenler belirleyici olurken elektronik kısımda güç, ağırlık, hız gibi özellikler göz önüne alınmıştır.



Resim 4.1 Model Seçilen Robotun Parçaları

4.3 Veritabanının Oluşturulması

4.3.1 Tabloların Belirlenmesi

Mekanik kısımda mekanik, motor ve besleme olmak üzere 3 tablo oluşturulmuştur. Mekanik tablosu robotu oluşturan parçaların her biri için özellikleri malzeme, ağırlık, hacim, en, boy, genişlik, sıcaklık ve maliyet olarak listelemektedir. Dolayısıyla mekanik parçalar istenilen özelliklerde ve sisteme uyum sağlayacak şekilde seçilebilecektir. Tablo 4.1’de gösterilen motor tablosu robotun hareketini sağlayacak motorun çap, uzunluk, ağırlık, voltaj, akım, hız, tork ve maliyet değişkenlerini içermektedir. Besleme tablosunda ise sisteme güç verecek ve motoru çalıştıracak pil seçimi çap, uzunluk, ağırlık, güç, voltaj ve akım özellikleriyle belirtilmektedir. Görüleceği üzere, tablolar parçaların sistemde değişkenlik gösterebilecek özellikleri üzerinden göz önüne alınarak hazırlanmıştır.

| ad | cap | uzunluk | agirlik | voltaj | akim | hiz | tork | maliyet | resim |
|------------------|------|---------|---------|--------|-------|-------|------|---------|-------|
| FF-050S-08220Y | 15,5 | 39,8 | 1 | 12 | 0,09 | 6270 | 0,49 | 8 | |
| FF-050S-09250 | 15,5 | 39,8 | 2 | 8 | 0,11 | 6300 | 0,69 | 8 | |
| FF-050S-13130 | 15,5 | 39,8 | 1,2 | 2 | 0,11 | 3000 | 0,29 | 8 | |
| UC-130-2090 | 20 | 38 | 1,5 | 3 | 0,73 | 9200 | 0,88 | 8 | |
| UC-130-07500 | 20 | 38 | 2 | 12 | 0,11 | 5920 | 0,69 | 6 | |
| FE-130-2270-MV | 20,1 | 38 | 2,1 | 1,5 | 0,65 | 6400 | 0,72 | 5 | |
| FE-130-15160-MV | 20,1 | 38 | 1,6 | 3 | 0,28 | 6000 | 0,63 | 9 | |
| FE-130S-12230-MV | 20,1 | 38 | 1 | 3 | 0,11 | 3139 | 0,5 | 7 | |
| FF-180SH-2852-CV | 20,4 | 42,75 | 1 | 2,4 | 0,86 | 6900 | 1,76 | 6 | |
| PC-130SF-9480-CV | 20,4 | 38 | 2 | 12 | 0,108 | 4800 | 1,21 | 6 | |
| PC-260P-08600-CV | 24 | 38 | 2 | 12 | 0,08 | 2810 | 1,18 | 5 | |
| UC-260S-12300-CV | 24 | 38 | 1 | 24 | 0,37 | 14000 | 3,82 | 7 | |
| * | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Resim 4.2 Motor Tablosu

Yukarıda elektromekanik sistemin tasarlanmasında kullanılan temel beş tablo açıklanmıştır. Bu tabloların yanı sıra, tasarımın ikinci aşaması olan tasarımda bulunan mekanik ve elektronik malzemelerin etkileşimlerinin incelenmesinde kullanılmak üzere soru ve etkileşim adında iki tablo daha oluşturulmuştur. Tablo 4.2’de gösterilen soru tablosu sistemde oluşabilecek problemleri analiz edip sebebini bulmaya yönelik hazırlanan sorulardan, iki sütun halinde oluşturulmuştur. Sütunlardan biri soru kodunu, diğeri ise soru metnini içermektedir. Sorular üç karakter olarak kodlanmıştır. İlk karakter aşağıda listelenen şekilde soru çeşidini belirlemektedir.

S : Soru

C : Cevap

U : Uyarı

O : Öneri

Z : Seçmeli soru

P : Problem yok

Soru çeşidi program içinde sorunun görüntülenme şeklini belirlemek için kullanılmaktadır. İkinci ve üçüncü karakterler ise sorunun kendi çeşidi içinde kaçınıcı sırada olduğunu göstermekte ve sorular arasında bağ kurmayı sağlamaktadır. Bu kısım sonraki bölümlerde parçalar arası etkileşimler incelenirken açıklanacaktır. Soru metni soru koduna karşılık gelen, çeşidine göre, soru, uyarı veya öneriyi içermektedir.

| sorukod | soru |
|---------|---|
| S01 | Motor titreşim yayıyor mu? |
| S02 | Motor ısı yayıyor mu? |
| S03 | Motor manyetik alan yayıyor mu? |
| S04 | Sistemde E.K. Üzerindeki bakır yollarda |
| S05 | E.K. Üzerinde artık maddelere |
| S06 | Şebeke gerilimi değeri cihazın çalışma |
| S07 | E.K. Üzerindeki bakır yolları bozabilecek |
| S08 | E.K. Motora yakın mı? |
| S09 | Sistem pille mi çalışıyor? |
| S10 | E.K.'in sıvıyı sızdırmayan bir koruması var mı? |
| S11 | E.K.'a temas eden sıvı iletken mi? |
| S12 | Kimyasal madde veya gazlar E.K.'ta yakın mı? |
| C01 | Titreşimi sınıflandırınız. |
| C02 | Isıyı sınıflandırınız. |
| C03 | Sistem nasıl bir ortamda çalışıyor? |
| U01 | Röle titreşimden etkilenir. |
| U02 | E.K. Manyetik alandan etkilenir. |
| U03 | E.K. ısıdan etkilenir. |
| U04 | Aşırı toz ve kirlenmeden |
| U05 | Pillerde zamanla paslanma oluşabilir. |
| U06 | Nem E.K.'ta zamanla çürümeye |
| U07 | Temas eden sıvı E.K. devrelerinde kısa devre |
| U08 | Artık maddelerden dolayı oluşan |
| U09 | E.K.'taki bakır yolların kapanarak kartın |

Resim 4.3 Soru Tablosu

Etkileşim tablosu soru tablosunda listelenen sorular arasında bağ kurmak için oluşturulmuştur. Sorular soru kodları ile temsil edilmiş, ilk soru “S01” olarak seçilmiş, son soru ise “sonraki” sütununa yazılan “SON” kelimesiyle belirlenmiştir. Sorular arasında geçişler sorunun cevaplanması veya cevaplanmadan geçilmesi şartlarına göre iki şekilde sağlanmıştır. Cevaplanması durumunda, sorunun “evet-hayır” şeklinde bir cevabı varsa bir sonraki soru “secenek_evet” ve “secenek_hayır” sütunlarından belirlenir. Şıkları olan bir soruya bu seçeneklere ilaveten “secenek3” ve “secenek4” sütunları bulunmaktadır. Eğer soru cevaplanmadan geçilmek isteniyorsa veya bir önceki soruya dönmek istenirse burada belirleyici olan “onceki” ve “sonraki” sütunlarıdır. Bütün adı geçen sütunlarda sorular üç karakterli soru kodlarıyla temsil edilmiştir. Dolayısıyla etkileşim tablosu ile soru tablosu birbirine soru kodlarıyla bağlıdır.

Tabloya herhangi bir soru, öneri veya uyarı eklenmek istediğinde soru kodunun doğru şekilde hazırlanarak soru tablosuna eklenmesi ve etkileşim tablosunda yeni soru ile mevcut sorular arasında gerekli bağların kurulması yeni sorunun sisteme entegrasyonu için yeterli olacaktır.

4.3.2 Veri Girişlerinin Yapılması

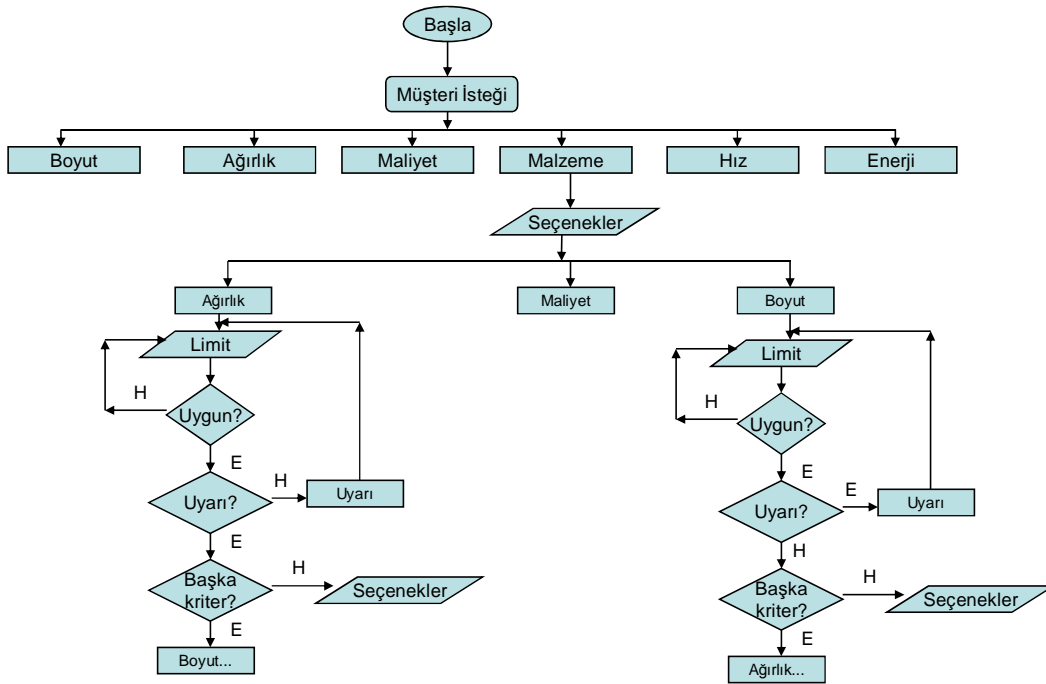
Tablolar tasarladığımız sistemin en önemli bölümüdür. Sistem yazılımı tamamen bu tablolarda bulunan veriler üzerine kurulmuştur. Yukarıda açıklanan altı tablo sistemin ilk analizinde şekillenmiş ise de programın gelişim sürecinde ihtiyaçlar doğrultusunda yenilenmiştir. Bütün tablolara başlangıç verileri manuel olarak girilmiştir. Mekanik bölüm tabloları seçilen örnek modellerin parçaları üzerinden oluşturulmuştur. İkinci bölüm olan etkileşim tabloları ise tamamen sistemin analiz ve geliştirilmesi süreçlerini kapsayan ve geliştirilmesi kullanıcı tarafından devam edebilecek olan uzun bir dönemde meydana gelmiştir. Dolayısıyla verilerin etkileşim tablolarına yazılması manuel olarak tamamlanmış olsa da kullanıcı eklemek istediği veriyi tabloya yazılım üzerinden ekleyebilecektir. Bu sistemin geliştirilmesi, sistemin kullanıcı isteklerine cevap verebilmesi, kullanılan iş alanına göre uyumlu hale gelebilmesi gibi birçok açıdan sistemin esnekliğini sağlamaktadır. Fakat sisteme eklenecek sorunun etkileşim tablolarına uyumlu olması gerektiğinden sorunun diğer sorularla bağlantısının kurulmasında kullanıcı karışık bir arayüzle karşılaşabilmektedir. Bu sebeple, yetki verilecek bir kullanıcı sisteme yetkili olarak giriş yaparak tablolara doğrudan ulaşabilecek ve tablolarda düzenleme yapabilecektir.

5. SİSTEM YAZILIMI

5.1 Algoritmanın oluşturulması

Analiz süreci, kullanıcı taleplerini incelemek, sistemin anahtar özelliklerini çıkarmak ve bu doğrultuda gereksinimleri belirlemeye yaramaktadır. Analiz sürecinin sonunda ise elde edilen veriler doğrultusunda sistem yazılıma dökülmeden önce, herhangi bir yazılım diline uyarlanabilecek, programın özetini ve akışını içeren bir algoritma oluşturulmalıdır. Algoritma, sistemi yazılımcı ve tasarımcıya özetleyerek bunların büyük resmi dışarıdan görmelerine yardım etmesi açısından önemlidir. Analiz sürecinde sistemi sürekli irdeleyen yazılımcı sistem içinde kaybolup ortaya çıkabilecek problemleri gözden kaçırabilmektedir. Bu bakımdan, dışarıdan oluşturulacak genel bir bakış sistemin genel yapısını görmekte yardımcı olacaktır.

Algoritma, ilk olarak, belirlenen anahtar özelliklerin birbirleriyle ilişkilerini düzenlemeyi gerektirir. Çünkü, bütün sistem bu özellikler üzerine kurulacaktır. Sistemin anahtar özellikleri mekanik kısımdaki parçalar ile etkileşimlerdir. Bu doğrultuda hazırlanan tablolar algoritmanın da temeli olacaktır.



Şekil 5.1 Akış Şeması

Şekil 5.1’de görülen diyagram müşteri isteği baz alınarak, sistemin ilk kısmı olan mekanik bölümün etkenleri boyut, ağırlık, maliyet, malzeme, hız ve enerji üzerine hazırlanan akış şemasının bir parçasıdır. Şemada malzeme değişkeni için düzenlenen akış algoritmada diğer tüm etkenler içinde söz konusudur. Burada önemli olan müşterinin (son kullanıcının) hangi etkeni öncelikli olarak seçtiğidir. Algoritma seçilen etkenlerin öncelik sıralamasına göre ilk etkenden başlayarak limit kontrolleri yaparak son değişkene kadar ilerlemektedir. Son olarak kullanıcıya seçilen değerlere uygun kullanılabilir malzemeler listelenecektir.

H-1: Girilen hız değerini sağlayacak motor/motorlar varsa seç ve AĞIRLIK seçeneğine yönlendir
Eğer girilen hız değerini sağlayacak motor yoksa UYARI: istediğiniz hız değerini sağlayacak motor yok. Lütfen 1-10 m/dk (sistemimizde

HIZ: K-1: Girilen hız değeri sistemimizin max ve min hız değeri arasında mı:

H-1: Evet ise; bu değeri sağlayacak motor/motorları seç ve AĞIRLIK seçeneğine yönlendir.

H-2: Hayır ise;

UYARI 1: Lütfen max...-min.. Değerleri arasında bir değer giriniz
çıksın ve yeniden hız değeri girilecek ekrana dön.

H-1 de seçilen motorla/motorlarla birlikte mekanizmamızın ağırlığı max ve min iki değer arasında olmuştur.

AĞ: K-2: Girilen ağırlık değeri mekanizmamızın ağırlık limitleri arasında mı:

A-1: Evet ise; bu değeri sağlayacak motorları seç ve BOYUT seçeneğine yönlendir.

A-2: Hayır ise;

(Eğer istenilen ağırlık değeri motor yüzünden sağlanamıyorsa)

UYARI 2: istediğiniz hız limiti nedeniyle sistemin ağırlığı min=x,max=y olduğundan istediğiniz ağırlık sağlanamamaktadır.
Bu durumda Seçenekler:

a) lütfen ağırlığı x ile y arasında bir değer giriniz (yeniden ağırlık değeri girilecek ekrana dön **A-3**)

b) lütfen hız değerini farklı bir değer giriniz (yeniden hız değeri girilecek ekrana dön **A-4**)

BYT: Girilen x,y,z boyutu mekanizmamızın x,y,z boyutunu sağlıyor mu:

B-1: Evet ise; MALİYET seçeneğine yönlendir.

B-2: Hayır ise;

UYARI 3: a) İsteddiğiniz hız ve ağırlık özellikleri sağlayan motorun boyutu girdiğiniz boyutu sağlamıyor, motoru değiştirmek istermisiniz?
(Eğer mekanizmanın boyutu motor yüzünden değişiyorsa)

B-3: Evet ise; Motor seçenekleri sunulur ve hız değeri girilen ekrana dönülür.

B-4: Hayır ise;

UYARI: Mekanizmanın boyutu min .. Değerini sağlıyor, boyut değerini yeniden giriniz (boyut girilen ekrana dönülür)

b) Mekanizmanın boyutu istediğiniz değeri sağlamıyor, min X değerinde, bu boyutta olsun istermisiniz?

B-5: Evet ise; MALİYET seçeneğine yönlendir

B-6: Hayır ise;

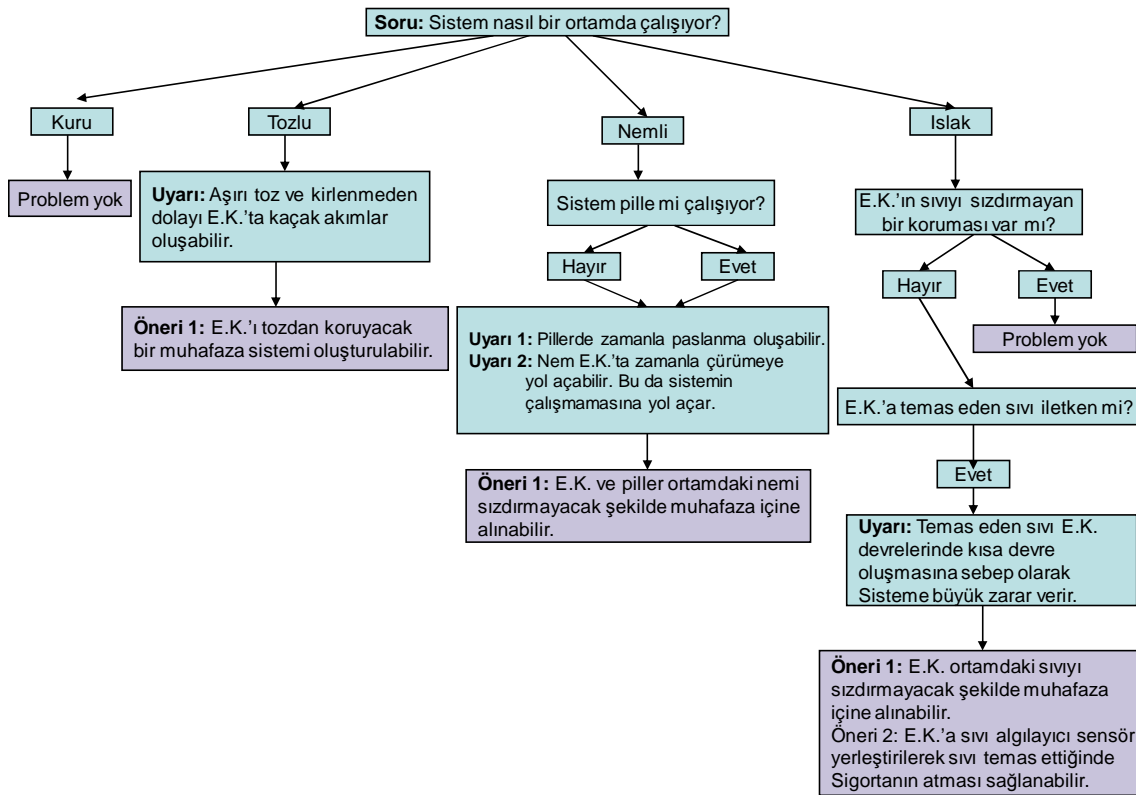
UYARI: lütfen boyut değerini başka bir değer girin (boyut değeri girilen ekrana dönülür)

Şekil 5.2 Algoritma

Şekil 5.2’de algoritmanın bir bölümünü açıklayıcı niteliktedir. İkinci bölüm olan etkileşim modülüne ait algoritma bir önceki algortmadan farklı olarak etkenler üzerine

değil sorular üzerine kurulmuştur. Her bir soru bir etken gibi ele alınmış bir uyarı veya öneriye kadar devam etmiştir. Bu arada gerekiyorsa ara sorular ile çözüme gidilmeye çalışılmıştır.

Soru kodları akış diyagramı çizildikten sonra belirlenen soru çeşitlerine ve sayılarına göre oluşturulmuştur. Algoritmada ilk soruya verilebilecek yanıtlar evet veya hayır olarak belirlenmiştir. Fakat ikinci soru üç şıklı yani çoktan seçmelidir. Bu gibi durumlar göz önüne alınarak tabloda çoktan seçmeli sorular için bir çözüm düşünülmüş ve seçenekli sorular farklı bir kodla temsil edilmiştir. Algoritmanın büyük resmi göstermesi anahtar özellikleri ve etkileşimleri fazla olan bu gibi sistemlerde tasarımı anlaşılır hale getirmekte önemli rol oynamaktadır. Bir soruya ait akış şeması o sorudan başlayarak soruların bittiği, bir uyarı veya önerinin sunulduğu yere kadar devam etmektedir. Şekilde üçüncü sorunun “evet” olarak cevaplanmasında bir uyarı ve arkasından üç adet öneri sunulurken, “hayır” olarak cevaplanmasında yalnızca “problem yok” uyarısı gösterilmektedir. Böyle bir durumda sıradaki soru ile algoritma yeniden başlayacaktır.



Şekil 5.3 Etkileşim Algoritması Sonucu

Şekil 5.3'teki akış şeması etkileşimin daha fazla olduğu bir soruyu temsil etmektedir. “Sistem nasıl bir ortamda çalışıyor?” çoktan seçmeli sorusuna verilecek yanıtlardan her biri için farklı sorularla akış devam etmektedir. Sorular farklı kollardan devam ederek seçimlere göre farklı öneri ve uyarılar sunmaktadır. Akış şemasında sorular doğrultusunda ilerlenmesinin yanı sıra geriye doğru dönülmek istenebileceği de göz önünde bulundurulmalıdır. Öyle ki, herhangi bir soruda bir önceki soruya geri dönülmek ve değişik bir cevapla devam edilmek istendiğinde akışın geriye çevrilmesi gerekmektedir. Yalnız bu geri dönüş sistemin çalışmasını bozmamalıdır. Dolayısıyla, algoritma yazılıma uygulanırken doğrudan bir önceki soruya geçmek yerine akışın başladığı soruya dönülmelidir. Örneğin, şekli ele alırsak, “E.K.’ya temas eden sıvı iletken mi?” sorusundan geriye dönülmek istendiğinde, bir önceki soru olan “E.K.’nın sıvıyı sızdırmayan bir koruması var mı?” sorusuna değil, akışın başı olan “Sistem nasıl bir ortamda çalışıyor?” sorusuna dönüş yapılmalıdır. Böylece soruya ait akış bozulmamış olacaktır.

5.2 Tabloların Algoritmaya Uyumu

Tasarım sisteminin temelinde, anahtar özelliklere ait bilgileri barındıran tablolar bulunmaktadır. Sistemdeki altı tablo tabloların oluşturulması bölümünde ayrıntılarıyla açıklanmıştır. Algoritmanın hazırlanmasından sonra tabloların algoritmayla uyumu irdelenmektedir. Tablolarda tutulan verilerin formatları analiz sürecinde belirlenmiştir. Fakat sistemin akış şeması çıkarılması sırasında farklı formatlara ihtiyaç duyulduğu veya gereksiz sütunların kaldırılması gerektiği anlaşılabilmektedir. Bu sebeple algoritma yalnızca tablolar ve veri formatları göz önünde bulundurularak tekrar ele alınmalıdır. Bu sırada önemle üzerinde durulması gereken konu, veri formatlarının hem akışa uyacak hem de etkileşimlere cevap verebilecek seviyede olup olmadığıdır. Örneğin, etkileşim tablosu tamamen sorular arası bağ kurmayı kolaylaştırmak üzere oluşturulmuştur. Bu tabloda bulunması gereken sütunlar başlangıçta yalnızca iki şıklı sorular (evet ve hayır) için tasarlanmışken, akış şeması incelendiğinde soruların yalnızca iki şıklı değil çoktan seçmeli de olabileceği görülmüş, bu sebeple, etkileşim tablosuna çoktan seçmeli sorular için iki sütun daha eklenmiştir. Bu şekilde yapılan

düzeltilmelerden sonra tablolar birbirleriyle ve algoritmayla uyumlu hale getirilmiştir. Altı tablodan biri olan etkileşim tablosu Şekil 5.1’de gösterilmiştir.

| etkilesim | soruno | sorukodu | secenek_ev | secenek_ha | secenek3 | secenek4 | sonraki | onceki |
|-----------|--------|----------|------------|------------|----------|----------|---------|--------|
| | 1 | S01 | C01 | P01 | | | S02 | S01 |
| | 2 | S02 | C02 | P01 | | | S03 | S01 |
| | 3 | S03 | U02 | P01 | | | C03 | S02 |
| | 4 | S04 | S05 | P01 | | | S06 | C03 |
| | 5 | S05 | P01 | U08 | | | S06 | S04 |
| | 6 | S06 | P01 | U10 | | | S07 | S04 |
| | 7 | S07 | S12 | P01 | | | SON | S06 |
| | 8 | S08 | U03 | P01 | | | S03 | S02 |
| | 9 | S09 | U05 | U06 | | | S04 | C03 |
| | 10 | S10 | P01 | S11 | | | S11 | C03 |
| | 11 | S11 | U07 | | | | S04 | S10 |
| | 12 | S12 | U09 | P01 | | | SON | S07 |
| | 13 | C01 | Z01 | Z02 | Z03 | | S02 | S01 |
| | 14 | C02 | Z08 | Z09 | | | S03 | S02 |
| | 15 | C03 | Z04 | Z05 | Z06 | Z07 | S04 | S03 |
| | 16 | U01 | O01 | O02 | O03 | | S02 | S07 |
| | 17 | U02 | O04 | O05 | | | C03 | S03 |
| | 18 | U03 | O06 | O07 | | | S03 | S02 |
| | 19 | U04 | O08 | | | | S04 | C03 |
| | 20 | U05 | O09 | | | | S04 | S09 |
| | 21 | U06 | O09 | | | | S04 | S09 |
| | 22 | U07 | O10 | O11 | | | S04 | S11 |
| | 23 | U08 | O12 | O13 | | | S06 | S05 |
| | 24 | U09 | O14 | O15 | | | SON | S12 |

Resim 5.1 Etkileşim Tablosu

5.3 Programlama Dilinin Seçilmesi

Elektromekanik ürün tasarımı için geliştirilen bu yazılım sistemi veritabanı bağlantıları ve etkileşimler içerdiğinden en uygun yazılım dili seçilirken veritabanı nesnelere kolay kullanılabilir ve anlaşılabilir olması, ayrıca son kullanıcı tarafından yapılacak eklemelere olanak sağlayacak esnekliğe sahip olması göz önünde bulundurulmuştur. Yazılım kısmının başlangıcında kolay ulaşılabilir olması, açık kaynak kodlu olması ve veritabanı nesnelere ile kullanılabilir olması sebebiyle Delphi yazılım dili tasarım için uygun bulunmuştur. Delphi ile yazılımın geliştirilmesi sırasında programın eksiklikleri görülmüş ve sisteme yeterli gelmediği anlaşılmıştır. Tablolar bölümünde söz edilen altı tablo Delphi programının içine entegre edilmiş veritabanı programında hazırlanmıştır. Dolayısıyla, tablolara dışarıdan erişim olanağı bulunamamış olup; güncelleme yapılamamaktadır. Programın kullanıcı isteklerini karşılayacak oranda esnek olmadığı, ayrıca programın yazımında yeterli nesne ve

syntaksi bulunmadığı tespit edildiğinden yazılım dilinin değiştirilmesi uygun görülmüştür. Delphiye benzer şekilde nesne tabanlı programlamaya olanak sağlayacak, fakat kullanım yönünden kolaylığı ve zenginliği bulunan C#.NET dili yeni yazılım dili olarak belirlenmiştir. C#, zengin içeriği, kullanım yaygınlığı, gelişen teknolojiye uyumluluğu ve sunduğu tasarım kolaylığı ile veritabanı nesnelere bileşenlerinin kullanıldığı bir sistem için uygun yazılım dili olarak seçilmiştir. Tablolar ise Access veritabanında hazırlanmıştır. Access Delphinin veritabanı idesinden farklı olarak tablolar arasında bağ kurmaya, tabloları tek bir veritabanı üzerinde toplamaya ve sütunların özelliklerini çeşitlendirmeye olanak tanımaktadır. Bu nedenle, C# ve Access kullanımı programa esneklik ve zenginlik kazandırmıştır.

5.4 Yazılımın Geliştirilmesi

5.4.1 Algoritmanın Yazılıma Dönüştürülmesi

Algoritma sistemin tamamını görmemizi sağlamaktadır. Algoritma üzerinden genel şekil belirlenir. Akış başla komutunun bulunduğu yerden başlayarak çeşitli noktalarda sona erer. Başlangıçtan bu bitiş noktalarına kadar her bir bölüm birer modül olarak adlandırılacak ve ileri ki bölümlerde incelenecektir. Yazılım dili olarak seçtiğimiz C# diline algoritmanın uygulanmasında öncelikle modüller belirlenmiştir. Belirlenen modüller doğrultusunda akış şemasına paralel olarak ana ekrandan başlanarak son kullanıcının kullanımına en uygun arayüz hazırlanmıştır. Resim 5.1’de gösterilen karşılama ekranı programa giriş niteliğinde ve iki kısma girişi sağlamaktadır.



Resim 5.2 Program Giriş Sayfası

5.4.1.1 Modüllerin Belirlenmesi

Algoritmanın başlangıcından her hangi bir bitiş noktasına kadar olan her bir bölüm bir modül olarak ele alınacaktır. Akış şeması incelendiğinde, mekanik kısmın ve etkileşim kısmının ayrı başlangıç noktaları olduğu görülmektedir. Bu bakımdan modüller belirlenirken ve algoritmanın yazılıma uyarlanmasında sistem iki parça olarak ele alınacaktır.

Mekanik kısım içinde, sistemin anahtar özellikleri olarak belirlediğimiz hız, maliyet, ağırlık ve boyut kavramları birer modül olarak karşımıza çıkmaktadır. Bu özelliklerin öncelik sıralaması program esnasında kullanıcı tarafından belirlenecek olup; akış şeması bu sırayı takiben devam edecektir.

Etkileşim kısmında, etkileşim tablosunda birinci soru olarak belirlenen sorudan problem analizi başlatılarak sorulara kullanıcının vereceği cevaplara göre programın akışı belirlenecektir. Sorular arasındaki bağ yine etkileşim tablosu sayesinde kurulmuştur. Mekanik ve etkileşim modülleriyle ilgili detaylı açıklama sonraki bölümlerde yapılacaktır.

Programın yazılma sürecinde, etkileşim kısmı analiz açısından uzun yazım açısından kısa bir süre gerektirmiştir. Analiz program yazımında en önemli süreçtir. Doğru bir akış şeması yazılımın temelidir. Algoritmanın oluşturulmasında sistem defalarca incelenmiş, farklı bakış açılarıyla değerlendirilmiştir.

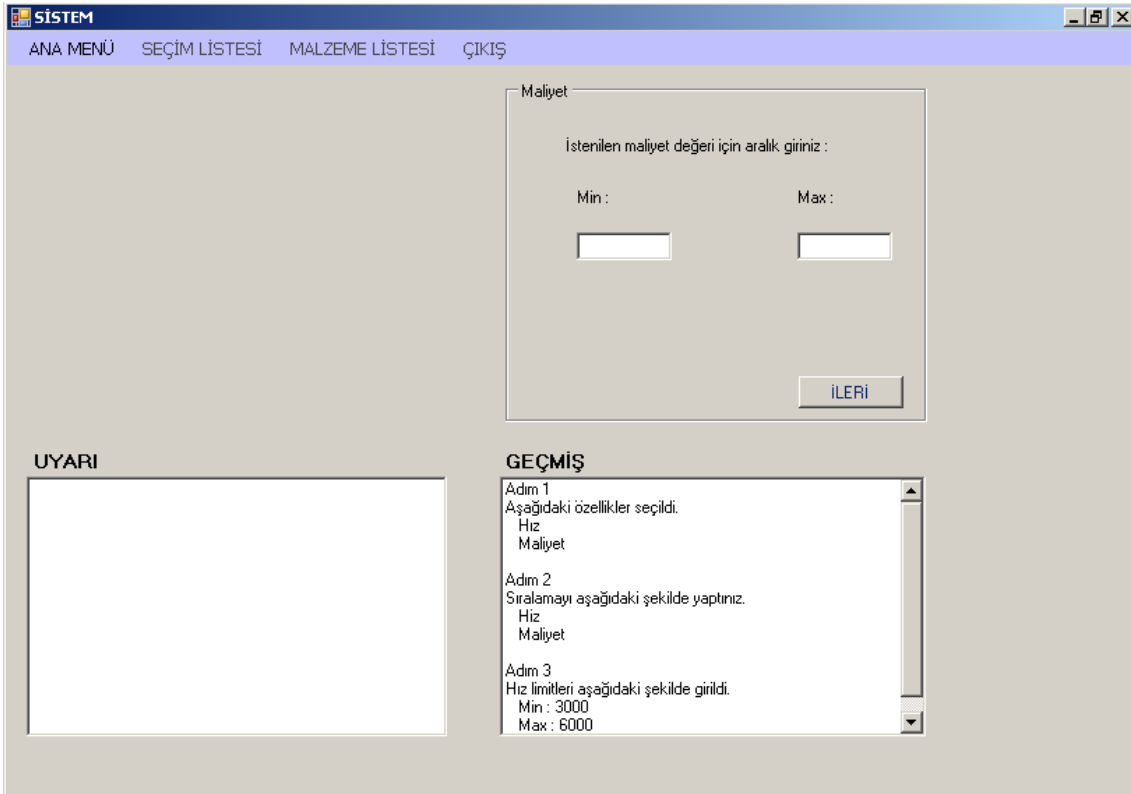
5.4.1.2 Mekanik Kısım

Sistemin mekanik kısmı, elektromekanik sistemlerde kullanılan malzemelerin, anahtar özelliklerine göre incelenmesi ve sistem içinde birbirleriyle uyumunu ele almaktadır. Örneğin, model olarak seçtiğimiz robotu çalıştıracak güce sahip bir motorun kullanıcı istekleri doğrultusunda belirlenen boyut değerleriyle uyumlu olup olmadığı bu bölümde değerlendirilecektir.

Modül giriş ekranından “Mekanik Kısım” seçeneğinin seçilmesiyle başlar. Kullanıcının belirlenen anahtar özelliklerden istediklerini seçebileceği Resim 5.2’de gösterilen ekran görüntülenir. Burada yapılan seçim malzemeler arasından yapılacak seçime ışık tutacaktır. Sistem tarafından ele alınan dört anahtar özellik vardır; hız, maliyet, ağırlık ve boyut. Yazılımın uygulanacağı sektöre veya kullanılacak malzemeye göre bu özellikler özelleştirilebilir veya artırılabilir.

Resim 5.3 Mekanik Kısım

Bir sonraki adımda, belirlenen dört özellik arasından yapılacak seçimlerin öncelik sıralaması yapılması istenmektedir. Öncelik sırasına göre kullanıcı bu özellikler için istediği değer aralıklarını belirlemektedir. Belirlenen değerlerde veya sistemin kullanılmasında oluşabilecek sorunlar uyarı bölümünde belirtilecektir. Bu adımların her birinde yapılan işlemler Resim 5.3'te gösterildiği üzere ekranın sağında bulunan geçmiş listesinde tutulmaktadır.



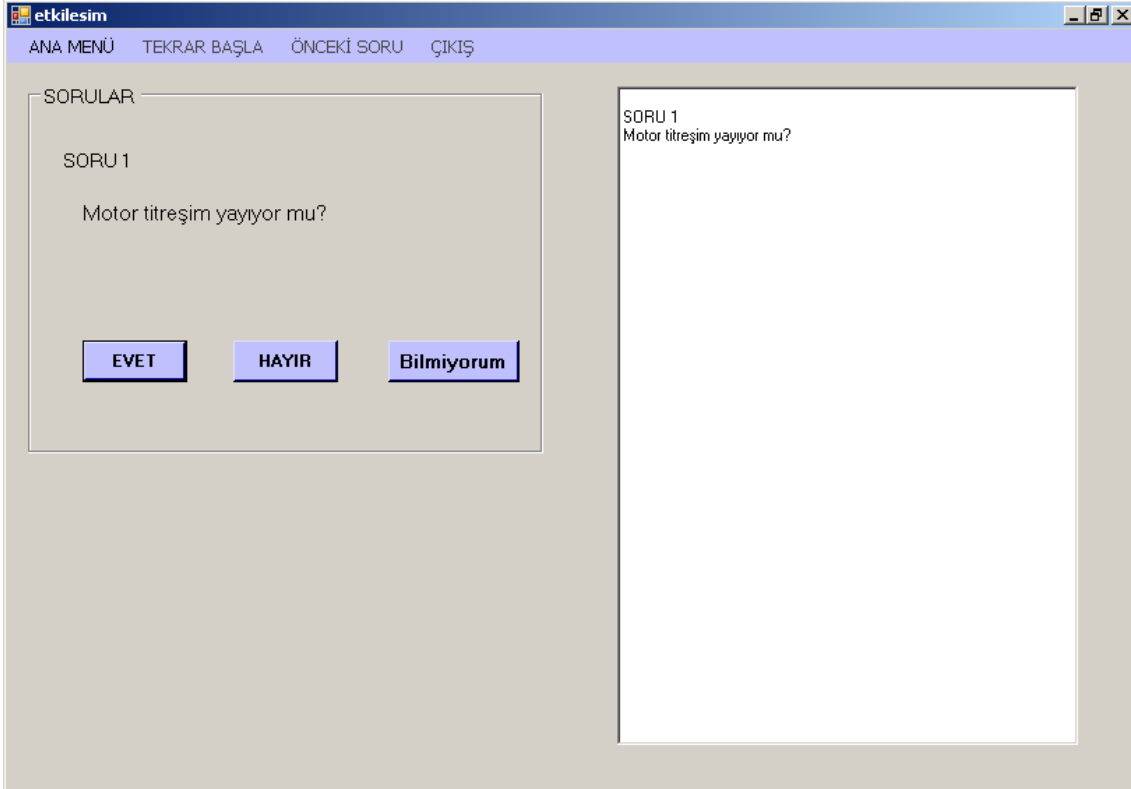
Resim 5.4 Mekanik Kısım Kriterlerin Belirlenmesi

Mekanik kısımda amaç, oluşturulacak sisteme uygun, ayrıca birbiriyle uyumlu parçaların seçilmesine yardımcı olmaktır. Kullanıcı, program içerisinde karşılaştığı sorulara vereceği cevaplarla malzeme seçimini yapmaktadır. Son aşamada kullanıcının belirlediği kriterlere uygun malzemeler liste halinde sunulacaktır.

5.4.1.3 Etkileşim Kısmı

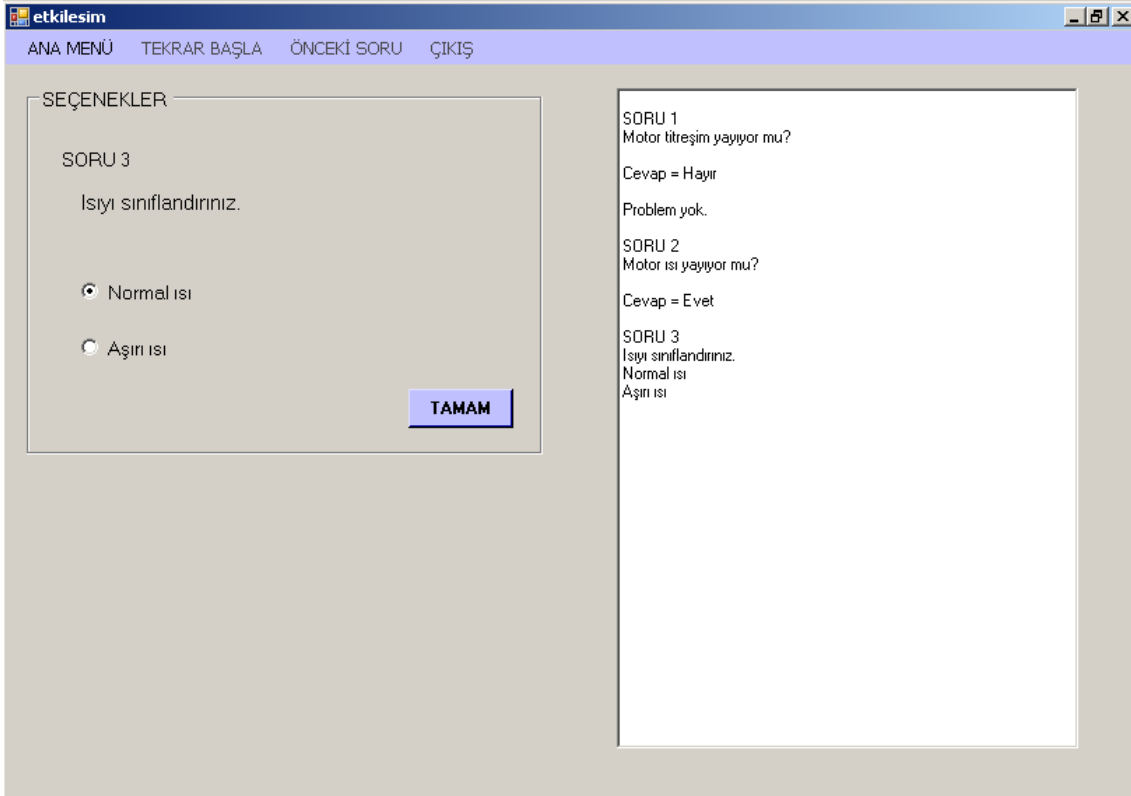
Sistemin etkileşim kısmında, mekanik parçaların birbirleriyle etkileşimi sonucu ortaya çıkabilecek sorunlara çözüm bulmaya yönelik uyarı ve öneriler

görüntülenmektedir. Programın bu bölümü tamamen sorulara verilen cevaplara göre şekillenmektedir. Algoritmanın oluşturulması kısmında anlatıldığı gibi, sistemin çalışması esnasında karşılaşılabilecek problemler analiz edilerek kullanıcıyı yönlendirecek sorular belirlenmiş, sonucunda soru ve etkileşim tablosu oluşturulmuştur. Soru tablosu soru kodlarına göre soru metinlerini içermektedir. Etkileşim tablosu ise soruların akışını yönetmekte, etkileşim akış şemasını oluşturmaktadır. Belirlenen sorulardan birisi ilk soru olarak seçilmiştir. İşlem akışı Resim 5.4'te gösterilen bu sorudan başlamakta, algoritmaya göre bir veya birden fazla yerde son sorular ile bitmektedir. Bütün sorular kendisinden önce, sonra ve kendisine verilecek cevaba göre daha sonra gelecek sorulara etkileşim tablosu sayesinde bağlıdır. Örneğin, sistemimizde ilk soru olan “Motor titreşim yayıyor mu?” sorusuna verilecek cevap evet olması halinde yeni bir soru gelirken, hayır olması halinde problem yok uyarısı ekrana gelecektir. Soruya bilmiyorum cevabı verilir ise bu kez yeni bir soruyla akış devam edecektir. Bütün bu akışı program etkileşim tablosu yardımıyla sağlamaktadır. Bu bakımdan sisteme yeni bir soru eklenmek istendiğinde etkileşim tablosunda gerekli bağlarının kurulması ve öncelikle soruya çeşidine göre bir soru kodu verilmesi gerekmektedir.



Resim 5.5 Etkileşim Modülü

Programı yazılım yönünden ele alacak olursak, seçtiğimiz nesne tabanlı yazılım dili olan C#'ın çok fonksiyonlu yapısından faydalanılmıştır. Her bir bölüm farklı formlarla hazırlanmış, her bölümde yapılan işlemler liste halinde tutulmuş, sistem çıkışında bu liste tarih ve saati ile birlikte kaydedilmiştir. Mekanik bölümde ve etkileşim bölümünde farklı pencereler aynı form üzerinde düzenlenmiştir. Yazılım daha sonra geliştirilebilecek şekilde basit, anlaşılır ve anlamlı şekilde düzenlenmiştir. Örneğin, değişkenlere temsil ettiği nesneye göre anlaşılabilir isimler verilmiştir. Menüler kullanıcının alışkın olduğu şekilde üst satırda, okunabilir harflerle ve kullanım sıralamasına uygun yerleştirilmiştir. Bütün ekranlara ana menüye giriş ve sistemden çıkış menüleri konulmuştur. Arayüzlerin oluşturulmasında son kullanıcı etkin rol oynamıştır. Sistem kullanıcıdan gelen geri dönüşlerle tekrar şekillendirilmiştir.



Resim 5.6 Etkileşim Modülü Soru Örneği

Sistemde yer alan sorular ve verilen cevaplar adım adım ekranın sağında yer alan listede Resim 5.5'te gösterildiği üzere listelenmektedir. Daha sonra bu liste sistemden çıkılması halinde tarihi ve saati ile birlikte kaydedilmektedir.

Etkileşim kısmı tamamen kullanıcıya sorun çözmede yardımcı olmak amacıyla hazırlanmıştır. Fakat program hazırlanırken fark edilmeyen problemler veya eksik olan uyarılar bulunabilmektedir. Bu gibi eksikler uzun analiz sürecinde yakalanmaya çalışılmış, daha sonra yazılım esnasında değerlendirilmiş, en son olarak ta test aşamasında farklı kullanıcıların sistemi kullanması sağlanarak giderilmeye çalışılmıştır.

5.5 Test Aşaması

Sistem hazırlanan algoritmaya uygun olarak yazılıma dönüştürülmüştür. Kullanıcı arayüzleri kullanım kolaylığı göz önüne alınarak hazırlanmıştır. Fakat sistemin kullanılacağı sektöre göre gereksinimler çeşitlilik göstereceğinden farklı kullanıcılarla test ortamı sağlanmıştır. Hedef kitle olan son kullanıcının, hazırlanan arayüzler aracılığıyla sistemi test etmesi istenmiştir. Böylece teorikte göz ardı edilen bazı özellikler belirlenebilmektedir. Test aşamasında, göz önüne alınan kriterler;

1. Kullanıcının kullanma kılavuzuna ihtiyacı olmadan programı kullanabilmesi,
2. Arayüz tasarımlarının kullanıcıya hitap etmesi,
3. Ekran yazılarının okunabilir olması,
4. Ekranlar arası geçişin kolay olması,
5. Geriye dönüşün olması,
6. Algoritmanın doğru çalışması,
7. Programın yazılım kaynaklı hata vermemesi,
8. Sonucun kullanıcının isteğine uygun formatta olması,
9. Uyarı ve önerilerin çözüme götürmesi,
10. Programın geliştirilebilir olmasıdır.

5.6 Sorunların Giderilmesi

Sistem çeşitli kullanıcılar tarafından test edilmiş, kriterler değerlendirilmiştir. Sonuçta yazılım üzerinde aşağıdaki değişiklikler yapılmıştır:

1. Arayüz tasarımında çok renkli, karışık yapıdan vazgeçilerek sade ve kullanışlı ekranlar kullanılmıştır.
2. Ekran yazılarında çok büyük olmayan, ince ve kesin yapılı harflerden oluşan bir yazı stili seçilmiştir.

3. Ara ekranlara ana menüye geçiş, bir önceki ekrana dönüş ve sistemden çıkış için menüler eklenmiştir.
4. Algoritmanın etkileşim ekranında bazı sorularda takıldığı tespit edildiğinden yazılım güncellenmiştir.
5. Programın mekanik kısmında tabloların yüklenmesinde resim içeren nesnelerin yüklenmesi sırasında yazılımın hata verdiği görülmüş olduğundan resim nesnelere yüklenmesi kaldırılmıştır.
6. Her bir bölümde yapılan işlemler adım adım listelenerek programdan çıkış sırasında programın kayıtlı olduğu dosyaya tarih ve saati isim olarak kullanarak kayıt edilmektedir.
7. Önerilerin eksik kaldığı bölümlerde uyarılar eklenmiş veya sorular çoğaltılarak sorunun analizi kuvvetlendirilmiştir.
10. Programın farklı sektörlerde kullanılabileceği, ayrıca daha sonra parça sayısının veya etkileşimle ilgili soruların artabileceği düşünülerek sistemin geliştirilebilmesi için bir arayüz oluşturulmasına, ayrıca ayrı bir modülün bu yönde geliştirilmesine karar verilmiştir.

6. TARTIŞMA VE SONUÇ

Tasarım ekiplerinin çalışma şekillerini, yeteneklerini ve araçlarını içeren örnek olaylar vasıtasıyla, elektromekanik ürünlerin robot tasarımlarını önemli derecede etkileyen temel özellikler belirlenmiştir. Hiyerarşide aşağıya doğru gidildikçe, bunlar üretim/imalat süreci ve montajın temel özellikleri ile bağlanacaktır. Hiyerarşi içinde, ana anlaşmazlıklar belirlenmiş, ağırlıklandırılmış ve uygun düğümlerle temsil edilmiştir. Bu, araştırmada geliştirilecek olan ürün geliştirme uzman sisteminde kullanılan anlaşmazlıklar süreç yeterliliğinin ve ürün gereksinimlerinin belirlenmesinin temellerini oluşturacaktır. Proje boyunca, elektromekanik ürün tasarım sisteminde var olan disiplinler koordineli olarak çalışmış ve bunun önemi vurgulanmıştır. Disiplinler arası ürün tasarımında karşılaşılan sorunlar proje bazlı ele alınarak çözümler bütün projelere uyarlanabilecek şekilde sunulmaya çalışılmıştır. Analiz sürecinden başlayan sistemin tasarım kısmı, uygulama bölümünde sistem gereksinimleri doğrultusunda bir yazılım programına dönüştürülmüştür. Programın amacı kullanıcıya gereksinimlerini belirlemede yardımcı olmak ve sistemde karşılaşılabilecek sorunlara çözüm önerileri sunmaktır. Program geliştirilebilir ve diğer sistemlere adapte edilebilir olmasına karşın başlangıçta tek bir örnek model üzerinden gidilerek oluşturulmuştur. Test süreciyle de kullanıcının geri bildirimleri değerlendirilmiştir. Sonuç olarak, gerek tasarım gerekse üretim kısmıyla bir bütün olarak ele alınan sistemin bütün aşamaları, bütün disiplinleri ve disiplin uzmanlarıyla değerlendirilerek sistemin sorunlarını çözmeye yönelik bir yazılım programı oluşturulmuştur.

Tübitak tarafından desteklenen proje 2 yıllık olarak planlanmış olmasına rağmen iş şartları nedeniyle son döneminde yakından takip edilememiştir. Dolayısıyla, test aşamasında yeterince uzmandan ve kullanıcıdan destek alınamamıştır. İleriki aşamalarda, test aşamasının tekrarlanarak sistemin kapsamının genişletilmesi, sorunları çözmeye yönelik soruları içeren veritabanının geliştirilmesi ve tasarım bölümünde yer alan yönlendirmelerin daha yararlı hale getirilmesinin uygun olacağı değerlendirilmektedir.

7. KAYNAKLAR

- Anonymous. 2006. Overcoming the Top Five Challenges in Electromechanical Product Development. Parametric Technology Corporation (PTC).
- A, Gayretli (2005) "Constraint-Based Conflict Management in a Cooperative Design Environment" Invited paper, Electronic Journal of Machine Technologies, 2005 (3) 1-9.
- Bliznakov, PI, Shah JJ and Urban SD (1996), "Integration Infrastructure to Support Concurrence and Collaboration in Engineering Design," Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference, Paper No. 96-DETC/E1M-1420, Irvine, CA, August.
- Breedveld, P. C (2004) "Port-Based Modelling of Mechatronic Systems", Mathematics and Computers in Simulation, Vol. 66, pp. 99-127.
- Concurrent Engineering : Contemporary issues and modern design tools, 1993, Chapman & Hall H.R Parsaei and W.G. Williams(Eds).
- Daniei J and Director S. W ., 1989, " An Object Oriented Approach to CAD tool Control within a design framework", proc of the 26th ACM/IEEE Design Automation conf, Las Vegas, June 25-27, pp.197-202.
- Eppinger SD, Whitney DE, Smith RP, Gebala DA (1994) "A Model-Based Method for Organising Tasks in Product Development", Research in Engineering Design-Theory Applications and Concurrent Engineering, Vol 6, No. 1 pp 1-13.
- Farr, R, Gayretli, A, Gindy, N and Speller, H (2002)" Collaborative Development using EMProDS: The Electromechanical Product Design System " IMechE -Journal of Manufacture -Part B, Vol. 216, pp. 453-457.
- Farr, R, Gayretli, A and Speller, H, (2002)" Nurturing the Knowledge Base in Distributed Organisations" in The Proceedings of 2nd International Conference on Responsive Manufacturing, Turkey, 26-28 June 2002.
- Gayretli, A, Farr, R, Speller, H and Gindy, N N Z, (2001)" A Product Development Tool for Integrated Electromechanical Product Design" The Proceedings of ICE'2001- 7th International Conference on Concurrent Enterprising, Germany, 27-29 June 2001.

- Hamdi M S, and Kaiser , K (1997) " Adaptable local level arbitration of behaviours",
Proc of the first International Conf. On Autonomous Agents. August 97. Marina
del Ray, CA. USA.
- Hardwick, M and Loffredo D (1995), "Using EXPRESS to Implement Concurrent
Engineering Databases," "Proceedings of the 1995 ASM E C omputers in
Engineering Conference and the Engineering Database Symposium, Boston, MA,
September.
- H.V.D. Parunak, A.D. Baker, and S.J. Clark, (1997) "The AARIA Agent Architecture:
An Example of Requirements-Driven Agent-Based System Design" in Proceeding
of the First International Conference on Autonomous Agents (ICAA'97), Marina
del Rey, CA, February 6-8.
- KIM TS, Han SH and Shin YJ (1996), "Product Data Management Using AP203 of
STEP Standard," "Proceedings of the 1996 ASME Design Engineering Technical
Conferences and Computers in Engineering Conference, Paper No. 96-
DETC/DAC-1069, Kirvine, CA, August.
- Kreps T H , (1996) "3-D-MEMCAD and computer-Aided planning tools", Instiute for
Manufacturing Automation and Production systems- FAPS- University of
Erlangen Germany- www.faps.uni-erlangen.de.
- Krishnan V, Eppinger S, Whitney DE (1995) "Accelerating Product Development by
the Exchange of Preliminary Product Design Information" Journal of Mechanical
Design Vol 17 No 4, pp 491-498
- Lin, L. j. (1992). "Self- improving reactive agents based on reinforcement learning,
planning and teaching, Machine learning, 8 (3/4), pp 293-321.
- R. Farr, A Gayretli, H. Speller and N. Gindy (2002)" A Software Tool for Integrated
Electromechanical System Design" in The Proceedings of 2nd International
Conference on Responsive Manufacturing, Turkey, 26-28 June 2002.
- R. Farr, A Gayretli, A Hodgson and N. Gindy (2003) "Virtual Teams and
Electromechanical Product Design" International Journal of Advanced
Manufacturing Systems, Vol. 6, Issue 2, pp. 49-63.
- Shah M, Jeon DK, Urban SD, Bliznakov P and Rogers M (1996), "Database
Infrastructure for Supporting Engineering Design Histories," Computer-Aided
Design, 28(5).

Special issue: Agent based Manufacturing, Journal of Intelligent Manufacturing, Vol 9, No 2 April (1998).

Whitney DE, (1996) "Why Mechanical Design Cannot be Like VLSI Design" Research in Engineering Design-Theory Applications and Concurrent Engineering Vol 8, No.3 ppl25-138.

Wood III, Agogino \VH and AM (1996), "Case-Based Conceptual Design Information Server for Concurrent Engineering" Computer-Aided Design, 28(5).

7.1. İnternet Kaynakları

Erişim Tarihi

| | |
|---|------------|
| 1- http://www.referansgazetesi.com/haber.aspx? HBR_KOD=29172&KTG_KOD=338 | 19.12.2008 |
| 2- http://www.scribd.com/doc/3098276/ Yazlm-Gelitirme-ve-Analiz | 07.01.2009 |
| 3- http://www.dogancakmak.com/post/2008/10/31/ WEB-TABANLI-YAZILIM-PROJELERI-ICIN- ADDIE-MODELINDEN-ESINLENILEREK- HAZIRLANMIS-ANALIZ-TASLAGI.aspx | 07.01.2009 |
| 4- http://ppm-turkiye.blogspot.com/2007/11/endstriyel- it-sistemlerinde-sistem.html | 07.01.2009 |
| 5- http://www.endustrimuhendisim.net/file.axd?file= sistemanalizi.pdf | 07.01.2009 |

ÖZGEÇMİŞ

Adı Soyadı : Yasemin BOY
Doğum Yeri : Afyon
Doğum Tarihi : 14.04.1983
Medeni Hali : Evli
Yabancı Dili : İngilizce

Eğitim Durumu (Kurum ve Yıl)

Lise : Afyon Süleyman Demirel Fen Lisesi
Lisans : Anadolu Üniversitesi Bilgisayar Mühendisliği (İng)
Yüksek Lisans: Afyon Kocatepe Üniversitesi Bilgisayar ABD

Çalıştığı Kurum/Kurumlar ve Yıl

1. İstanbul Nüfus ve Vatandaşlık Müdürlüğü
05.2007-... Programcı İdari İşler İstanbul (Avr.)
2. Afyonkarahisar Kocatepe Üniversitesi
09.2006-05.2007 Proje asistanı Teknik Eğ. Fak. Afyonkarahisar
3. Afyonkarahisar Kocatepe Üniversitesi
07.2005-09.2005 Stajyer Teknik servis Afyonkarahisar
4. ETİ A.Ş.
07.2004-08.2004 Stajyer Yazılım Eskişehir

Diğer konular

Sınavlar

- | | | | |
|---------|---------|-------------------------|----|
| 1. LES | 11.2005 | sayısal | 71 |
| ALES | 03.2008 | sayısal | 89 |
| 2. ÜDS | 03.2006 | fen bilimleri-ingilizce | 70 |
| 3. KPDS | 05.2006 | fen bilimleri-ingilizce | 67 |
| 4. KPSS | 07.2006 | kpssp3 | 82 |

Bilgisayar Bilgileri

| | | |
|-----|--------------|------|
| 1. | Visual Basic | iyi |
| 2. | VB .NET | iyi |
| 3. | C | iyi |
| 4. | C# .NET | iyi |
| 5. | Delphi | orta |
| 6. | ADO .NET | orta |
| 7. | SQL | orta |
| 8. | Access | iyi |
| 9. | Dreamweaver | iyi |
| 10. | HTML | orta |
| 11. | XML | iyi |
| 12. | Photoshop | orta |
| 13. | Flash | iyi |
| 14. | Fireworks | iyi |

Projeler

1. Lisans Tezi (11.2005 / 06.2006)

Çeşitli markaların bilgisayar ve donanım malzemelerini satan ve aynı zamanda teknik servis hizmeti veren bir firmanın stok takip, müşteri ve ürün kaydı, faturalama, satış ve teknik servis kayıtlarının tutulması gibi temel işlemlerini kapsayan; görsel programlamaya ve veritabanı yönetimine dayalı bir program yazımı. Programın yazılımlarında C# .NET 2005 ve MS Access kullanılmıştır.

2. Tübitak Projesi (09.2006 / 09.2008)

Elektromekanik Ürünler İçin Bir Tasarım Sistemi Geliştirilmesi

Bu projenin amacı elektromekanik ürünlerin anahtar karakteristiklerine dayalı olarak çalışan bir ürün sistemi meydana getirmektir. Sonuç olarak, elektromekanik ürünlerin temel özelliklerin tespiti, montaj ve imatları ile ilgili temel özellik ve ilişkilerin tanımlanması ve dökümantasyonunun yapılması ve sonuçta bir elektromekanik tasarım el kitabının hazırlanması; disiplin sınırlarını anlayabilen ve bu sınırları geçebilen ürün geliştirme sırasındaki karmaşık ilişkileri destekleyecek yeterliliğe sahip bilgisayar

destekli bir sistem; ürün geliştirme aracının kurallarını ve ilkelerini açıkça belirleyen yeni bir metodoloji ortaya çıkarmak hedeflerdir.

3. Uzaktan Eğitim Projesi (09.2006 / 09.2008)

E-öğrenmeyi örgün öğrenime alternatif olabilecek veya zaman ve mekan birlikteliği olmadan eğitim sağlayabilecek teknolojiyi kullanan bir eğitim sistemi olarak nitelendirebiliriz. Afyon Kocatepe Üniversitesi Enformatik Bölümü bünyesinde gerçekleştirilen projede amaç uzaktan eğitim sistemi ile eğitim veren başta yüksek lisans programları olmak üzere bölümler açmaktır.

Eğitimler

| Eğitimin Adı | Kurum | Süresi |
|---------------------|------------|---------|
| • Flash Eğitimi | Bilge Adam | 2 hafta |
| • Photoshop Eğitimi | Bilge Adam | 3 gün |

Belge Ve Sertifikalar

- Anadolu Üniversitesi Onur Belgesi
2002-2003 Bahar Dönemi
2005-2006 Güz Dönemi
- Microsoft Bilgisayar İşletmenliği Sertifikası
- Akademik Bilişim '07 Konferansı Katılım Belgesi (sunucu)
- Bilge Adam Photoshop CS2 Sertifikası
- Bilge Adam Flash 8.0 Sertifikası