

TESEKKÜR

Tez çalışması boyunca yardımlarını esirgemeyen, bilgi ve tecrübelerini paylaşmaktan kaçınmayan, yol gösterici nitelikteki tüm desteklerinden ötürü basta tez danışmanım Prof. Dr. Burhanettin CAN'a, bu tezin ortaya çıkmasını sağlayan ve kendisini her asamada yanımda buldugum sayın Prof. Dr. Ercan Öztemel'e, tez çalışması boyunca çalışmalarımı izleyen ve yapıcı görüş ve eleştirilerini bildiren tez izleme jürime, bu çalışma boyunca devamlı yorumlarını bildiren tüm arkadaşlarıma ve bu çalışmanın her aşamasında, her zaman sabırla yanımda buldugum aile fertlerimin hepsine sabırlarından dolayı tesekkür ederim.

Mart, 2006

Ali GÜRBÜZ

İÇİNDEKİLER

TESEKKÜR	I
İÇİNDEKİLER	II
ÖZET.....	V
ABSTRACT.....	VIII
YENİLİK BEYANI.....	XI
KISALTMALAR LİSTESİ.....	XIII
SEKİLLER LİSTESİ.....	XIV
TABLolar LİSTESİ.....	XVII
BÖLÜM I.....	1
GİRİŞ VE AMAÇ	1
I.1 GİRİŞ	1
I.2 TEZİN GEREKÇESİ	2
I.3 TEZİN AMACI.....	5
I.4 KAPSAM.....	5
BÖLÜM II	7
GENEL BİLGİLER.....	7
II.1 Kavramsal Modelleme	7
II.1.1 Model ve Modelleme	7
II.1.2 Kavramsal Modellemenin Amacı	8
II.1.3 Sistem Yaşam Döngüsü İçerisinde Modeller.....	9
II.1.4 Kavramsal Modelleme Nedir?	10
II.1.5 KM Nasıl Gerçekleştirilir?.....	12
II.1.6 KM Nereelerde Kullanılır?	15
II.2 Kavramsal Modelleme Teknikleri	15
II.2.1 Geleneksel Kavramsal Modelleme Teknikleri.....	16
II.2.2 Gelişmiş Kavramsal Modelleme Teknikleri	29
II.2.3 Kavramsal Modelleme’de Önceki Çalışmalar	36
II.3 Yapay Zeka	37
II.3.1 Kural Tabanlı Uzman Sistemler:.....	38
II.3.2 Kural Tabanlı Uzman Sistem Yapısı	41
II.3.3 Kural Tabanlı Uzman Sistem Süreci.....	42
II.3.4 Çıkarım Mekanizması Teknikleri	43

II.4	Zeki Kavramsal Modelleme'nin Üstünlüğü.....	45
II.5	KM Teknikleri ve Mevcut Çalışmaların İrdelenmesi	46
BÖLÜM III.....		48
KAVRAMSAL MODEL GELİSTİRME SÜRECİ		48
III.1	Bilinen Kavramsal Modelleme Süreci.....	48
III.2	In-CM Kavramsal Model Gelistirme Süreci.....	49
BÖLÜM IV		55
KAVRAMSAL MODELLEME YAKLASIMI VE REFERANS MODEL.....		55
IV.1	Kavramsal Modelleme Yaklasimi.....	55
IV.2	Referans Model.....	55
BÖLÜM V.....		60
IN-CM :GENEL YAPI		60
V.1	In-CM Zeki KM Gelistirme Yazilimi'nin Amaci.....	60
V.2	In-CM Zeki KM Gelistirme Yazilimi	60
V.3	In-CM Bilgi Tabani.....	61
V.3.1	In-CM KM Gelistirme Sürecinde Referans KM.....	62
V.3.2	Kurallar	65
V.3.3	Kullanici Kavramsal Modeli	67
V.4	In-CM Bilgi Tabani Editörü.....	67
V.5	In-CM Çıkarım Mekanizması.....	68
V.6	In-CM Grafik Editörü	69
BÖLÜM VI.....		71
ÖRNEK UYGULAMA.....		71
VI.1	Gelistirilecek olan “Kontrol Mühendisligi Egitimi” KM ‘in Amaci	71
VI.2	Geleneksel Yollarla KM Gelistirme	72
VI.3	In-CM Yazilimi Ile Kavramsal Model Gelistirme	74
VI.4	Gelistirilen Sistemin Zeki Özellikleri	100
VI.5	In-CM Zeki KM Gelistirme Yazilimi ve Incelenen Yazilimların Karsilastirilmesi	100
BÖLÜM VII.....		104
SONUÇLAR		104
BÖLÜM VIII.....		107
TARTISMA VE DEGERLENDIRME.....		107
KAYNAKLAR		108

EK-A.....	113
A.1 In-CM Ana Program Tanitimi	113
A.2 In-CM Bilgi Tabani Editörü Tanitimi	113
A.2.1 BTE’de Gerçekleştirilen İşlemler	115
A.3 In-CM Çıkarım Mekanizması Tanitimi	124
A.3.1 Çıkarım Mekanizması’nda Gerçekleştirilen İşlemler	126
A.4 In-CM Grafik Editör Yazılımı Tanitimi	129
ÖZGEÇMİS.....	131

ÖZET

ZEKİ KAVRAMSAL MODEL GELİSTİRME ARACI

Bu tez çalışmasında, sistemleri en etkin şekilde geliştirmeyi mümkün kılacak fiziksel tasarımdan önce kavramsal olarak sistemleri tasarlayarak sistem elemanları ve birbirleri ile ilişkilerini önceden görerek değerlendirmeyi sağlayacak bir modelleme yaklaşımı geliştirilmiştir. Bu modelleme yaklaşımını esas alan ve disiplinler arası kavramsal modelleri gerçekleştirebilen bir yazılım oluşturulmuştur. In-CM adı verilen bu yazılımın en temel özelliği kullanıcıların sistemleri geliştirirken içindeki bulundukları karmaşık durumu kolaylaştıracak zeki özelliklere sahip olmasıdır.

Tezde öncelikle kavramsal modelleme konusu tartışılmıştır. Özetle, kavramsal modelleme, sistem mühendisliği, bilgi mühendisliği ve yazılım mühendisliğinde kullanılan, genel olarak, fiziksel veya sosyal sistemlerin anlaşılabilirliğini kolaylaştırmak için, sistemin, alt parçalara bölümlendirilerek (biçimsel olarak -teknik terimleri kullanmadan-) ifade edilmesi işlemidir. Literatür incelemesinde, öncelikle mevcut kavramsal modelleme süreci ve teknikleri incelenmiştir. Bu incelemede kavramsal modelleme tekniklerinin geleneksel ve gelişmiş olmak üzere iki kategoride toplandığı görülmüştür. Mevcut kavramsal modelleme teknikleri aşağıdaki kısıtları içerdikleri tespit edilmiştir:

- Bu teknikler ile problem alanı objektif bir şekilde tüm yönleri ile gösterilememektedir. Bu teknikler, üzerinde çalışılan problemin sadece bir yönünü ele almakta ve o problemin o yönü ile ilgili kavramsal modeli ortaya koymaktadır. Bu çalışmada önerilen modelleme yaklaşımı ile bu soruna çözüm üretilmiştir.
- Mevcut teknikler “*problemin ne olduğundan*” çok “*problemin nasıl çözümleneceğini*” içermektedir. Bu çalışmada problemin doğal niteliklerini dikkate alan ve “*ne olduğunu*” irdeleyebilen bir modelleme yaklaşımı önerilmiştir.
- Bilinen tekniklerin kullandıkları grafiksel gösterimler nesne yönelimli analiz sembolojileridir. Bundan dolayı, çizilen modeller, problemin çözümüne odaklı bir şekilde oluşturulmaktadır. Bu da problemin tam

anlasilmadan çözümü üzerine çalışılması demektir. Bu gerçeği dikkate alarak bu çalışmada problemin tam olarak anlasilmesini problemin alt elemanlarının ve onların ilişkilerinin neler olması gerektiğini ortaya koyan ve gösterebilen bir modelleme yaklaşımı ve ilgili gösterim metodolojisi oluşturulmuştur.

Bu çalışmada önerilen ve yukarıda açıklanan modelleme yaklaşımı herhangi bir sistemi veya herhangi bir problemi ilgili alan içerisindeki aktörler (eylemleri gerçekleştirenler), süreç (işlemler dizisi), görev (temel amaç), alt yapı (yazılım, donanım vs.) gibi öğelerini içerecek şekilde tasarlanmıştır. Bu yaklaşım ile problem alanı bir bütün halinde birkaç açıdan kapsanacak şekilde modellenmiş ve ilgili kavramsal modelin oluşturulması mümkün olmuştur. Önerilen yaklaşım bilinen kavramsal modelleme sürecine de ters düşmemektedir. Ancak bu yaklaşım süreci de güncelleyecek öneriler ortaya çıkartmıştır. Böylece, güncellenen kavramsal modelleme süreci bu alanda çalışmalara katkı sağlayacaktır. Önerilen yöntemin en önemli faydası herhangi bir alanda o alana özgü değişik özelliklerde kavramsal modelleme yapmayı kolaylaştıracak bir *Referans Kavramsal Model (RKM)*'in oluşturulmasını mümkün kılmasıdır. RKM problem alanının tamamını kapsamakta olup oldukça kapsamlı bir meta modeldir. Kullanıcı kendi istek ve ihtiyacı doğrultusunda referans modelden kullanıcıya özel daha dar kapsamlı modeller oluşturabilir.

Kavramsal modelleme süreci güncellenip kavramsal modelleme yaklaşımı oturtulduktan sonra bu yaklaşımı kullanan ve disiplinler arası kavramsal modelleme gerçekleştirebilen bir uzman sistem yazılımı geliştirilmiştir. Bu yazılım, uzman sistemlerin çıkarım mekanizması, bilgi tabanı, gösterim mekanizması gibi temel elemanlarını içermektedir. Program içerisinde kullanıcı isteklerini yapısal hale getirebilmek için geliştirilmiş olan RKM bilgi kalibi (pattern) kullanılmıştır. Bu kalibin yanında kalip içerisindeki bilgiler arası kural tanımlamaları da bulunmaktadır. RKM'in ve kuralların tanımlanması için In-CM Bilgi Tabanı Editör yazılımı geliştirilmiştir. Alan uzmanı bu yazılım modülü sayesinde istediği disiplin için RKM oluşturabilmekte ve kurallar tanımlayabilmektedir.

RKM'den kullanıcıya özel bir kavramsal modelin oluşturulması In-CM Çıkarım mekanizması yazılımı ile gerçekleştirilebilmektedir. Kullanıcıya kavramsal model geliştirilir iken geliştirilmiş olan uzman sistem tarafından destek verilmekte ve

yol gösterilmektedir. Bu yazilimin ıktisi ile de In-CM Grafik Editr'nde kullanıcı kavramsal modeli izilmektedir.

Bu alısmada yukarıda aıklanan zelliklerde geliştirilmiş olan “In-CM Zeki Kavramsal Model Geliştirme” yazılımı gerek bir problem zerinde alıtırılarak nerilen yaklaşıml ve yntemlerin (RKM gibi) uygulanabilirliği gsterilmiştir. alısmanın katkısını aık olarak gsterebilmek amacı ile mmkn olduğunca basit bir problem seilmştir. Ancak geliştirilen yntem problemin karmaşıklığından olumsuz ynden etkilenmeyecek durumdadır ve her trl probleme uygulanabilir niteliktedir. Bu alısmada rnek olarak herhangi bir niversitede kontrol mhendisliği eğitimi verebilecek bir blm aılabilmesi iin nceden kavramsal olarak bu blm aıp sistem elemanları arasındaki ilişkileri izleyebilecek bir model geliştirilmiştir.

Tezin ierisinde yapılan literatr taraması kısaca zetlenmiş kavramsal modelleme sreci gzden geirilerek gncellenmiş hali verilmiş nerilen modelleme yaklaşımı tanıtılmış, oluşturuılan RKM zellikleri aıklanmış ve kontrol mhendisliği rneği detaylı olarak tartışılmıştır.

Mart, 2006

Ali GRBZ

ABSTRACT

INTELLIGENT CONCEPTUAL MODELING TOOL

In this study, a modeling approach which can be used to develop systems conceptually before physical design is developed. In-CM Conceptual Modeling Tool, shortly called In-CM, which can be used to develop interdisciplinary conceptual models is designed and implemented based on the proposed modeling approach. The main property of the In-CM is intelligence that facilitates developing conceptual model of complex systems.

Firstly, conceptual modeling is discussed in the study. Conceptual modeling is one of the major activities in system engineering, knowledge engineering and software engineering, but its use is not limited to these areas. Conceptual modeling is, in general terms, an activity of creating model of systems using terms of concepts facilitating the understanding of social and scientific phenomenon. In the literature search, firstly, current conceptual modeling process and techniques are examined. In the examination, it is shown that conceptual modeling techniques consist of two sub categories, classical and developing. Both of them have following constraints:

- All aspect of problem domain is not modeled with them. These techniques take only one side of the domain and show it, for instance, data flow diagrams shows only data flows in the domain, or ER diagrams show only entities and their attributes in the domain. This problem is solved by proposed modeling approach
- Current techniques are related to “how the problem is solved” rather than “what the problem is”. In this study, a modeling approach which concerns with “what the problem is” and its natural attributes is proposed.
- Current techniques also use object oriented symbology. So, models are drawn based on problem solution. This causes working on the problem without understanding it. In order to solve this constraint, a modeling methodology and its graphical notation are announced.

In this study, proposed and explained modeling approach is designed along with understanding of all aspect of the problem, defining all actors (perform tasks), tasks (main goals), processes (sequencial operations) and infrastructures (hardware, software etc.). Problem or system domain is modeled with some aspect of problem domain by the approach. Proposed modeling approach is not opposite to current conceptual modeling process, but new approach is developed with expanded current process. Thus, proposed modeling approach will provide some benefits to studies in this field. Main advantage of the approach is proposed a Reference Conceptual Model (RCM) that utilizes in any field to develop conceptual models which are different from each other. RCM is a meta-model and covers all of problem domain. Users can develop conceptual models from RCM. However, the scope of these models are more limited than the scope of RCM.

Current conceptual modeling process is studied and updated in accord with this study's aim and an expert system software that is used to develop conceptual model interdisciplinary is implemented. The software has knowledge base, and inference engine like all expert systems. In the knowledge base, Reference Conceptual Model (RCM) is used as a pattern to formalize the raw user requirements. Beside it, there are relations among data in the knowledge base. Deepness and correctness of the RCM is contingent on knowledge level of subject matter expert (SME). In-CM has a knowledge base editor to define RCM and relations. SME can describe RCM and relations by using In-CM Knowledge Base Editor.

Developing a conceptual model according to proposing process can be done by using In-CM Inference Engine. While developing conceptual model, it provides assistance and shows uncompleted branches. At the end of this process, an XML file is produced. This file includes user conceptual model and can be shown graphically on the In-CM Graphic Editor.

In this study, "In-CM Intelligent Conceptual Model Generator" is sampled with a real life problem to show applicable of proposed approach. In order to show contribution of the study, a simple problem is selected. However, proposed approach is independent from simplicity of the problem and applicable to all kinds of problems. In this study, conceptual model of establishing "control engineering education" in any university is developed by using implemented software called Intelligent Conceptual Model Generator.

In the thesis, literature is summarized and current conceptual modeling process is studied and updated. Later, proposed modeling approach, RCM and its properties are explained. Finally, an example about control engineering education is discussed in detail.

March, 2006

Ali GÜRBÜZ

YENİLİK BEYANI

ZEKİ KAVRAMSAL MODEL GELİSTİRME ARACI

Bu tez çalışması kapsamında, kavramsal modelleme yapılabilen ve kullanıcıyı yönlendirebilen “yapay zeka”ya dayalı kavramsal model geliştiren bir yazılım geliştirilmiştir. Geliştirilen sistemin eğitim alanında bir uygulaması gösterilmiştir. Bu çalışma, Kavramsal Modelleme (KM) ile Yapay Zeka (YZ) biliminin bileşkesinden oluşan bir çalışmadır.

Bu tez çalışmasında yapılan yenilikler aşağıda belirtilmiştir:

- Herhangi bir sistemi veya herhangi bir problemi ilgili alan içerisindeki aktörler (eylemleri gerçekleştirenler), süreç (işlemler dizisi), görev (temel amaç), alt yapı (yazılım, donanım vs.) gibi öğelerini içerecek şekilde bir kavramsal modelleme yaklaşımı önerilmiştir. Bu yaklaşım ile problem alanı bir bütün halinde birkaç açıdan kapsanacak şekilde modellenmiş ve ilgili kavramsal modelin oluşturulması mümkün olmuştur.
- Önerilen yaklaşım bilinen kavramsal modelleme sürecine de ters düşmemektedir. Ancak bu yaklaşım süreci de güncelleyecek öneriler ortaya çıkartmıştır. Böylece, güncellenen kavramsal modelleme süreci bu alanda çalışmalara katkı sağlayacaktır. Tanımlanan sürece “In-CM Kavramsal Modelleme Süreci” adı verilmiştir. Bu süreç ile kavramsal modelleme disiplinler arası geliştirilebilir hale getirilmiştir.
- Disiplinler arası kavramsal modellemeyi sağlayacak bir yaklaşım geliştirilmiştir. Buna dayanarak “Genel Kavramsal Model (GKM)” adı verilen bir yapı önerilmiştir. Önerilen yaklaşım sistemlerin genel bir ayrıştırmasını kavramsal model olarak gösteren bir yapıdır. Genel Kavramsal Model içerisinde sistemlerin kavramsal modeli, Amaç, Disiplin, Altyapı, Aktör, Süreç ve Görev olmak üzere altı değişik yönden oluşmaktadır.
- Tanımlanan süreç esas alınarak disiplinler arası kavramsal model gerçekleştirilen, kullanıcıya destek veren yapay zekaya dayalı bir yazılım

gelistirilmistir. Gelistirilen bu yazilim, sistemler arasi kavramsal modelleme gereklestiren ve yapay zekaya dayali ilk sistemdir. Bu yazilim ile farkli alanlarda kavramsal model gelistirilebilir. Bu yazilim üç alt modülden olusmaktadır. In-CM Bilgi Tabani Düzenleyici, Çikarim Mekanizmasi ve In-CM Grafik Editörü bu üç modüldür.

- Önerilen yöntemin en önemli faydasi herhangi bir alanda o alana özgü degisik özelliklerde kavramsal modelleme yapmayi kolaylastiracak bir *Referans Kavramsal Model* (RKM)'in olusturulmasini mümkün kilmasidir. RKM problem alaninin tamamini kapsamakta olup oldukça kapsimli bir meta modeldir. Kullanici kendi istek ve ihtiyaci dogrultusunda referans modelden kullaniciya özel daha dar kapsimli modeller olusturabilir.
- RKM yapisi sistem ayristirilmasinda kullanilabilecek olan bir meta modeldir. Bu meta model ile genel olarak sistemler alti farkli açıdan kavramsal olarak tasarlanabilmektedir. Bir RKM'dan çok sayida GKM olusturulabililmektedir.

Mart, 2006

Prof. Dr. Burhanettin CAN

Ali GÜRBÜZ

KISALTMALAR LISTESİ

DFD	:	Data Flow Diagram
DGD	:	Durum Geçis Diyagramları
EM	:	Enformasyon Modelleme
ERM	:	Entity Relationship Model
GKM		Genel Kavramsal Model
In-CM	:	Intelligent Conceptual Modeling Tool
KKM	:	Kullanıcı Kavramsal Model
KM	:	Kavramsal Modelleme
NYA	:	NesneYönelimli Analiz
OOA	:	Object Oriented Analysis
OOAD	:	Object Oriented Analysis and Design
OOD	:	Object Oriented Design
RKM	:	Referans Kavramsal Model
SM	:	Sistem Mühendisi
SADT	:	Structured Analysis and Design Technique
SN	:	Semantic Networks
TDK	:	Türk Dil Kurumu
UML	:	Unified Modeling Language
YZ	:	Yapay Zeka

SEKILLER LISTESİ

Sekil II- 1 Sistem yaşam döngüsüne Blum yaklasimi (Dieste ve dgr. 2000).	10
Sekil II- 2 Genel “Sistem Yaşam Döngüsü”.	13
Sekil II- 3. Veri-Enformasyon-Bilgi Iliskisi (Pressman,1992).	13
Sekil II- 4. DFD notasyonu (Pressman,1992).	17
Sekil II- 5. Konferans Probleminin DFD ile olusturulmus kavramsal modeli (Dieste ve dgr. 2000).	18
Sekil II- 6. ER notasyonu gösterimi (Dieste ve dgr., 2000).	20
Sekil II- 7. Konferans Probleminin ER ile olusturulmus kavramsal modeli (Dieste ve dgr. 2000).	21
Sekil II- 8. EMM notasyonu (Hayy, 1999).	22
Sekil II- 9. Konferans Probleminin EMM ile olusturulmus kavramsal modeli.	23
Sekil II- 10. Semantik Ag ile bir kavramsal model (Mylopoulos, 2001).	24
Sekil II- 11. SADT modeli genel notasyonu.	25
Sekil II- 12. Konferans Probleminin SADT ile olusturulmus kavramsal modeli. ...	26
Sekil II- 13. Coad modeli	26
Sekil II- 14. UML notasyonu (Fowler ve Scoot 2001).	27
Sekil II- 15. Konferans Probleminin UML notasyonu ile olusturulmus kavramsal modeli (Dieste ve dgr. 2000).	28
Sekil II- 16. DFD Diyagramlari notasyonu.	28
Sekil II- 17. Konferans Probleminin durum geçis diyagramlari ile olusturulmus kavramsal modeli (Dieste ve dgr. 2000).	29
Sekil II- 18. KAOS meta-modeli örneği (Dieste ve dgr. 2000).	31
Sekil II- 19. Konferans probleminin KAOS ile gösterimine örnek (Dieste ve dgr. 2000).	31
Sekil II- 20. Konferans probleminin CASE* ile gösterimine örnek.	32
Sekil II- 21. IDEF1X Model örneği (Hay, 1999).	34
Sekil II- 22. i* Model örneği (Dieste ve dgr. 2000).	35
Sekil II- 23. TELOS Model örneği (Dieste ve dgr., 2000).	36
Sekil II- 24. Uzman sistem elemanlari.	42
Sekil III- 1. Bilinen Kavramsal Modelleme Süreci.	49
Sekil III- 2. “Problem” uzayina karsilik “Makine” uzayi (Dieste ve dgr., 2000). ..	50

Sekil III- 3. In-CM Problem çözüm uzayı.	51
Sekil III- 4. Yazılım Gelistirim Sürecinde Modelleme (Dieste ve dgr. 2000).	52
Sekil III- 5. In-CM Sistem Gelistirim Sürecinde Modelleme.	52
Sekil III- 6. In-CM Kavramsal Model Gelistirim Süreci.	53
Sekil IV-1. In-CM Referans Kavramsal Modelin genel yapısı	57
Sekil IV-2. In-CM Zeki KM Gelistirme Yazilimi Kapsamı.	58
Sekil V-3. In-CM Genel Program Yapısı.	61
Sekil V- 1. In-CM Referans KM’de Amaç detaylandırılması örneği.	62
Sekil V- 2. In-CM Referans KM’de Disiplin detaylandırılması örneği.	63
Sekil V- 3. In-CM Referans KM’de Altyapı detaylandırılması örneği.	63
Sekil V- 4. In-CM Referans KM’de Aktör detaylandırılması örneği.	64
Sekil V- 5. In-CM Referans KM’de Süreç detaylandırılması örneği.	64
Sekil V- 6. In-CM Referans KM’de Görev detaylandırılması örneği.	65
Sekil V- 7. In-CM Bilgi Tabanı Editörü genel görünümü.	67
Sekil V- 8. In-CM Çıkarım Mekanizması genel görünümü.	68
Sekil V- 9. In-CM Grafik Modülünün genel görünümü.	70
Sekil VI- 1. In-CM’de Yeni Kullanıcı KM olusturulması.	75
Sekil VI- 2. In-CM’de Kullanıcı KM olusturulmasında Kullanıcı bilgi ekranı.	76
Sekil VI- 3. In-CM’de KKM’e RKM’den istegin eklenmesi.	77
Sekil VI- 4. In-CM’de Kullanıcı KM Tamamlama Formu.	78
Sekil VI- 5. In-CM’de KKM Tamamlandıktan sonraki görünüm.	78
Sekil VI- 6. In-CM’de Grafik Editöründe KKM Amaç bölümü.	79
Sekil VI- 7. In-CM’de Grafik Editöründe KKM Disiplin bölümü.	97
Sekil VI- 8. In-CM’de Grafik Editöründe KKM Altyapı bölümü.	98
Sekil VI- 9. In-CM’de Grafik Editöründe KKM Aktör bölümü.	98
Sekil VI- 10. In-CM’de Grafik Editöründe KKM Süreç bölümü.	99
Sekil VI- 11. In-CM’de Grafik Editöründe KKM Görev bölümü.	99
Sekil A- 1. In-CM Ana Programı genel görünümü.	113
Sekil A- 2. In-CM’de BTE dil seçimi görünümü.	113
Sekil A- 3. In-CM BTE ana ekran görünümü.	114
Sekil A- 4. In-CM BTE’de RKM’e Yeni Alt Eleman Ekleme.	116
Sekil A- 5. RKM’e Yeni Alt Eleman Ekleme Diyalog kutusu görünümü.	116
Sekil A- 6. RKM’e Yeni Alt Eleman Ekleme örneği ekran görünümü.	118
Sekil A- 7. RKM’de Eleman Güncelleme diyalog kutusu görünümü.	118

Sekil A- 8. Niteliksel Kural Editörü ekran görünümü.	120
Sekil A- 9. Niteliksel Kural Editörü ile örnek bir kural oluşturma.	121
Sekil A- 10. Nicel Kural Editörü ekran görünümü.	122
Sekil A- 11. Nicel Kural Editörü ile örnek bir kural oluşturma.	123
Sekil A- 12. In-CM’de Çıkarım Mekanizması dil seçimi görünümü.	124
Sekil A- 13. In-CM Çıkarım Mekanizması ana ekran görünümü.	125
Sekil A- 14. Yeni Kullanıcı KM oluşturulması.	126
Sekil A- 15. Kullanıcı KM’ne eleman ekleme.	127
Sekil A- 16. Kullanıcı KM’nin tamliğinin bozulması.	127
Sekil A- 17. Adım Adım Tamamlamada Kullanıcı KM’si.	128
Sekil A- 18. Kullanıcı Kavramsal Model Tamamlama Formu.	129
Sekil A- 19. In-CM’de Grafik Editörü ana ekran görünümü.	130

TABLÖLAR LİSTESİ

Tablo V. 1 In-CM Kavramsal Model notasyonu.	70
Tablo VI. 1 XML Formatındaki Kullanıcı Kavramsal Modeli.	80
Tablo VI. 2 In-CM Zeki Kavramsal Model Geliştirme Yazılım Aracı'nın incelenen çalışmalar ile karşılaştırılması.	103

BÖLÜM I

GİRİS VE AMAÇ

I.1 GİRİS

Kavramsal modelleme, özellikle yazılım ve sistem geliştirme süreci içerisinde, son zamanlarda önemini arttıran bir etkinliktir. Sistem geliştirim yaşam döngüsü içerisinde, bir sistemin geliştirilmesi müşteri (kullanıcı) isteklerinin belirlenmesi ile başlar. Bu istekler üzerine bir tasarım ve bu tasarım temel alınarak sistemin gerçekleştirilmesi sağlanır. Daha sonra sistem gerekli doğrulama ve test işlemlerinden geçirilerek müşteriye teslim edilir. Ancak müşteri ile sistem geliştiricileri arasında:

- etkisiz ve yetersiz iletişim,
- yanlış anlaşılma,
- eksik uzmanlık bilgisi,
- yetersiz dokümantasyon,
- zaman kısıtı,

nedenlerinin hepsinden ya da bazılarından dolayı isterler eksik yada yanlış tespit edilebilir. Bu da sonuçta ortaya çıkacak olan ürünün, müşterinin istegini tam karşılayamamasına sebebiyet verebilir. Bu nedenle, sistem geliştirmede müşteri isteklerinin tüm yönleri ile ve eksiksiz tespit edilmesi büyük bir önem arz etmektedir. Aynı zamanda, tespit edilen istekler, müşteriye ve sistem geliştiricisine aynı şeyleri ifade etmelidir. Bu isteklerin eksiksiz ve tüm yönleri ile tespit edileceği, müşteri ile sistem geliştiricisinin bu isteklerden aynı şeyleri anladığını gösteren yapılara ihtiyaç vardır. Kavramsal modelleme bu amaç için kullanılan bir yaklaşım olarak görülebilir. Örneğin, bir konferans düzenlenmesi, içerisinde bir çok kişi, grup, alt yapı gereksinimi, görevleri ve süreçleri olan karmaşık bir etkinliktir. Bu işlemin kâğıt üzerinde el ile gerçekleştirilmesi bilgi, emek ve zaman yoğunluktadır. Ön bir analiz ve model oluşturulmadan gerçekleştirilen planlamalarda bir çok aksaklık veya gecikme görülebilir. O nedenle öncelikle böyle bir etkinliğin planlanması istendiğinde, bu etkinliğin amacı, kapsamı, tarihleri gibi müşteri istekleri belirlendikten sonra, bu isteklerden kavramsal düzeyde bir model oluşturulabilir.

Böylece bu etkinlik içerisinde yer alan herkes ile ortak anlaşırlılığın sağlanabileceği bir yapı ortaya konmuş olur. Bu konuda diğer bir örnekte, bir fabrika otomasyonuna yardımcı olacak olan bir uzman sistem geliştirilmesi istenmesidir. Bu etkinlikte de yine müşteri istekleri belirlendikten sonra oluşturulacak olan bir kavramsal model, sistem elemanlarını, verilerini, durumlarını ve ilişkilerini gösteren bir kavramsal model ile sistemin analizi yapılabilir ve müşteri ile müşterinin istekleri üzerinde ortak bir anlaşılabilirliğe ulaşılabilir.

Kavramsal modelleme sistem yaşam süreci içerisinde aşağıdaki amaçlar doğrultusunda kullanılır :

- Bir sistemin analiz edilirken anlaşılabilirliğini arttırmak,
- Müşteri ile sistem geliştirici arasında güvenilir bir iletişim sağlamak,
- Sistem için alan (domain) odaklı bir dokümantasyon sağlamak.

Bir sistemin uygulaması düşünülmesi dahi, kavramsal modelinin oluşturulması sistemin yapısal ve işlevsel olarak anlaşılmasını kolaylaştırır. Bu nedenle, kavramsal modellemenin sosyal bilimlerden, fen bilimlerine, hatta tıp bilimine kadar değişik disiplinler içerisinde kullanımlarına literatürde rastlamak mümkündür. Her disiplin, kavramsal modellemeyi kendi açısından ele alarak, kendilerine ait gösterim teknikleri geliştirmiştir.

Bu açıklamalardan sonra kavramsal modellemenin bir tanımı yapılırsa: “Kavramsal Modelleme (KM), genel olarak, fiziksel veya sosyal sistemlerin anlaşılabilirliğini kolaylaştırmak için, sistemin, alt parçalara bölümlendirilerek (biçimsel olarak -teknik terimleri kullanmadan-) ifade edilmesi etkinliğidir” (Protte ve dgr. 1994). Kavramsal modellemenin temel dayanağı, uygulama alanında yer alan sistem elemanlarının, aktörlerinin, verilerinin, durumlarının ve ilişkilerinin kavramsal bazda belirlenip, grafiksel notasyonlar ile ifade edilmesidir. Kavramsal modelleme, sistem geliştirme yaşam döngüsü içerisinde tasarım ve uygulama fazlarından önce gelir.

I.2 TEZİN GEREKÇESİ

Teknolojideki gelişmeler, günümüz problemlerini ve bu problemlere ait çözümleri oldukça karmaşık hale getirmiştir. Problemler içerisindeki aktör sayıları artmış, sistemler birden fazla alt sistemden, her alt sistem de kendine ait aktör, durum, ve olaydan oluşur hale gelmiştir. Tüm bunlar arası ilişkiler ise çok daha fazla

kurala ve kisitta baglanmis, ayni zamanda da karmasiklastirilmistir. Problemleri tüm bu yönleri ile görmek ve problemlerin çözümünü tam olarak ortaya koymak ise geniş uzmanlık bilgisi istemektedir. Bunlardan dolayidir ki, bu tür problemlerin çözümünde ya da sistemlerin geliştirilmesinde uzmana (insana) bagli hatalarin ortaya çıkma ihtimali artmistir. Ortaya çıkacak sorunlari çözebilecek yardımcı yapılara ihtiyaç vardır. İşte kavramsal modelleme bu hatalari önceden tespit etmek için kullanılan bir yapıdır. Ancak;

- Bilinen kavramsal modelleme teknikleri ile problem alanı objektif bir şekilde tüm yönleri ile gösterilememektedir. Bu teknikler, üzerinde çalışılan problemin sadece bir yönünü ele almakta ve o problemin o yönü ile ilgili kavramsal modeli ortaya koymaktadır. Problem alanının tamamını tek bir kavramsal model altında farklı açılardan değerlendirecek, incelemeyi mümkün kılacak ve sistemin özellikleri ile elemanlarının birbirleri arasındaki ilişkilerini ortaya koyacak bir yaklaşıma ihtiyaç vardır.
- Mevcut kavramsal modelleme teknikleri “*problemin ne olduğundan*” çok “*problemin nasıl çözümleneceği*” ile ilgilenmektedir. Problemin doğal niteliklerini dikkate alan ve “*ne olduğunu*” irdeleyebilen bir modelleme yaklaşımına ihtiyaç vardır.
- Bilinen tekniklerin kullandıkları grafiksel gösterimler nesne yönelimli analiz sembololojileridir. Bundan dolayı, çizilen modeller, problemin çözümüne odaklı bir şekilde oluşturulmaktadır. Bu da problemin tam anlaşılmadan çözümü üzerine çalışılması demektir. Problemin tam olarak anlaşılmasını problemin alt elemanlarının ve onların ilişkilerinin neler olması gerektiğini ortaya koyan ve gösterebilen bir modelleme yaklaşımı ve ilgili gösterim metodolojisine ihtiyaç vardır.
- Bilinen kavramsal modelleme sürecini problemin çözüm yöntemi ve gösterimine odaklanmış bir yapıdır. Problemin detay özelliklerini ortaya koyabilecek şekilde sürecin genişletilmesine ihtiyaç vardır.
- Yapılan literatür taraması herhangi bir konuda hazırlanan bir referans modelin bir benzerinin yapılması durumunda çalışmaların yeniden yapılmasını gerektiğini göstermektedir. Halbuki herhangi bir konuda tüm problem alanını kapsayabilecek bir referans model geliştirilebilir

ise aynı konuda farklı kavramsal modeller oluşturulması daha kolay olabilir. Bu kapsamda geliştirilecek olan bir referans model çok büyük yararlar sağlayacaktır.

Kavramsal modelleme konusunda yazılım sistemlerinden de faydalanmak mümkündür. Özellikle yapay zeka teknolojisinin kavramsal model geliştirme konusunda önemli katkıları olacaktır. Yapay Zeka (YZ) günümüz problemlerine sunmuş olduğu çözümler ile iş hayatımızdan günlük yaşamımıza içine bir çok alanda karşımıza çıkmaktadır. Rich ve Knight (1991), yapay zekayı, insanların günümüzde daha iyi yaptıkları işleri bilgisayarlara nasıl yaptırılabilir sorusunun cevabını veren çalışmalar olarak tanımlamaktadırlar. Yapay zekada, insanların bir olay karşısında karar vermeye ve olaya çözüm getirmeye çalışırken nasıl bir düşünce yolu izledikleri taklit edilmektedir. Bu sistemlerde düşünce işlemi basit adımlara bölünmekte ve bu adımları kullanarak problemi çözen bir bilgisayar programı tasarlanmaktadır. Yapay zeka çalışmaları sonucunda ortaya çıkan ürünlerin, özellikle gelişmiş ülkelerde, günlük hayatta kullanımları, insanlara bir çok fayda sağlamakta, işlemlerin maliyetlerini azaltmakta, kârliliklerde artışlar olmakta, işlem zamanları kısaltılabilmekte ve problemlere daha iyi çözümler üretilebilmektedir. Bu noktadan hareket ile, yapay zeka da, kavramsal modelleme de, birbirine benzer faydaları sunmaktadırlar. Kavramsal model kullanmak ile sistem geliştirmelerde hata oranı düşmekte, sistemler daha kısa sürede hızlı bir şekilde tasarlanıp, gerçekleştirilmeye geçilebilmekte, geliştirilen sistemlerin doğrulanması tam olarak yapılabilmekte, müşteri sonuçta istediği ürüne mümkün olduğunca tam olarak kavuşabilmektedir.

Kavramsal model geliştirmeye, yapay zekanın destek vermesi ile sistem geliştirmede, hata oranı daha da azaltılarak, daha güvenilir ve doğru sistemlerin geliştirilmesi sağlanabilir. Böyle bir sistemin tasarlanabilmesi için genel bir çerçeveye ve örnek uygulamalara ihtiyaç vardır.

Yukardaki gerekçe ve ihtiyaçlardan dolayı bu tez çalışmasında, hem kavramsal modelleme alanında önemli sorunlara çözümler üretilmiş hem de yapay zeka tekniklerinden uzman sistemler kullanarak “Zeki Kavramsal Model Geliştirme Aracı” yazılımı geliştirilmiştir.

I.3 TEZİN AMACI

Kavramsal Modelleme alıřmaları incelendiğinde, KM'nin sosyal bilimlerden fen bilimlerine deęisik disiplinlerde kullanilmasına ragmen, bu alıřmaların yazılım, bilgi ve simölasyon sistemlerinde yoğunlařtıęı görölmektedir. KM'nin kullanildięi disiplinlerde, kavramsal modelleme tekniklerinin ve geliştirme süreçleri farklılık göstermektedir. Bu da bir alanda uzman olan ve o alanla ilgili kavramsal model geliştirebilen bir kişinin dięer bir alan içerisinde bu uzmanlığını ve bildięi KM tekniklerini kullanamaması anlamına gelmektedir. Bu sorun disiplinler arası bir kavramsal modelleme teknięine ile özöllebilir.

Yapay zeka alanındaki uygulamalar incelendiğinde YZ tekniklerinin kavramsal model geliřtirmede de kullanilabileceęi ve önemli faydalar sağlayabileceęi görölmektedir.

Bu iki teknolojiyi birlikte kullanarak bu tez kapsamında sunlar amaçlanmıřtı:

- Bilinen kavramsal modelleme sürecini güncellemek,
- Problemin özüm yönteminden ok ierięi ile ilgilenecek bir kavramsal modelleme yaklařımı geliřtirmek,
- Hızlı, doęru ve güvenilir kavramsal modeller oluřturmayı sağlayacak bir referans model ortaya koymak,
- Kavramsal modeli farklı aılardan gösterecek grafiksel gösterim yöntemi oluřturmak,
- Yapay zeka teknikleri ile kavramsal modelleme sürecini mümkün olduęu kadar otomatik olarak geliřtirecek bir sistem geliřtirmek,

I.4 KAPSAM

Kavramsal modelleme, genel sistem geliştirme ve özellikle de yazılım aęirlikli projelerde yoğunluklu olarak kullanılmaktadır. Sistemin yada yazılım projesinin yapısına göre KM geliştirme teknikleri farklılık göstermektedir.

Bu alıřma kapsamında, KM eylemleri ve yapay zeka teknolojilerinin sistem içerisinde nerelerde, nasıl kullanilabileceęi tez alıřmasında detaylı olarak incelenmiştir.

Bu alıřma, yukarıda belirtilen amaç doęrultusunda tanımlanan kavramsal modelleme süreci ve bu süreç temel alınarak geliştirilen yazılım ile sınırlıdır. Bu yazılımın uygulanabilirlięi eğitim alanında örneklenmiştir.

Gerçekleştirilen çalışmanın sonuçlarını içeren bu tez beş bölümden oluşmaktadır. İlk bölümde, giriş, amaç ve kapsam anlatılmıştır. İkinci bölümde, gerçekleştirilen literatür çalışması sonucunda ulaşılan bilgiler irdelenerek KM'nin ne olduğu, hangi teknikler ile gerçekleştirildiği; yapay zekanın ne olduğu ve mevcut teknikleri; KM konusunda önceki çalışmalar ile mevcut KM tekniklerinin kısıtları ve bu çalışma kapsamında geliştirilen yazılım için tanımlanan kavramsal model geliştirme süreci bulunmaktadır. Üçüncü bölümde, tanımlanan süreç temel alınarak geliştirilen yazılım tanıtılmıştır. Dördüncü bölümde ulaşılan sonuçlara yer verilmiştir. Beşinci bölüm ise tartışma ve değerlendirmeleri kapsamaktadır.

BÖLÜM II

GENEL BİLGİLER

Bu bölüm içerisinde gerçekleştirilen literatür taraması sonucunda ulaşılan bilgiler irdelenerek KM'nin ne olduğu, hangi teknikler ile gerçekleştirildiği; benzeri şekilde yapay zekanın ne olduğu ve mevcut teknikleri de değerlendirilmiştir. Aynı zamanda bu bölüm içerisinde KM konusunda önceki çalışmalar ile mevcut KM tekniklerinin kısıtları ve bu çalışma kapsamında geliştirilen yazılım için tanımlanan kavramsal model geliştirme süreci de anlatılmaktadır.

II.1 Kavramsal Modelleme

II.1.1 Model ve Modelleme

Model, Dieste ve arkadaşlarının 2000'de yapmış olduğu çalışmada aşağıdaki şekillerde tanımlanmıştır (Dieste ve dgr. 2000) :

1. Bir şeyin küçük yada gerçek kopyası, örneğin bir gemi modeli.
2. Yapılmış olan bir şeyin figürü yada kalibi, örneğin dökümcülükte kullanılan kil modelleri.
3. Direkt olarak gösterilemeyecek yada gözlemlenemeyecek olan bir şeyin örneklenmesine ve tanımlanmasına yardımcı olacak olan şey, örneğin atomun modeli.
4. Belirli ilişkilerin durumlarını veya nesnelerini, matematiksel olarak tarif etmek için kullanılan çıkarımlar, veriler ve önerimler yapısı
5. Bilinen yada hayal edilen bir sistemin teorik tahmini.

Dieste ve arkadaşlarının 2000'de yapmış oldukları model tanımlarından sonuncusu kavramsal modelleme kapsamındaki model tanımını göstermektedir. Literatür içerisinde yer alan diğer model tanımlarından bazıları şunlardır:

1. *“Bir sistemin, bir aygıtın veya bir nesnenin özelliklerinin tümünün yada belirli bir bölümünün gösterimidir. (U.S.DoD,1998)”*
2. *“Gerçeğin bazı yönlerinin temsili gösterimidir. (URL.1, 2002)”*
3. *“Model, ilgilenilen alan ile ilgili bir olgunun, olayın yada şeyin tümünün yada belirli bir bölümünün, belirlenmiş gösterim notasyonları ile, belirli amaçlar için basitleştirilmesidir.” (Arrington, 2001)*

Bu tanımlamaların ortak noktalarından hareket ile bu tez kapsamında kullanılacak olan model tanımı su şekilde yapılmıştır:

Model, üzerinde çalışılacak olan bir sistemin, bir olayın yada bir aygıtın tümünün yada ilgilenilen bölümünün, istenilen amaç doğrultusunda, belirli grafiksel şekiller ile temsili gösterimidir.

Modelleme ise “bir sistem, aygıt yada olayın modelinin olusturulması etkinliğine” denir.

II.1.2 Kavramsal Modellemenin Amacı

Kavramsal modelleme, özellikle yazılım ve sistem geliştirme süreci içerisinde, son zamanlarda önemini arttıran bir etkinliktir. Sistem geliştirme yaşam döngüsü içerisinde, bir sistemin geliştirilmesi müşteri (kullanıcı) isteklerinin belirlenmesi ile başlar. Bu istekler üzerine bir tasarım ve bu tasarım temel alınarak sistemin gerçekleştirilmesi sağlanır. Daha sonra sistem gerekli doğrulama ve test işlemlerinden geçirilerek müşteriye teslim edilir. Ancak müşteri ile sistem geliştiricileri arasında:

- etkisiz ve yetersiz iletişim,
- yanlış anlaşılma,
- eksik uzmanlık bilgisi,
- yetersiz dokümantasyon,
- zaman kısıtı,

nedenlerinin hepsinden ya da bazılarından dolayı istekler eksik yada yanlış tespit edilebilir. Bu da sonuçta ortaya çıkacak olan ürünün, müşterinin istegini tam karşılayamamasına sebebiyet verebilir. Bu nedenle kavramsal modelleme aşağıdaki amaçlar doğrultusunda kullanılır:

- Sistem geliştirmede müşteri isteklerinin tüm yönleri ile ve eksiksiz tespit etmek,
- Tespit edilen isteklerin, müşteriye ve sistem geliştiricisine aynı şeyleri ifade etmesini sağlamak,
- Sistem için alan (domain) odaklı bir dokümantasyon sağlamak.
- Sistemin veya problemin elemanlarını, verilerini, durumlarını ve ilişkilerini göstermek

- Bir sistemin analiz edilirken anlasilabilirliğini arttirmek,
- Müsteri ile sistem geliştirici arasına güvenilir bir iletişim sağlamak,

Bir sistemin uygulaması düşünülmesi dahi, kavramsal modelinin oluşturulması sistemin yapısal ve işlevsel olarak anlaşılmasını kolaylaştırır. Bu nedenle, kavramsal modellemenin sosyal bilimlerden, fen bilimlerine, hatta tıp bilimine kadar değişik disiplinler içerisinde kullanımlarına literatürde rastlamak mümkündür. Her disiplin, kavramsal modellemeyi kendi açısından yukarıdaki amaçlar doğrultusunda ele alarak, kendilerine ait gösterim teknikleri geliştirmiştir.

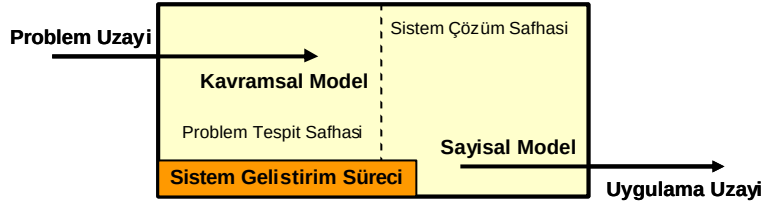
II.1.3 Sistem Yaşam Döngüsü İçerisinde Modeller

Bu tez çalışması kapsamında kavramsal modellemenin disiplinler üstü bir yaklaşımla incelenmesi amaçlandığı için, modellerin sistem yaşam döngüsü içerisinde nasıl kullanıldığına bakılmıştır. Sistem yaşam döngüsü içerisinde modeller, iki temel amaç için kullanılır (Dieste ve dgr. 2000) :

1. Kullanıcı tarafından ifade edilen ve geliştirilmesi istenilen sistemin, çözmesi beklenen sorunun tanımlanması ve anlaşılmasıdır. Bu amaçla oluşturulan modeller genel olarak **“Tanımlayıcı veya Kavramsal Model”** olarak adlandırılırlar.

2. Sisteminin nasıl gerçekleşeceği ve bu gerçekleştirme içerisinde hangi eleman ve ilişkilerin bulunacağı, çözüm için hangi teknik ve yaklaşımların kullanılacağına gösterilmesidir. Bu amaçla oluşturulan modellere de **“Sayısal Model”** olarak adlandırılırlar.

Dieste ve arkadaşları 2000'deki çalışmada, Blum tarafından 1996 yılında yayınlanan “Programlamanın Ötesinde : Tasarımın Yeni Çağına Doğru” isimli kitabında önerdiği, sistem yaşam döngüsü içerisinde modellerin kullanımı yaklaşımını incelemişlerdir. Sonuç olarak bu yaklaşımın, sistem geliştirmede, modellerin nerede kullanıldıklarını net olarak belirlediğini ifade etmişlerdir. Sözü edilen çalışmadan alınan Blum yaklaşımı Şekil II-1’de görülmektedir. Şekildeki “büyük dikdörtgen” sistem geliştirme sürecini ifade etmektedir.



Sekil II- 1 Sistem yaşam döngüsüne Blum yaklaşımı (Dieste ve dgr. 2000).

Blum yaklaşımında iki temel safha vardır.

1. Birinci safha, “**problem tespit safhası**”, ki bu safhada problemin tespit edilip anlasılması amaçtır. Bu nedenle problem alanına ait kavramların tespiti yapılır. Tespit edilen kavramlar, problemin anlasılmasına ve sistem tasarımının yapılmasına yardımcı olacak olan kavramsal modelin geliştirilmesinde kullanılır.

2. İkinci safha “**sistem çözüm safhası**”dır. Bu safhada anlaşılmış olan problemin nasıl çözüme ulaştırılacağı amaçtır. Yani mevcut sorunun ne şekilde hangi teknikleri kullanarak çözüleceği, sistemin işlevselliğinin nasıl olacağına dair modeller geliştirilir. Sayısal model ismi verilen bu modellerin geliştirilmesinde odak noktası *problemin çözüm alanı*dir. Çözüm alanına göre sayısal model teknikleri de farklılık gösterir.

Bu tez çalışması, Blum yaklaşımın birinci safhasında ifade edilen kavramsal modelleri kapsamaktadır.

II.1.4 Kavramsal Modelleme Nedir?

Kavramsal modelleme, değişik disiplinler içerisinde yaygın olarak kullanılmasına ve her disiplin kavramlara kendine göre bir anlam yüklemesine rağmen, KM orijin olarak enformasyon sistemlerindeki veritabanları ile ilgilidir ve veriler ile bu veriler arası ilişkileri göstermek için kullanılır (Dieste ve dgr. 2000, Hay 1999, Lommens 2001, Manley 2000, Meersman ve Weigand. 1997, Pace 1999).

Genel bir tanım yapmadan önce **kavram** ve **kavramsal** terimlerinin sözlük anlamlarını verilmesi faydalı olacaktır. Türk Dil Kurumu’nun elektronik sözlüğünde kavram “*Nesnelerin veya olayların ortak özelliklerini kapsayan ve bir ortak ad altında toplayan genel tasarım, mefhum, nosyon*” şeklinde; kavramsal ise

“Kavramla ilgili, kavram niteliğinde olan” şeklinde tanımlanmaktadır (TDK, 2006). Dolayısı ile, kavramsal modelleme içerisinde ortak özelliklere sahip olan “şeyler” kavram adı altında bir grup haline getirilmekte ve bu gruplar arası ilişkiler tanımlanmaktadır.

KM ile ilgili literatürde değişik disiplinlere ait farklı tanımlamalara rastlanmaktadır (Burg ve Riet 1996, Protte ve dgr. 1994, Pace 2000, Meersman ve Weigand 1997). Bu tanımlamalar dikkate alınarak kavramsal modelleme ile ilgili genel bir tanımlama şu şekilde yapılabilir:

“Kavramsal modelleme, fiziksel veya sosyal sistemlerin anlaşılabilirliğini kolaylaştırmak için alt parçalara bölümlendirilerek sistemin (biçimsel olarak -teknik terimleri kullanmadan-) gösterilmesi eylemidir” (Protte ve dgr. 1994).

Diğer bir ifade ile ‘KM, tasarımcılar ile, uygulayıcı ve sistem geliştiriciler arasında iletişimi ve ortak anlaşılabilirliği sağlayan bir yapıdır’. KM’nin temel dayanağı, uygulama alanının kavramsal bazda modellerinin oluşturularak, aktörlerin, ilişkilerin, amaçların, sınırlamaların ve kuralların belirlenmesidir.

KM’nin enformasyon sistemleri açısından tanımı ise şu şekilde verilmiştir:

“Bilgi sistemlerinin geliştirilmesinde, insâ edilmek istenen sistemde kullanıcı açısından önemli olan ve vurgulanması gereken noktaların tespit edilmesi için isteklerin soyut biçimde gösterilmesi işlemidir” (Burg ve Riet 1996).

Kavramsal modelleme ile ilgili Dehne ve dgr. 2000’de yapılan tanım ise şu şekildedir :

“Bir sistemin davranışını ya da ayrıştırılmasını gösteren yapılardır”. (Dehne ve dgr. 2000).

Bir kavramsal modele bakıldığında, model üzerinde çeşitli grafiksel gösterimler ile ifade edilmiş olan elemanlar (entity), eylemler (activity), görevler (task), süreç (process) ve aktörler (actor) görülecektir. Bunlar kavramsal model insâ edilirken kullanılan genel bloklardır. Bu blokların grafiksel şekilleri kavramsal modelin gerçekleştirildiği disipline göre farklılık gösterir. Ancak disiplin değişse dahi, kavramsal modelin insâ edilebilmesi için, bu blokların tespiti müşteri (kullanıcı) isteklerinden elde edilir. Müşteri (kullanıcı) isteklerinden oluşturulan **kavramsal modeller aşağıdaki amaçlar doğrultusunda kullanılır :**

- Bir sistemin analiz edilirken ya da tasarlanırken anlaşılabilirliğini arttırmak.(URL.2 2006, Pace, 1999)

- Bilgi modeli geliştirilmesinde kullanıcıya ve modeli geliştiriciye kolaylık sağlamak (Pace, 1999, Dieste ve dgr., 2000).
- Model geliştirici ile sistem geliştirici arasına güvenilir bir iletişim sağlamak (Dieste ve dgr., 2000, Pace, 1999).
- Bir sistem için tasarım ve uygulama fazlarını belirlemek (Dieste ve dgr. 2000).
- Sistem için alan(domain) odaklı bir dokümantasyon sağlamak (Pace, 1999).
- Sistem geliştirildikten sonra, sistem üstündeki düzenlemeler için bir ön çalışma alanı (ayrıntılı tasarım – blueprint-) oluşturmak (Pace, 1999, Dieste ve dgr., 2000).

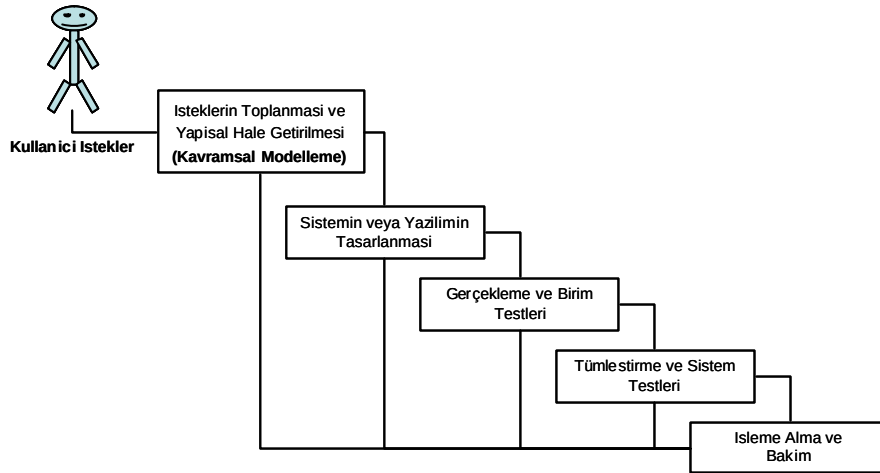
Görüldüğü gibi, değişik disiplinlerde grafiksel olarak farklı gösterimlere sahip olan kavramsal modellerin, farklı alanlar içerisindeki ortak kullanım amacı, iki kelime ile genellenebilir **“problemin analizi”** ve **“ortak anlaşılabilirlik”** sağlanmasıdır. Bu nedenle bu tez çalışması kapsamında yapılan tüm çalışmalarda bu iki amaç odak noktası olarak korunmuştur.

II.1.5 KM Nasıl Gerçekleştirilir?

Sistem mühendisliğinde, bir sistemin geliştirilmesi, kullanıcının ihtiyaçlarını ve isteklerini, doğal dilde, sistem geliştiricilerine aktarması ile başlar. Bu adım, sistem mühendisliğinde kullanıcı isteklerinin tespiti ya da toplanması şeklinde ifade edilir. Bu adımdan sonra toplanan isteklerin tasarım aşamasına geçmeden önce yapısal hale getirilmesi gereklidir. Bunun yapılmasının sebebi, doğal dil anlatımı farklı kişilerce farklı anlaşımlara sebep olabilmektedir. Aynı zamanda sisteme ait unutulmuş noktalarında tespit edilmesi açısından da yapısallık önemlidir. Kavramsal Modelleme ile kullanıcı istekleri yapısal hale getirilir. Böylece, istekler üzerinde sistem geliştirici ve kullanıcı daha rahat iletişim kurarak doğru noktalar üzerinde çalışabilir, unutulmuş ya da gözden kaçmış noktaları tespit edebilir ve geliştirilmesi düşünülen sistemin geneli üzerinde rahatlıkla anlaşılabilirler.

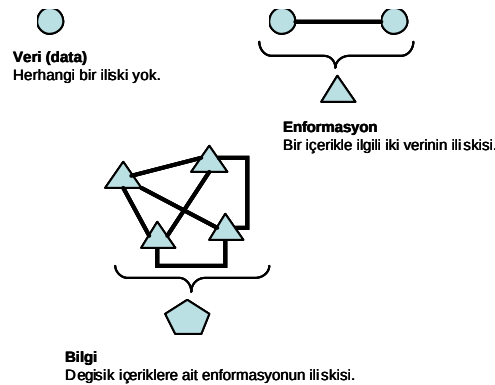
Sistem geliştirme yaşam döngüsünde, kullanıcı isteklerinin yapısal hale getirilmesinden sonraki adım, tasarımıdır. Bu adımda sistem modülleri, gerekli alt modüller ve bu modüller arası arayüzler belirlenir. Daha sonra tasarımı yapılan

sistem gereklenir. Gerekli test ve dođrulama iřlemlerinden sonra sistem kullanıcıya teslim edilir. Sistem yaşam dngs Sekil II-2’de grlmektedir.



Sekil II- 2 Genel “Sistem Yaşam Dngs”.

Sistem Yaşam dngs ierisinde grldđ gibi KM Kullanıcı iřterlerinin toplanması ve yapısal hale getirilmesi adımı ierisinde yer almaktadır. Kullanıcı iřteklerinin yapısal hale getirir iken kullanılan kavramsal modelin girdileri kullanıcının kafasındaki, veri, enformasyon ve bilgiden oluřmus olan bir yigindir. Veri, bir sisteme ait en kk ve islenmemiř “yapı”dır. Bu veriler, islenerek sistem hakkındaki enformasyon oluřturulur. Diđer bir ifade ile, veriler arası iliřki ve etkileřimlerin kurularak oluřturulan yapıya “enformasyon” denir. Elde edilen enformasyonlar arası etkileřim ve iliřkilerden de “bilgiye (knowledge)” ulařılır. Tm bu ulařımlarda elemanlar, alt elemanlar, iliřkiler ve etkileřimler mevcuttur. Sekil II-3’de bu yapı gsterilmiřtir (Pressman,1992).



Sekil II- 3.Veri-Enformasyon-Bilgi İliřkisi (Pressman,1992).

Kavramsal model ile kullanıcının istediği sistemin genel bir iskeleti çıkartılır. Bu çerçeve daha sonra istenilen alanın uzmanları tarafından tamamlanarak sistemin genel yapısı ortaya konulur. Kullanıcı ve alan uzmanının görüşleri doğrultusunda oluşturulan kavramsal modeller, kullanıcının istediği sistemin, sistem geliştiriciler tarafından doğru anlaşılmasını sağlar. Değişik uygulama alanlarında kullanılması rağmen kavramsal model geliştirme yaklaşımları genelde benzerlik gösterirler. Kavramsal model geliştirme temel olarak üç adıma sahiptir:

- Doğal dil ile ifade edilen isteklerinin toplanması,
- Alan uzmanlık bilgisi ile eksik yada yanlış noktaların tamamlanması,
- Kavramsal model gösterim tekniklerinden biri yada birkaçının kullanılması ile kavramsal modelin geliştirilmesidir.

Kavramsal model, kullanıcı tarafından okunabilir ve anlaşılabilir bir yapıda olduğu gibi, kavramsal modeli kullanacak olan sistem tasarımcısı tarafından kabul görecektir düzeyde teknik bilgiyi içerecek olan bir yapıda da olmalıdır (Hay,1999).

Sistem tasarımcısı açısından kavramsal modelin tamamlılığı aşağıdaki terimlerin model içerisinde gösterilmesi ile gerçekleştirilir (Hay, 1999):

- Elemanlar ve nitelikleri
- İlişkiler
- Alt ve üst tipler
- İlişkiler arası kısıtlar

Kullanıcı ve sistem tasarımcısı açısından kavramsal modelin teknik yönden “okunabilirliği”nin tam olması, modelin grafiksel gösterimine bağlıdır. Bu nedenle, kavramsal model, iyi bir grafiksel tasarım prensibine sahip olmalıdır. Bir modelin iyi bir grafiksel tasarıma sahip olması (Hay, 1999):

- Her sembolün sadece tek bir anlamı olması
- Her kavram için sadece bir tek sembolün kullanılması ile sağlanır.

Kavramsal modeller, kullanıcılar açısından istediği sistemin doğru anlaşılıp anlaşılmadığının bir kontrolü olduğu gibi, sistem tasarımcıları açısından ise sistemin nasıl olacağını gösteren ve tasarımı kolaylaştıran bir yapıdır.

II.1.6 KM Nerelerde Kullanilir?

Bilgisayar bilimleri disinda bir çok alanda kavramsal modellemenin kullanildigi ve çok sayıda kavramsal model olusturma çalismalarinin var oldugu literatür arastirmasinda ortaya çikan bir sonuçtur [örnekler için bkz.: Shekhar ve dgr.,1997, Wand ve Weber 2002, Berild, 2005, Heckel ve Guo 2004, Kunii ve dgr. 2003, URL.4, 2006, Storey, 2005, Guru & Savory, 2004, Bochicchio & Fiore, 2005]. Degisik bilimler altinda olusturulan kavramsal modellerde, çalışma alaninin farklıligindan ortaya çikan farklı yol, yöntem ve teknikler kullanılabilmektedir [Worboy, 1994, Wiegand, 1999, ICOM/CIDOC, 1999, Muth, 2001, Gomez ve dgr.,200]. Tüm bu teknik ve çalısmalarda ortaya çikan ortak sonuç, kavramsal modeller, karmaşık sistemlerin anlaşılabilirliğini arttırmak için kullanılan yapılaridir. Her disiplin kendine ait problemlerin analiz edilmesinde öncelikle kavramsal olarak sistemlerin anlaşılabilirliğini incelemiştir [Worboy, 1994, Muth, 2001, Gomez ve dgr., 2000]. Daha sonra ulasilan sonuçlar disipline ait gösterim teknikleri ile grafiksel olarak gösterilmiştir. Böylece disiplinin kendi içerisinde ortak anlaşılabilirliğe ulasmaya çalışılmıştır.

Bu çalışma kapsamında yazılım ve bilgi mühendisliği kapsamında geliştirilen ve kullanılan teknikler detayli olarak incelenmiştir.

II.2 Kavramsal Modelleme Teknikleri

Bu çalışma kapsamında teknikler bilgisayar bilimleri açısından ve iki ana başlık altında incelenecektir.

- Geleneksel kavramsal modelleme teknikleri :
 - Veri akis diyagramlari (Data Flow Diagrams – DFD–),
 - Eleman iliski diyagramlari (Entity Relationship Diagrams – ERD–),
 - Enformasyon Mühendisliği Modeli (Information Engineering Modeling),
 - Semantik Aglar (Semantic Networks– SN–),
 - Yapısal analiz ve tasarım tekniği (Structured Analysis and Design Techniques – SADT –),
 - Nesne Yönelimli Analiz (Object Oriented Analysis – OOA–),
 - Birleştirilmiş Modelleme Dili (Unified Modeling Language – UML–),

- Durum geis diyagramlari (State Transition Diagrams – STD–).
- Gelismis kavramsal modelleme teknikleri :
 - KAOS,
 - CASE,
 - IDEF1X,
 - i* ,
 - TELOS.

Asagidaki blmlerde bu teknikler ana hatlariyla incelenecektir.

II.2.1 Geleneksel Kavramsal Modelleme Teknikleri

Geleneksel kavramsal modelleme tekniklerinin en belirgin zellikleri, problemalani ile ilgili az sayida kavrami grafiksel olarak gsterebilmeleridir. Bu gruptaki tekniklerin rneklemenin en iyi yolu, bu teknikleri ynelimlerine gre gruplanmasidir. Bu gruptaki teknikler  grupta toplanabilir (Dieste ve dgr. 2000).

1. Islevsel kavramsal model teknikleri: Bu grup ierisinde yer alan teknikler problem uzayi ierisinde kullanılan verilerin deėisimlerinin tanımlanması esasına dayanır. “Verilerin deėisimi” kavrami, bu tekniklerde “sre” kelimesi ile ifade edilir. Sre TDK’nin elektronik szlėnde *“aralarında birlik olan veya belli bir dzen veya zaman iinde tekrarlanan, ilerleyen, gelisen olay veya hareketler dizisi ”* olarak tanımlanır (TDK, 2006). Bu tez alısması kapsamında “Verilerin deėisimi” yada “sre” kavrami, girdi olarak bir veri seti alan ve ıktı olarak bir veri seti veren yapı olarak kullanılmıştır. Bu tekniklere veri akis diyagramlari (DFD) bir rnektir.

2. Nesne ynelimli kavramsal model teknikleri : Bu grup ierisinde yer alan tekniklerde, problem uzayi ierisindeki, nesneler, siniflar, bunlar arasındaki ilişkiler ve etkileşimler tanımlanır. Nesne Ynelimli Analiz ve Birlesmis Modelleme Dili (UML) bu tr tekniklere rnektir.

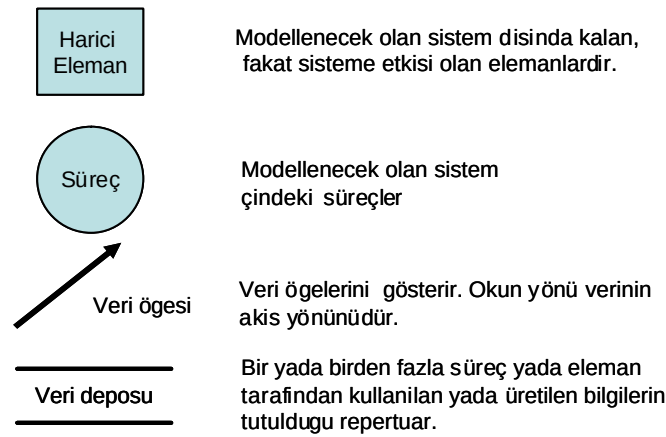
3. Durum ynelimli kavramsal model teknikleri : Bu grup ierisinde yer alan tekniklerde, problem uzayi ierisindeki nesnelerde , olaylarda zamanla meydana gelen deėisimi tanımlanır. Durum geis diyagramlari bu tr tekniklere rnektir.

Geleneksel kavramsal modelleme teknikleri ile problem uzayinin farkli yönleri ifade edilmesine ragmen; problem uzayinin tamamı tek bir klasik kavramsal modelleme tekniğı ile ifade edilmemektedir. Örneğın, veri akis diyagramları ile problem alanı içerisindeki veri akisleri ifade edilirken bu teknik ile problem uzayı içerisindeki elemanlar gösterilemez. Aşagıda, geleneksel tekniklerinden bazıları anlatılacaktır.

II.2.1.1 Veri Akis Diyagramları

Veri akis diyagramları literatür içerisinde DFD (Data Flow Diagrams) kısaltması ile kullanılmaktadır. DFD'ler yukarıdaki bölümde açıklandığı gibi klasik teknikler içerisinde işlevsel teknikler altında yer almaktadırlar. Problem uzayı içerisinde, akan bilgi çeşitli dönüşümler ile yeniden düzenlenir. DFD'ler bu dönüşümleri grafiksel olarak ifade etmek için kullanılırlar (URL.3, 2006, Pressman, 1992). Şekil II-4'de DFD'lerin gösterim notasyonu görülmektedir.

- **Dörtgen** : Harici elemanı (external entity), yani donanım, kullanıcı yada başka bir program yada sistemi etkileyen bir başka sistemi, göstermek için kullanılır. Dis eleman modelleneyecek olan sistemin sınırları dışında yer almaktadır. Ancak eylemleri yada varlığı ile sistemi etkilemektedir.



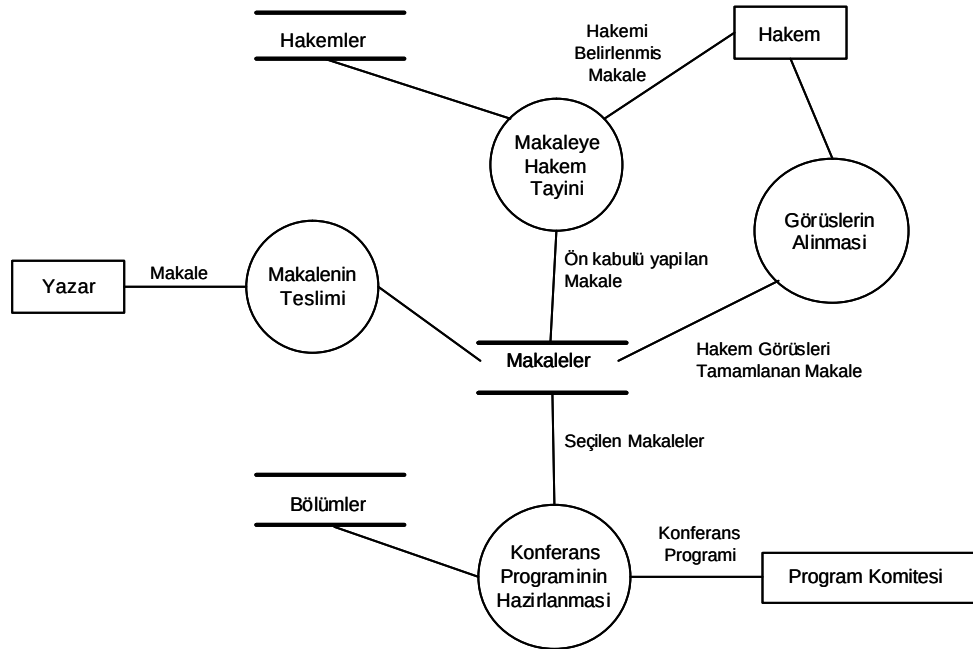
Sekil II- 4. DFD notasyonu (Pressman,1992).

- **Daire**: Süreci (process) yada veri dönüşümünü göstermek için kullanılır. Yani bir veri buraya girer ve burada işlenerek çıkar. Modellenecek olan sistemin sınırları içerisinde yer alır. Daire ile temsil edilen her bir süreç aslında sistem içerisinde birden fazla alt süreci içerebilir.

- **Ok :** Bir yada birden fazla veri ögesini (data item) ilişkilendirir. Okun yönü, veri akis yönünü gösterir. DFD’deki her ok’un yanında mutlaka bir tanımlayıcı olmalıdır.
- **Çift çizgi:** Bir veri deposunu (data store) gösterir. Daha çok yazılım sistemlerinde kullanımı olmaktadır. Islenen yada islenecek olan verilerin depolanmasını temsil eder.

DFD’ler oldukça basit notasyona sahip oldukları için oldukça yaygındır. Özellikle yazılım sistemi geliştirimlerinde isterler analizde kullanılırlar. Ancak kullanım amaçları akis diyagramları ile karıştırılmamalıdır. DFD’ler detaylı gösterimlere, döngüler, şartlar gibi, girmeden enformasyon akisini modellerler. Şekil II-5’de görüldüğü gibi, “***bir konferansın organize edilmesinin***” kavramsal modeli DFD ile gerçekleştirilmiştir. Bu modele göre, konferans dört süreçten oluşmaktadır. Bunlar

- [1] Makalenin teslimi
- [2] Makaleye hakem tayini
- [3] Görüşlerin alınması
- [4] Konferans programının hazırlanması



Şekil II- 5. Konferans Probleminin DFD ile oluşturulmuş kavramsal modeli (Dieste ve dgr. 2000).

Bu süreçlerin tanımlanması uzmanına göre değişiklik gösterebilir. Model içerisinde üç tane veri deposu tanımlanmıştır. Bunlar:

- [1] Hakemler
- [2] Makaleler
- [3] Bölümler'dir.

Modele disarindan dahil olarak sisteme etkisi olan yada sistemden etkilenen üç eleman vardır. Bunlar:

- [1] Yazar
- [2] Hakem
- [3] Program Komitesi

Bu model okunursa : Yazarlar, makalelerini teslim ederler. Teslim edilen makaleler “Makaleler” veri deposunda toplanır. Daha sonra “Hakemler” veri deposundan makaleye hakem tayini yapılır. Hakemler kendilerine gönderilen makaleleri inceler ve görüşlerini bildirirler. Görüş alınan makaleler daha sonra yine “Makaleler” deposunda toplanır. Bundan sonra daha önceden belirlenen “Bölümlere” göre makaleler ayrılarak konferans programı hazırlanır. Hazırlanan bu program program komitesine teslim edilir. Bu model konferans etkinliğinin küçük bir bölümünün genel hatları ile gösterilmesidir. Konferans etkinliği düzenlenmesi diğer teknikleri anlatırken de kullanılacaktır.

DFD üç önemli özelliği vardır.

- Herhangi bir sistem içerisindeki bilgi akışının gösterimidir.
- Süreçler daha detaylı bilgi verilmek için yeniden düzenlenebilir.
- DFD'leri kontrol akısından çok, veri akıları ile ilgilenir.

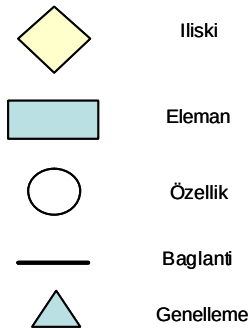
II.2.1.2 Eleman İlişki Modeli

Eleman İlişki Modeli (Entity Relationship Model –ER-) tekniği, 1976 yılında Peter Chen tarafından önerilmiştir (Chen, 1976). Ancak günümüze kadar değişik araştırmacılar bu teknik üzerine çeşitli çalışmalar gerçekleştirmiş ve çeşitli eklemeler yapmışlardır (Pressman 1992, Hay,2000). ER tekniği elemanların (entity), ilişkilerin (relationship) ve özelliklerin (attribute) gösterimini esas alır. ER'in grafiksel gösterim notasyonu aşağıdaki elemanlardan oluşur (Chen,1976, Hay,1999):

- **Dikdörtgen:** Sistem içerisindeki elemanlar “dikdörtgen” ile gösterilmekte olup içerisine elemanların adı yazılır. Elemanların özellikleri daire içerisinde elemanlara bir çizgi ile bağlanarak gösterilir. Örneğin, “Makale” ve “Makale ID”'si gibi

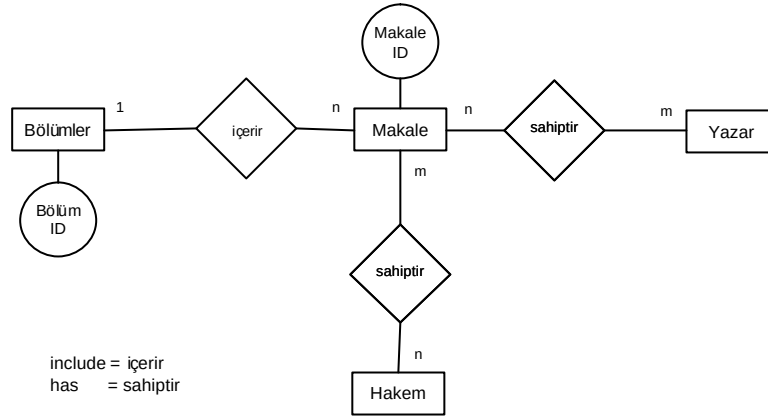
- **Baklava dilimi** : Elemanlar arasi iliskiler diger gösterim tekniklerinden farkli olarak 2 boyutlu bir sekilde “baklava dilimi” ile gösterilmektedir. Bir baklava dilimi ile bir yada birden fazla eleman birbirine baglanabilir.
- **Daire** : Elemanlara ait özellikler “daire” ile gösterilirler. Örneğin, “Makale ID”’si veya “Bölüm ID”’si gibi
- **Çizgiler** : Elemanlar, iliskiler ve özellikler arasında baglantılar ise “çizgiler” ile tanımlanır. Çizgilerin yanındaki sayılar (1, 0, m, n gibi) ise bir elemanın kaç tane diger eleman ile iliskide olabilecegini belirtir. Örneğin, “Bölümler” ve “Makale” arasında yer alan “1..n” anlamı: “ Her bölüm n tane makaleye sahip olabilir. Ancak her makale sadece 1 bölüm altında konferans programında yer alabilir”’dir.
- **Üçgen** : Genel bir eleman ve bu elemandan türeyen alt elemanlar var ise, bu genelleme “üçgen” ile gösterilir. Bu örneğimidxe üçgen sembolü kullanılmamıştır.

Sekil II-6’da ER tekniginin grafiksel gösteril notasyonu görölmektedir.



Sekil II- 6. ER notasyonu gösterimi (Dieste ve dgr., 2000).

ER diyagramlarının kullanımı arastirmacılar arasında farklılık gösterebilmektedir (Pressman, 1992). Sekil II-7’de konferans probleminin Chen’in tanımladığı Eleman iliski teknigi ile oluşturulmuş olan kavramsal modeli görölmektedir.



Sekil II- 7. Konferans Probleminin ER ile oluşturulmuş kavramsal modeli (Dieste ve dgr. 2000).

Bu model, Bölümler, Makale, Yazar ve Hakem'den oluşan dört tane elamana sahiptir. Bölümlerin ve Makale'nin her birisinin kendisine ait bir tanımlayıcısı vardır. Bu tanımlayıcı daire içerisinde elemanlara çizgi ile bağlanmıştır. Her makale “m” tane yazara sahip olabilir. Her yazarda bu etkinliğe “n” tane makale ile katılabilir. Ayrıca her makalenin incelenmesi için “n” tane hakem tayin edildiği gibi, her hakeme de “m” tane makale incelenmesi için gönderilebilir. Ancak her makale sadece ve sadece “1” bölüm içerisinde sunulabilir. Bu karsın bir bölüm içerisinde “n” kadar makale içerebilir. Bu modelin genel okunusu bu şekildedir.

Chen tarafından ortaya konulan ER modelleme tekniği üzerine, farklı disiplinlerde farklı araştırmacılar, disiplinin ve çalışmanın ihtiyacına göre bazı ekleme ya da değişiklikler yaparak, bu tekniği genişletmişlerdir. (Hay,1999). Chen tarafından ortaya konulan bu teknik, kavramsal modeller için tüm detay gösterimleri içermese bile, kavramsal modelleme de kullanılan ilk tekniklerden biridir.

II.2.1.3 Enformasyon Mühendisliği Modelleme Tekniği

Enformasyon mühendisliği modelleme tekniği (EMM), orijin olarak Avustralya'da Clive Finkelstein tarafından 1970 yılların sonlarına doğru geliştirilmiştir. Finkelstein çalışmasını Amerika ve Avrupa'da duyurmak için James Martin ile işbirliği yapmış ve bu tekniği 1981 yılında yayınladıkları “Enformasyon Mühendisliği (*Information Engineering*)” isimli çalışmaları ile duyurmuşlardır. Daha sonraki yıllarda, Martin ve Finkelstein kendi çalışmalarını ferdi olarak sürdürmüş ve kendi sürümlerini oluşturmuşlardır. Bu sürümler arasında küçük farklar vardır. EMM tekniğinde, Finkelstein, sisteme tasarımcı gözü ile bakıp, elemanları “ileri referanslar için saklanacak veriler” olarak görür iken, Martin, sisteme analist gözü ile

bakip, elemanlari “hakkında bilgi toplanacak olan gerçek yada soyut sey” olarak görür. EEM teknigi grafik gösterimde :

- **Dikdörtgen:** Elemanlari göstermektedir. Elemanların isimleri direkt olarak dikdörtgen içersine yazılmaktadır.

- **Çizgiler :** Elemanlar arasi iliskileri temsil etmektedir.

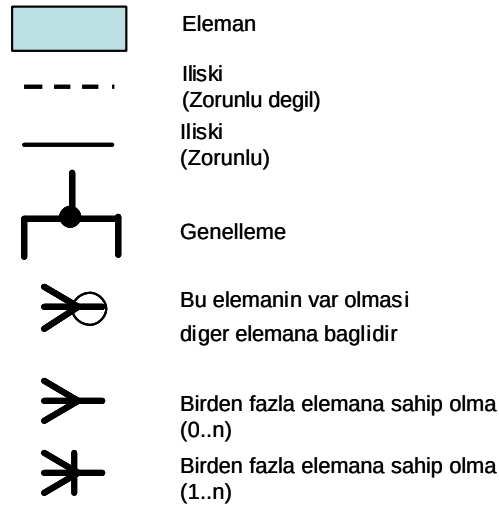
- **İçi dolu nokta (daire) :** Genel bir eleman ve bu elemandan türeyen alt elemanlar var ise, bu genelleme “daire” ile gösterilir.

- **Daire :** Eger ilk elemanın var olması ikinci elemana bağlı değil ise bu durum ikinci elemanın yanında küçük bir daire gösterilir.

- **Dik Kısa Çizgi:** İlk elemanın var olması ikinci elemanın mutlaka var olması gerekli ise, yani bir kompozisyon (bilesim, olusum) ifade ediliyor ise, o zaman bu durum ikinci elemanın yanında dikey, kısa çizgiler konularak gösterilir.

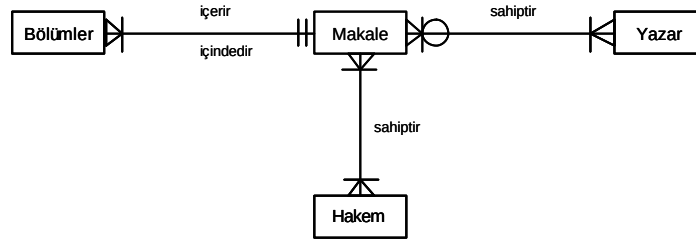
- **Ayakli Çizgi :** İlk eleman birden fazla ikinci elemana sahip olabiliyor ise, o zaman ikinci elemana yakın yerde ayakli bir çizgi bulunur

Sekil II-8’de EMM’nin genel grafik gösterim notasyonu görülmektedir.



Sekil II- 8. EMM notasyonu (Hayy, 1999).

Sekil II-9'da ise bir EMM ile oluşturulmuş “konferans probleminin” kavramsal modeli görülmektedir. Bu model de, elemanlar, dikdörtgen ile gösterilmektedir. Eleman isimleri, dikdörtgenlerin içine yazılmaktadır. EMM notasyonuna dikkat edildiğinde özelliklerin bu teknik içerisinde gösterilmediği görülecektir. Finkelstein'in sürümüne göre özellikler “Eleman Listesi (Entity List)” adlı belge içerisinde tutulmaktadır. Martin sürümüne göre ise özellikler “kabarcık semasi (bubble chart)”’ında, diğer adı ile veri akis semalarında gösterilmektedir (Hayy,1999). Bu kavramsal modelde dört tane eleman bulunmaktadır. Bunlar, Bölümler, Makale, Hakem ve Yazar'dır. Bu elemanlardan “Bölümler”, 1..n'e kadar “Makale”yi içerir. Tersten okundığında ise “Makale” sadece 1 bölüm içindedir. Bir makalenin var olması için en az 1 tane yazara sahip olmalıdır. Yazar'lar 1..n kadar makaleye sahip olabilirler. Her makale 1..n'e kadar hakeme yollanabilir. Aynı zamanda her hakemde 1..n'e kadar makaleyi değerlendirebilir.



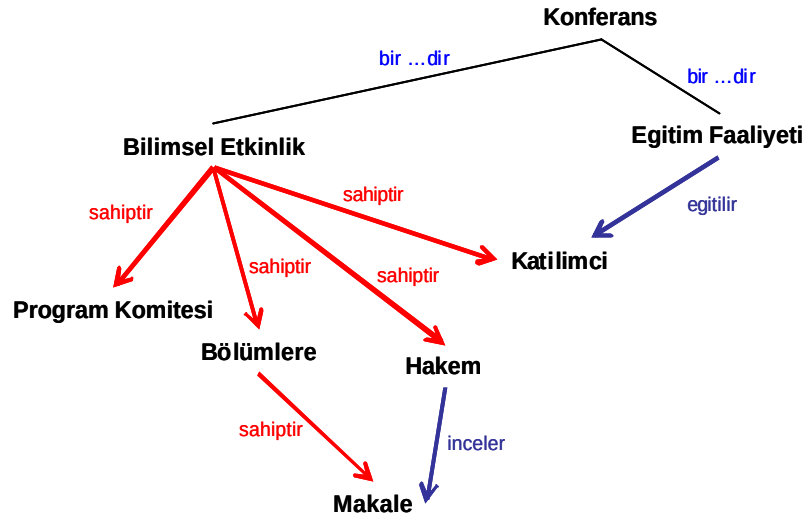
Sekil II- 9. Konferans Probleminin EMM ile oluşturulmuş kavramsal modeli.

Bu teknik ilk bakıldığında pratik olarak görülmesine rağmen, özelliklerin model içerisinde gösterilmemesinden dolayı, modelin okunabilirliği ve kullanılabilirliği azalmaktadır. Bu durumda da, modelin kullanıcı ve tasarımcı tarafından anlaşılması zorlaşmaktadır.

II.2.1.4 Semantik Aglar:

1966 yılında Ross Quillian, çalışmada semantik ağı kullanarak insan belleğini modellemeye çalışmıştır (Mylopoulos, 2001). Sematik ağ ile modellemede, kavramlar ve ilişkiler vardır. Bu modelleme üzerinde nitelik kalitimleri rahatça görülebilir. Bu kalitim kesin yada varsayılandır. Tek bir kalitim olduğu gibi birden

fazla kalitimda olabilir. Sekil II-10'da konferans problemi için



Sekil II- 10. Semantik Ağ ile bir kavramsal model (Mylopoulos, 2001).

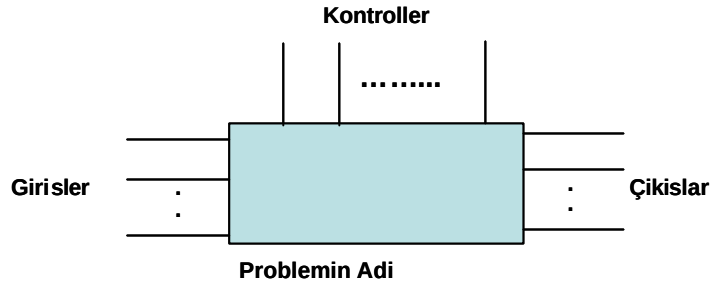
semantik ağ ile oluşturulmuş model görülmektedir. Bu yapıya da dikkat edilecek olur ise tümdengelim yapısında bir çözümle yapıldığı görülecektir. Aralardaki ilişki “bir...dir (is-a)” ile ifade edilmektedir. Ayrıca, kalımlar ve ilişkilerde gösterilmektedir. Semantik ağlar da bir tür klasik kavramsal modellerdir. Kullanıcı yönünden okunabilirliği kolaydır. Bu modelin okunmasında, Konferans bir Bilimsel Etkinliktir. Bilimsel Etkinlikler Program Komitesine, Bölümlere, Hakemler ve Katılımcılara sahiptir. Bölümler ise Makale'lere sahiptir. Her makale Hakemler tarafından incelenir. Konferans aynı zamanda bir eğitim faaliyetidir. Katılımcılar eğitilir.

Semantik Ağlarda kullanılan grafik notasyonunda Ok'lar “Özellikleri”, düz çizgiler ise “Genelleme”yi ifade eder.

II.2.1.5 Yapısal Analiz ve Tasarım Tekniği

Semantik ağ ve ER tekniğinin duyurulmasından sonra, Doug Ross tarafından “fikirlerin iletisimi için bir dil” sloganı ile ortaya atılan Yapısal Analiz ve Tasarım Tekniği (Structured Analysis and Design Technique -SADT-), yazılım sistemlerinin isterlerinin belirlenmesinde daha çok kullanım alanı bulmuştur. Bu teknige göre bir yazılım sisteminin işlevsel çevresinin modellenmesinde kullanılması için faydalıdır. Uygulama veri ve eylem kavramları ile modellenir. SADT modellerine bakıldığında

kutu içerisinde kutu şeklinde bir notasyon karsimiza çikar. Sekil II-11’de SADT modelinin genel notasyonu görölmektedir.



Sekil II- 11. SADT modeli genel notasyonu.

Bu yapıya göre bir problemde, Giris, Kontrol ve Çikis olmak üzere 3 tür veri akisi gerçekleşir. Girisler dikdörtgenin solundan giren oklar ile; Kontroller, dikdörtgenin üstünden giren oklar ile; Çikislar ise dikdörtgenin solundan çıkan oklar ile gösterilir.

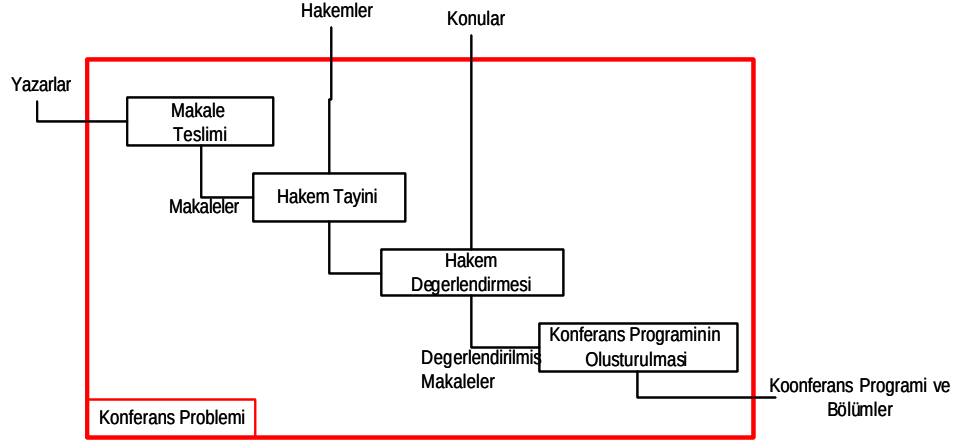
Bu notasyona göre,

•**Büyük Dikdörtgen:** Problem alanini ifade eder. Problemin adi dikdörtgenin sol altına yazilir. Bu alan içerisinde ise yer alan *sekiller ile model çizilir*. Bu sekiller:

- o **Küçük Dikdörtgen** : Eylemleri
- o **Oklar** : Eylemlerin akis yönünü
- o **Etiketler** : Okların yanında bulunan etiketler ise her bir eylemin çıktisini gösterir.

•**Oklar:** Veri akislerini ifade ederler.

Sekil II-12’de “konferans probleminin” SADT ile oluşturulmuş kavramsal modeli görölmektedir.

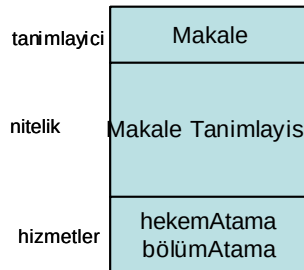


Sekil II- 12. Konferans Probleminin SADT ile oluşturulmuş kavramsal modeli.

II.2.1.6 Nesne Yönelimli Analiz

Nesne yönelimli analiz (NYA), uygulama alanının nesne tabanlı modelini çıkartmayı amaçlar (Sommerville, 2001). Çözülmesi amaçlanan problem alanı içerisindeki elemanları ve eylemleri nesneler ile ifade eder. NYA tekniği nesneleri yapılandırmak için sınıflandırma, genelleme ve kümeleme yöntemlerini kullanır. Eylemler nesneler ile birleştirilmiştir. Bir nesne üzerinde meydana gelen eylem, nesnenin durumunu değiştirir.

1991 yılında Peter Coad tarafından duyurulan analiz tekniğinde problem alanı ile ilgili elemanlar ve nesneler ile tanımlanır (Coad ve Yourdon 1991). Her bir nesne kendi özellik ve davranışını kapsar. Coad'ın modelinde bir uygulamanın modellenmesi için beş tür kavram vardır : nesneler, nitelikleri, yapıları, hizmetleri ve özneleri. Sekil II-13'de Coad modeli ile ortaya konulan bir yapı görülmektedir.

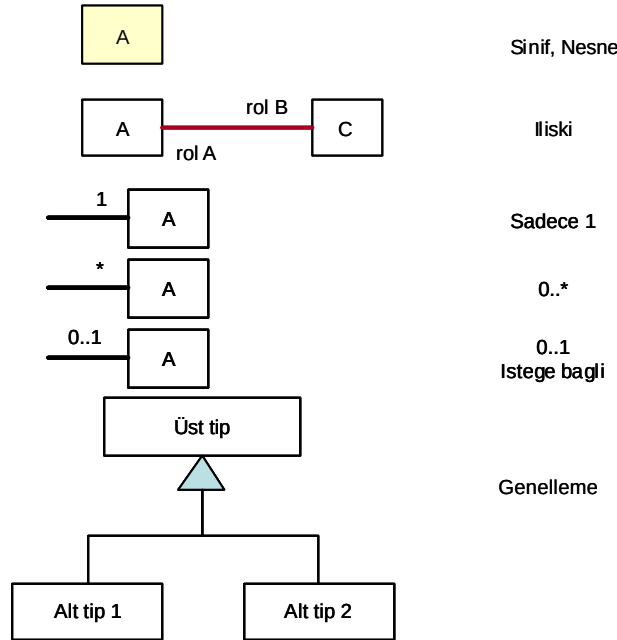


Sekil II- 13. Coad modeli

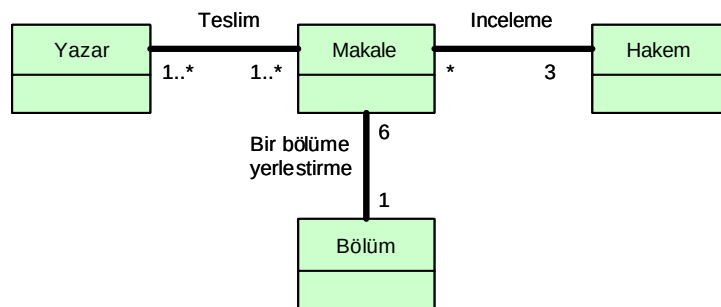
II.2.1.7 Birleştirilmiş Modelleme Dili

Birleştirilmiş modelleme dili (UML) Booch, Rumbaugh ve Jacobson'un nesne yönelimli analiz tekniklerinin birleştirilmiş halidir (Fowler ve Scoot 2001). UML,

tasarım işlevlerini kolaylıkla anlatmak için kullanılan, genellikle grafiksel notasyonlardan oluşan bir yapıdır. Şekil II-14’de UML notasyonundan örnekler görülmektedir. Bu teknik oldukça geniş bir grafiksel göstergim notasyonuna sahiptir. Bu notasyonun tamamı için çalışması incelenebilir. Şekil II-15’te ise konferans probleminin UML ile oluşturulmuş bulunan nesne diyagramı görülmektedir. Bu modele göre Yazar, Makale’yi teslim eder. Bundan önceki modellerde olduğu gibi Bir yazar 1..* kadar makaleye sahip olabilir. Her makalede 1..* kadar yazara ait olabilir. Makale’ler 3 Hakem tarafından incelenir. Her hakem * tane makale inceleyebilir. Her makale 1 Bölüm içerisinde yer alır. Her Bölüm 6 makaleye sahip olmalıdır. Bu modele bakan kişi model üzerinden bu bilgileri anlamalıdır.



Şekil II- 14. UML notasyonu (Fowler ve Scoot 2001).



Sekil II- 15. Konferans Probleminin UML notasyonu ile oluşturulmuş kavramsal modeli (Dieste ve dgr. 2000).

II.2.1.8 Durum Geçiş Diyagramları

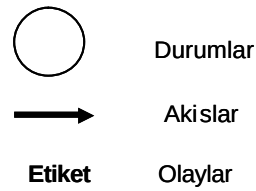
Simdiye kadar anlatılan kavramsal modelleme teknikleri incelendiğinde görülecektir ki, olaylar zaman sıralaması içerisinde anlatılmazlar. Olayların zaman sırasını anlatılmasının önemli olduğu durumlarda durum geçiş diyagramları kullanılır. Durum geçiş diyagramları (DGD) kullanılarak oluşturulan kavramsal modeller içerisinde veri yapıları ve veri dönüşüm bilgileri bulunmaz. Şekil II-16’da DGD’da kullanılan notasyon görülmektedir. Bu notasyona göre,

•**Daire** : Durumları gösterir.

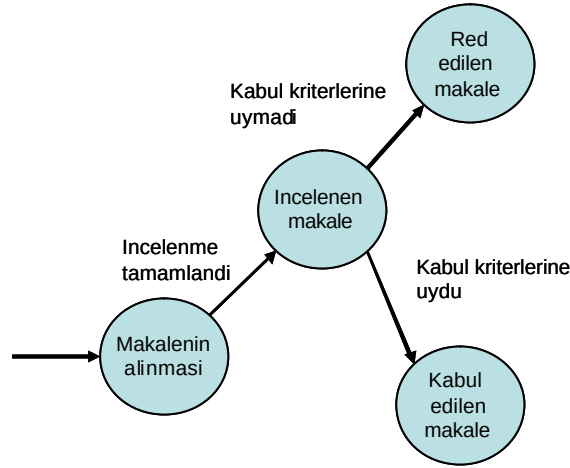
•**Oklar** : Olayların zamanlama sırasını yani akışını gösterir.

•**Etiketler** : Okların yanında bulunan etiketler ise gerçekleşen olayları gösterir.

Şekil II-17’de konferans probleminin durum geçiş diyagramları ile oluşturulmuş olan kavramsal modeli görülmektedir.



Sekil II- 16. DFD Diyagramları notasyonu.



Sekil II- 17. Konferans Probleminin durum geiş diyagramları ile oluşturulmuş kavramsal modeli (Dieste ve dgr. 2000).

Bu modelde esas olan durumlardır. Konferans yapısı içerisinde dört durum vardır. Bunlar makalenin alınması, İncelenmesi, Red edilmesi ve kabul edilmesi durumlarıdır. Makalenin alınmasından incelenen makaleye geiş inceleme tamamlandıktan sonra gerekleşir. Eğer makale kabul kriterlerine uyarsa kabul edilir. Uyuz ise red edilir. Bu yapıda aktör ve elemanlar gösterilmez.

Buraya kadar anlatılan geleneksel kavramsal modelleme teknikleri, çözümleri istenen problemlerin tüm yönleri ile tek bir kavramsal model içerisinde gösterilmesi için yeterli gösterim gücüne sahip değildir. Ancak bu tekniklerden birkaçı bir araya getirilerek problem alanı ile ilgili genel bir kavramsal model oluşturulabilir. Ayrıca, geleneksel teknikler ile oluşturulan kavramsal modeller içerisindeki kavramlar ya kullanıcı perspektifine ya da tasarımcı perspektifine göre tanımlanır (Dieste ve dgr. 2000). Örneğin, DFD’de problem alanı içerisinde yer alan veri akışlarının tanımlanması yapılır iken, nesne diyagramlarında, nesnelerin tipleri ve nesneler arası ilişkiler tanımlanır. DFD’ler kullanıcı açısından tercih edilirken, nesne diyagramları tasarımcı açısından tercih edilir.

Geleneksel tekniklerin kısıtlarının giderilmesi ve problem alanının daha geniş bir yapıda görülebilmesi için çeşitli çalışmalar gerekleştirilmiştir. Bu çalışmalar sonunda aşağıdaki bölümde anlatılan gelişmiş kavramsal modelleme teknikleri ortaya konmuştur.

II.2.2 Gelişmiş Kavramsal Modelleme Teknikleri

Geleneksel kavramsal model tekniklerinin 2 temel kısıtı vardır.

1. Birincisi, kullanılan geleneksel kavramsal modelleme tekniginin kabiliyetine göre, problem alanı ile ilgili sınırlı sayıda kavram tanımlanır. Oysaki, geliştirilmesi istenen bir sistem içerisinde çok sayıda kavram, girdi, çıktı, işlem, etmen, veri, ilişki, durum ve geçiş bulunabilir. Tüm bunların hepsinin birden tek bir teknik kullanılarak gösterilmesi, klasik teknikler açısından mümkün değildir.

2. İkinci kısım ise, birinci kısmın sonucu ortaya çıkmaktadır. Tüm problem alanının ifade edilmesi için birden fazla kavramsal modelin oluşturulması gereklidir. Örneğin, bir problemin kavramsal modelinin oluşturulmasında, problem içindeki nesnelerin belirlenmesinde nesne yönelimli analiz diyagramları, problem içerisindeki veri akışlarının ortaya konulması ve kullanıcı açısından problem alanının analist tarafından nasıl algılandığının gösterilmesi için DFD'lerin, ve durum geçiş diyagramlarının oluşturulması gerekir.

Geleneksel kavramsal modellerin bu kısıtlarının ortadan kaldırılmasının iki yolu vardır.

- Mevcut klasik tekniklerin ontolojisini zenginleştirmek. Yani teknikler tarafından ifade edilebilen kavramların çeşidini ve sayısını arttırmak.

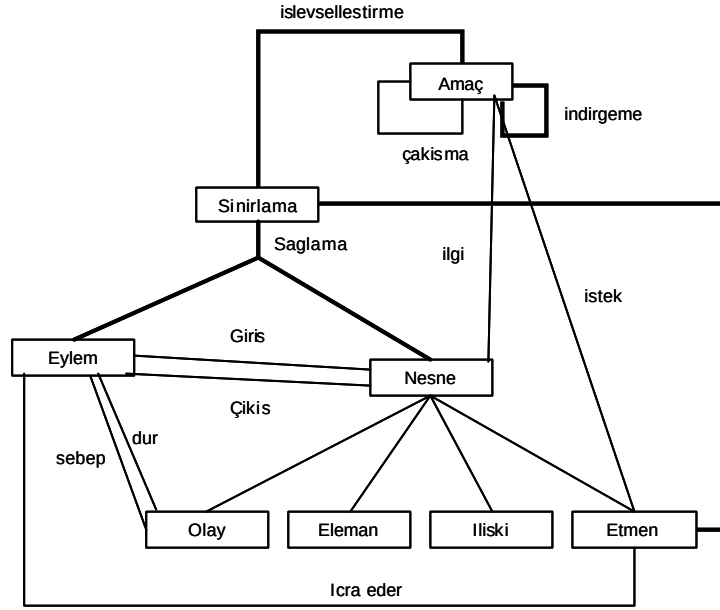
- Kullanıcıya gerekli olduğunda yeni kavramlar tanımlama imkanını vererek ontolojiyi kullanıcının genişletmesini sağlamak.

Bu noktadan hareketle mevcut kavramsal modelleme teknikleri geliştirilmiş ve KAOS, CASE, IDEFIX, i*,TELOS gibi yeni gelişmiş kavramsal modelleme teknikleri önerilmiştir.

II.2.2.1 KAOS

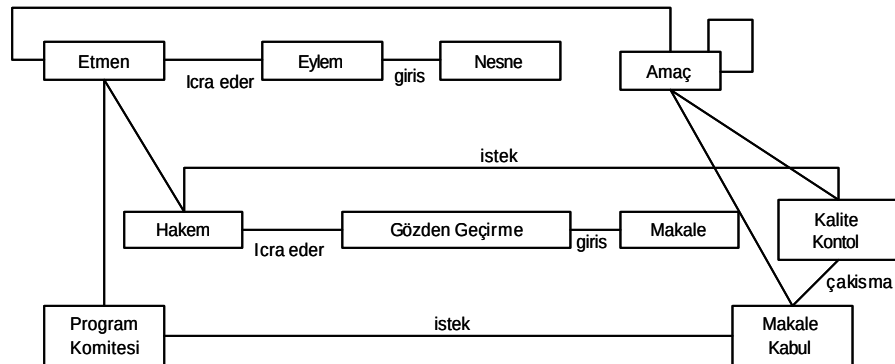
Lamsweerde ve arkadaşları tarafından tanımlanan KAOS (Knowledge Acquisition in automated Specification) bilgi mühendisliğinde kullanılan gelişmiş kavramsal model tekniklerinden biridir (Lamsweerde ve dğr. 1991). KAOS geniş bir gösterim notasyonuna sahiptir. Gösterim notasyonunun geliştirilmesi için meta-model tanımlamalarını kullanılır. *Meta-Model*, kavramsal model içerisinde gösterilecek olan kavramların tanımlanmasıdır. Meta-model tanımlamalarında genellikle Eleman İlişki Modellerindeki gösterim sembolojileri kullanılır. Şekil II-18'de bir KAOS meta-modeli görülmektedir. Bu meta-model de görüldüğü gibi,

KAOS ile klasik modeller içinde gösterilemeyen amaç, etmen (agent), sinirlama, eylem gibi kavramların tanımı yapılmaktadır. Böylece bu elemanlar artık kavramsal model içerisinde gösterilebilir. Aynı zamanda, meta model içerisinde gerekli tanımlamalar yapılır ise, KAOS hem statik, hem de dinamik olayları kavramsal model içerisinde gösterilebilir (Dieste ve dgr. 2000).



Sekil II- 18. KAOS meta-modeli örneği (Dieste ve dgr. 2000).

Sekil II-19'da konferans modelinin KAOS ile gösterimi görülmektedir. KAOS ile ilgili detaylı anlatımlar. Lamsweerde ve arkadaşları tarafından gerçekleştirilen çalışmada bulunabilir (Lamsweerde ve dgr. 1991).

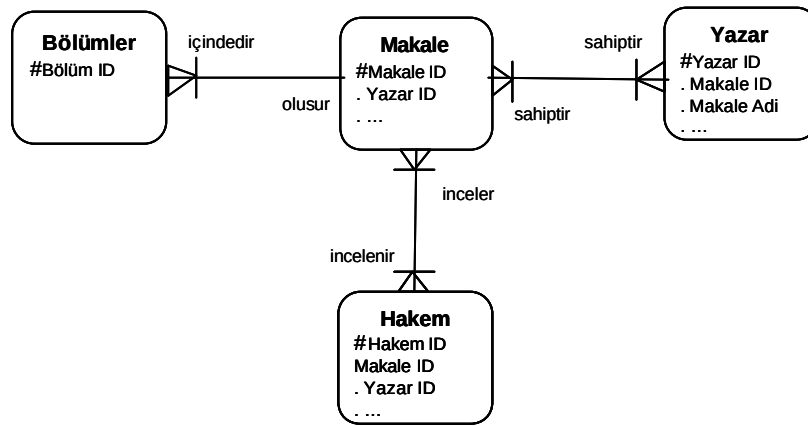


Sekil II- 19. Konferans probleminin KAOS ile gösterimine örnek (Dieste ve dgr. 2000).

Bu modele bakildiginda, Etmen'in Eylemleri icra ettigini ve Eylemlerinde Nesne'leri giris olarak kullandiklarini görölür. Ayni zamanda model içerisinde bir Amaç tanımlaması da mevcuttur. Bunların tanımlaması meta model ile gerçekleştirilmiştir. Konferans probleminde 2 etmen vardır: Hakem ve Program Komitesi. Hakemler Makale'leri giris olarak alarak Gözden Geçirme eylemini gerçekleştirirler. Amaç Konferans için kalite kontrolü sağlamaktır. Eger makale kalitesi ile konferansi amaçları ve diğer kriterleri uyarsa bir çıkışma olur. Böylece makale kabul edilir. Program Komitesinin amacı Konferans Etkinliği için istegi kabul edilen makalelerdir.

II.2.2.2 CASE

Bu teknik ilk olarak İngiliz şirketi CACI tarafından geliştirilmiş, daha sonra Richard Barker tarafından geliştirilerek, Oracle'n CASE*Method projesine adapte edilmiştir (Barker, 1990). Bu nedenle bu teknik literatürde Barker tekniği olarak da isimlendirilir. Sekil II-20'de bu teknik kullanılarak hazırlanan bir kavramsal model örneği görölmektedir.



Sekil II- 20. Konferans probleminin CASE* ile gösterimine örnek.

Bu teknik içerisinde elemanlar (entity) yuvarlak köseli dikdörtgenler ile gösterilirler. Elemanlara ait özellikler ise bu dikdörtgenlerin içerisine yazılır. Ancak, özelliklerin gösteriminde eleman ilişki diyagramlarında olduğu gibi küçük yuvarlak dairelerde kullanılabilir. Eger özelliklerin gösteriminde daireler kullanılacak ise bu durumda istege bağlı özellikler normal daireler ile zorunlu olan özellikler ise koyu renkli küçük daireler ile gösterilir. Elemanlar arası ilişkiler ise diğer kavramsal modelleme tekniklerinin bir çoğunda olduğu gibi çizgiler ile gösterilir. İlişki kesikli çizgi ile gösterilmiş ise bu ilişkinin zorunlu olmadığını, düz sürekli bir çizgi ile

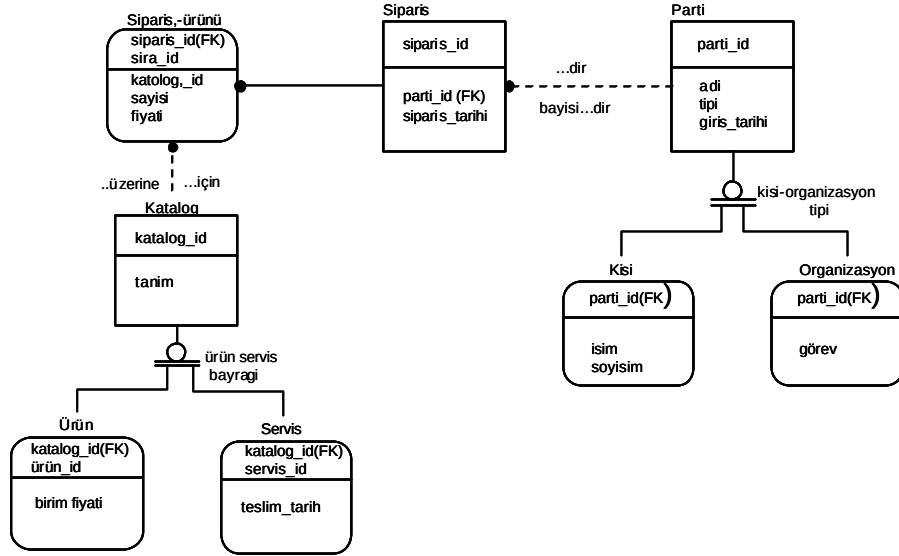
gösterilmiş ise bu ilişkinin zorunlu olduğunu gösterir. CASE yada diğer adı ile Barker tekniği ile ilgili detaylı bilgiler Barker tarafından veya Hay tarafından gerçekleştirilen çalışmalarda bulunabilir (Barker, 1990, Hay, 1999).

CASE* ile oluşturulmuş olan bu model üzerinde Bölümler, Makale, Hakem ve Yazar elemanları bulunur. Bu elemanlar yuvarlak köseli dikdörtgenler ile gösterilmiş ve özellikleri dikdörtgenin içine yazılmıştır. Tüm ilişkiler sürekli çizgi ile gösterilmiştir. Bunun anlamı tüm ilişkiler zorunludur. Yani Yazar ile Makale mutlaka birbirini gerektirir. Yazarsız bir makale düşünülemez. Aynı zamanda her makale mutlaka en az bir Hakem tarafından incelenmesi gereklidir. Üçgenin önündeki | (dik çizgi) çizgi bunu gösterir.

II.2.2.3 IDEF1X

Ağırlıklı olarak ABD’lerinde gerçekleştirilen projelerde kullanılan bir tekniktir (Hay,1999). Bu teknik içerisinde Elemanlar köseli ve köseleri yuvarlak dikdörtgenler ile gösterilirken, elemanlara ait özellikler bu dikdörtgenlerin içine yazılır. Köseli dikdörtgenler bağımsız elemanları gösterirler. Bu elemanlar kendilerine ait bir tanımlayıcıya sahiptir ve diğer bir elemandan türetilmiş değildir. Köseleri yuvarlak dikdörtgenler ise bağımlı elemanları gösterirler. Bu elemanlar da kendilerine ait bir tanımlayıcının yanında diğer bir eleman ile ilişkiye sahiptir. Elemanların ismi dikdörtgenlerin dışına yazılır. Dikdörtgenlerin içi ikiye ayrılmıştır. Dikdörtgenlerin üst kısmına elemanı tanımlayıcı öznitelikler yazılırken, alt kısmına ise tanımlayıcı olmayan genel nitelikler yazılır.

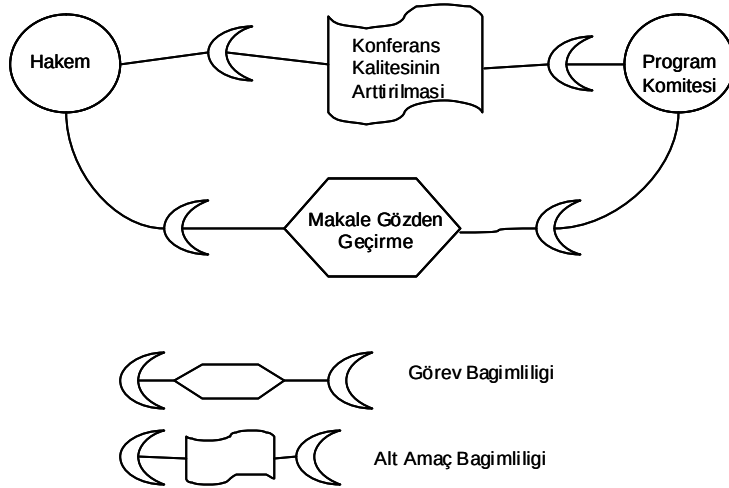
IDEF1X’de ilişkiler düz ve kesikli çizgiler ile gösterilir. Eğer iki eleman arasından bir ilişki var ise veri tabanlarında olduğu gibi türeyen eleman üst elemanın tanımlayıcısını kendi içerisinde gösterir. Şekil II-21’de ise bir IDEF1X model örneği görülmektedir. IDEF1X ile detaylı bilgi ve diğer teknikler ile karşılaştırılması Hay tarafından 1999 yılında gerçekleştirilen çalışmada bulunabilir (Hay, 1999).



Sekil II- 21. IDEF1X Model örneği (Hay, 1999).

II.2.2.4 i*

Gelişmiş kavramsal modelleme teknikleri arasında yer alan i* Yu tarafından doktora çalışması kapsamında tanımlanmıştır (Yu, 1995, Dieste ve dgr. 2000). i*'da çalışma alanı içerisinde yer alan farklı aktörlerin birbirleri ile etkileşimlerinin tanımlanması mümkündür. Bu aktörler insan olabileceği gibi alan içerisinde yer alan elemanlarda olabilir. i* birbirinden farklı ve birbirinin tamamlayıcısı olan iki alt modelden oluşur. Bunlar Stratejik Bağımlılık Modeli (SBM) ve Stratejik Mantık Modeli (SMM)'dir. SBM'de aktörlerin kendi görevlerini yerine getirebilmeleri için gerekli olan diğer aktörler ile aralarındaki bağlantıları gösterilir. Bağlantıyı sağlayan aktör bağlayıcı, diğerine de bağlanan denir. SMM'de ise her bir aktörün kendi iç işlerini gerçekleştirirken yapmış olduğu işlemler gösterilir. i* içerisinde amaç bağlantısı, görev bağlantısı, kaynak bağlantısı ve alt amaç bağlantısı olmak üzere dört tür bağlantı vardır. Bu bağlantılar ve i* ile detaylı bilgi Yu tarafından 1995 yılında tamamlanan çalışmasında bulunabilir (Yu,1995). Sekil II-22'de alt amaçları gösteren bir i* modeli görülmektedir.

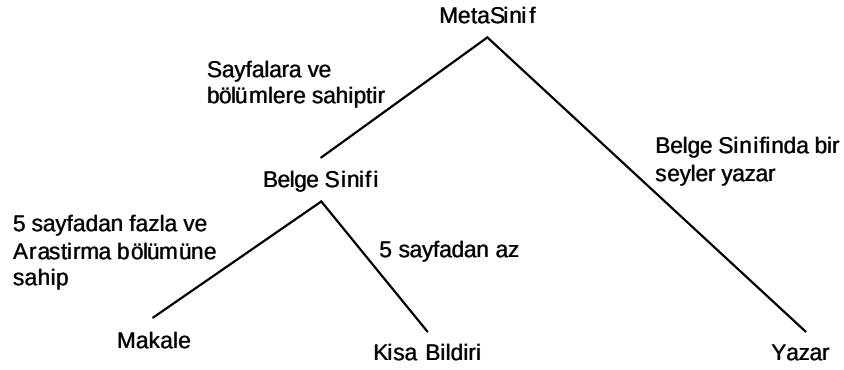


Sekil II- 22. i* Model örneği (Dieste ve dgr. 2000).

Bu kavramsal model göre Hakem ile Program Komitesi arasında bir görev bağımlılığı ile bir alt amaç bağımlılığı vardır. Program Komitesi gözden geçirilmesi için makaleleri Hakem'e iletmelidir. Hakem kendisine iletilen makaleleri gözden geçirmelidir. Program komitesinin ve Hakem'in bir amaç birliği vardır. O da düzenlenecek olan konferansın kalitesinin arttırılmasıdır.

II.2.2.5 TELOS

TELOS, Mylopoulos ve arkadaşları tarafından duyurulan diğer bir gelişmiş kavramsal modelleme teknigidir (Mylopoulos ve dgr. 1990). TELOS, nesne yönelimli programlamada kullanılan sınıf kavramı üzerine geliştirilmiştir. Bundan dolayı sınıflar üzerinde gerçekleştirilen genelleme (aggregation), sınıflandırma (classification) ve miras (inheritance) gibi nesne yönelimli programlamada kullanılan işlemleri destekler. Sekil II-23'de konferans probleminin TELOS ile gerçekleştirilmiş kavramsal modelinin bir bölümü görülmektedir. TELOS ile ilgili detaylı anlatımlar. Mylopoulos ve dgr. 1990'da gerçekleştirilen çalışmada bulunabilir.



Sekil II- 23. TELOS Model örneği (Dieste ve dgr., 2000).

TELOS ile olusturulan Konferans Kavramsal modele göre Meta Siniftan, Belge Sinifina genelleme vardır. Belge sinifi sayfa ve bölümlere sahiptir. Belge Sinifi’da Makale ve Kisa Bildiri siniflarina bir genelleme vardır. Makaleler 5 sayfadan fazladir ve arastirma bölümüne sahiptir. .Kisa Bildirinin özelligi 5 sayfadan az olmasidir. Meta Sinifinda türeyen Yazar, Belge sinifinda bir seyler yazar.

II.2.3 Kavramsal Modelleme’de Önceki Çalışmalar

Literatür taramasinda, kavramsal modelleme alanında farklı disiplinler içerisinde gerçekleştirilen çalışmalar görülmüştür [örnekler için bkz.: Wiegand, 1999, ICOM/CIDOC, 1999, Wand ve Weber 2002, Berild, 2005, Heckel ve Guo 2004, Kunii ve dgr. 2003, URL.4, 2006, Zhang ve Chen, 2005, Andrade ve dgr. 2004, Khatri ve dgr. 2004, Boichichio ve Fiore, 2005, Guru ve Savory, 2004]. Bu çalışmalardan bazıları kavramsal modelleme teknikleri ile ilgili olup önceki bölümde, özellikle gelişmiş kavramsal modelleme teknikleri başlığı altında anlatılmıştır. Bu bölüm içerisinde ise kavramsal modelleme için bir teknik geliştirmekten çok kavramsal modelleme için yazılım geliştirmeye yönelik çalışmalar incelenmiştir.

“The Toolkit for Conceptual Modeling (TCM)” kavramsal modeller olusturmak için geliştirilen bir yazılımdır. Bu çalışma Hollanda’da Dehne ve arkadaşları tarafından gerçekleştirilmiştir. TCM, yazılım sistemlerinin diyagram, ağaç yada tablolar şeklinde kavramsal modellerinin olusturulmasını sağlayan bir yazılımlar koleksiyonudur. Bu çalışma kapsamında “kavramsal model” tanımı Dehne & Zandschulp tarafından şu şekilde yapılmıştır: “bir sistemin davranışını ya da ayrıştırılmasını gösteren yapıdır”. Bu nedenle TCM içerisinde, Eleman İlişki

Modellemesi, Veri ve Olay akis modellemesi, durum geis modellemesi ve UML editörlerine sahiptir. Bu yazilimin zekilik özelliđi yoktur, tamamen kullanıcı bazlı alışır. TCM’in alışma platformu Unix işletim sisteminde X Windows’tur. TCM GNU General Public License (GPL) ile dağıtılır. TCM ilgili belgeler ve kodlara <ftp://ftp.cs.utwente.nl/pub/tcm> ulaşılabilirdiđi gibi, TCM’in web ana sayfası ise <http://www.cs.utwente.nl/~tcm> ‘dir.

Kavramsal modelleme iin geliştirilen diđer bir yazılım da ICOM (Intelligent Conceptual Modeling)’dur. ICOM kullanıcıya genişletilmiş ER (eleman iliski) diyagramları ile kavramsal model geliştirmesine yardımcı olan bir yazılımdır. Zekilik özelliđine sahip olan bu yazılım veri tabanı modellemesinde kullanıcı belirtimlerini mantıksal olarak tamamlamaya alışır. Ayrıca oluşturulan ER diyagramlarında tutarsizlikler var ise bununla ilgili uyarılarda bulunur. ICOM alışmasının amacı, veritabanı tasarımlarında kullanılacak olan ve bilgilerin ER diyagramlarında gösterilmesini sağlayacak bir kavramsal modelleme aracı olusturmaktır. Bu yazılım su an sadece veri tabanı modellemesinde ER diyagramlarını desteklemektedir. UML notasyonunda kavramsal model olusturulması iin alışmalar devam etmektedir. Bu yazılım hakkında detaylı bilgi “<http://www.cs.man.ac.uk/~franconi/icom/> ” web sayfasından elde edilebilir.

ICOM’un “Zeki kavramsal modelleme (Intelligent Conceptual Modeling)” başlığı bu tez alışmasının başlığı ile benzerlik göstermektedir. Ancak ICOM ile bu alışmanın amaçları ve uygulama alanı farklıdır. ICOM veri tabanlarını hedef alıp, klasik kavramsal modelleme tekniklerinden “Eleman İliski Modelini” kullanırken, bu alışmada ise sistemler arası bir kavramsal modellemeyi ve bu alışma kapsamında önerdiği grafiksel gösterim yapısını kullanmaktadır.

II.3 Yapay Zeka

Yapay zeka, insanın düşünme sistemini anlayarak benzeri bir yapıyı ortaya ıkarmayı amaçlayan bir bilgisayar bilimidir. Literatürde yer alan diđer bir tanıma göre ise yapay zeka, “zeki davranışların otomasyonu ile ilgilenen bir bilgisayar bilimidir (Russell ve Norvig 1995). Yapay zekada, insana özgü bilgi edinme, algılama, düşünme ve karar verme gibi yeteneklerin benzerlerinin bilgisayar programları ile gerçekleştirilmesi amaçlanmaktadır. Farklı bilim dalları yapay zeka alanındaki gelişmelerden etkilenmiş ve bu etkilesimin bir sonucu olarak, doğal dil

isleme, bulanik mantik, genetik algoritmalar, yapay sinir aglari, uzman sistemler ve zeki etmenler gibi uygulamalar ortaya çıkmıştır.

Yapay zekanın sağlamış olduğu avantajlar aşağıda özetlenmiştir (Öztemel ve dgr. 2001) :

- İş yükünü azaltır.
- Analiz ve değerlendirmede objektiflik sağlar.
- Bu sistemler ile uzmanlık bilgisine kolay ulaşım vardır.
- Karmaşık ve çok parametrelili bilgilerin işlenmesini kolaylaştırır.
- Bilgilerin ve algoritmaların kullanımında esneklik sağlar.
- Kullanıcı ile bilgisayar sistemleri arasında kullanıcı dostu yapılar sağlar.
- Sistem güvenilirliğini ve etkinliğini artırır.
- Sağlanan çözümlerin sebeplerini açıklayabilirler.

Yapay zekanın bu avantajlarının yanında bazı dezavantaj ve kısıtlamalarda sahiptir. Bunlar (Öztemel ve dgr. 2001) :

- İnsan bilgisinin bilgisayar sistemlerine transfer edilmesi, doğru bilgilerin tedarigi, işlenmesi zor ve zaman gerektirir.
- İnsandaki bilgiler genellikle belirli alanlara aittir. Bu nedenle yapay zeka programları genellikle alan bağımlidir.
- Yapay zeka sistemleri, özellikle zeki sistemler, bir çözüm sağlarlar. Ancak bu çözüm kesin ve en iyi çözüm değildir.
- İnsan davranışlarının doğru şekilde analiz edilmesi ve modellenmesi oldukça zordur.

Yukarıda verilen avantajlar ve dezavantajların sayısı artırılabilir ve literatür içerisinde bu konu hakkında birçok çalışma bulunabilir. Yapay zeka çalışmaları değişik teknolojilerin doğmasına neden olmuştur. Çoğu laboratuvar çalışmaları düzeyinde olan 60'dan fazla yapay zeka teknolojisi vardır (Öztemel, 2003).

II.3.1 Kural Tabanlı Uzman Sistemler:

II.3.1.1 Bilgi Nedir, Uzman Kimdir ?

Uzman sistemler, uzmanın görevini yerine getirirler ve bunu yaparken de onun bilgisini kullanırlar. Bu nedenle öncelikli olarak bilgi nedir ve uzman kime denir sorularının cevaplanması gereklidir.

Bilgi, bir alanin veya konunun teorik yada pratik olarak anlasilmasidir. Diger bir ifade ile o alan yada konu hakkindaki bilinenlerin toplamidir (Negnevitsky, 2005). Bilgi güçtür ve o bilgiye sahip olanlar “**uzman**” diye adlandirilir. Farkli bir tanim olarak uzman, bir alan hakkında detayli bilgi ve tecrübeye sahip kisidir. Bir alan uzmanı o alan ile ilgili genis ve derin bilgi birikimi ile pratik deneyime sahiptir. Bir konuda uzman olan bir kisi diger bir konuda ayni uzmanligi gösteremez. Alanında uzman olan kisi, diger insanlardan daha becerikli olarak o alan içindeki isleri en hizli sekilde gerçekleştirir.

II.3.1.2 Uzmanlar Nasıl Düşünür ?

Insanın zihinsel süreci oldukça karmasik bir yapidir. Bu yapıyı bir algoritma ile göstermek zordur. Ancak, uzmanlar kendi alanları içerisindeki sorunları çözerken yaklaşımlarını belirli bir yapı içerisinde ifade edebilirler. Bu yapı genellikle kural tabanlı bir yaklaşımdır. Bunu bir örnek üzerinde açıklamak gerekli ise (Negnevitsky, 2005) :

Bir çocuk bir caddede karsidan karsiya geçmek istiyor. Sizde bu konunun uzmanisiniz. Bu konu üzerinde yillardir çalisiyorsunuz ve bu konuyu o çocuga öğretebilme yetenegine sahipsınız. Bunu nasıl yaparsınız ?

Çocuga trafik lambasının yesil olduğunda karsiya güvenle nasıl geçebileceğini ve isik kirmizi olduğunda durması gerektiğini açıklarsınız. Uzmanın bu bilgisi su sekilde formüle edilebilir:

*EGER “trafik isigi” yesil
ISE ‘karsiya geç’*

*IF the “traffic light” is green
THEN action is ‘go’*

*EGER “trafik isigi” kirmizi
ISE ‘dur’*

*IF the “traffic light” is red
THEN action is ‘stop’*

Bilginin bu sekilde EGER-ISE (IF-THE) kuralları ifade edilmesine “üretim kuralı” yada kısaca “kural” denir (Negnevitsky, 2005). Kurallar yapay zeka içerisinde en temel bilgi gösterim yapılarıdır. Üretilen kurallar alan içerisinde yer alan problemlerin alan uzmanları tarafından nasıl çözümlendiğini gösterir.

II.3.1.3 Kuralların Özellikleri

Kurallar sart (Eger) ve eylem (Ise) olmak üzere iki bölümden oluşmaktadır.

Kuralın söz dizimi:

Eger <sart>

Ise <eylem>

Kurallarda bir sartın gerçekleşmesi durumunda birden fazla eylem gerçekleştirilebilir yada o eyleme bağlı diğer kurallar işletilebilir. Aynı zamanda Eger kısmi birden fazla sattan oluşabilir. Bu durumdan şartlar birbirlerine VE (AND) yada VEYA (OR) bağlaçları ile bağlanırlar. “VE” bağlacı ile bağlanan şartta, eylemin gerçekleşmesi için ikisinin de yada tümünün şartının sağlanması gereklidir. “OR” bağlacında ise eylemin gerçekleşmesi için şartlardan birinin gerçekleşmesi yeterlidir. Örneğin:

Eger <sart1> VE <sart2>

Ise <eylem1>

Burada <eylem1>’in gerçekleşmesi için <sart1> ve <sart2> nin sağlanması gereklidir.

Eger <sart1> VEYA <sart2> VEYA<sart3>

Ise <eylem2>

Bu durumda ise <sart1>, <sart2> yada <sart3> den birinin gerçekleşmesi durumunda <eylem2> gerçekleşecektir.

Kurallar uzman sistemin durumuna göre ilişkileri, önerileri, direktifleri veya stratejileri gösterir (Negnevitsky, 2005, Durkin,1994). Örneğin

Direktif

EGER “trafik isigi” kırmızı

ISE ‘dur’

Öneri

EGER mevsim kış

VE hava rüzgarlı

ISE dışarı çıkmadan kabanını giy

İlişki

EGER mevsim kış

VE yerler ıslak

ISE yağmur yağmıstır

Strateji

*EGER motor çalışmıyor
ISE yakiti kontrol ediniz ;
Ilk kontrol sonu*

*EGER Ilk kontrol sonu
VE yakit deposu dolu
ISE aküyü kontrol ediniz ;
Ikinci kontrol sonu*

II.3.2 Kural Tabanlı Uzman Sistem Yapısı

Uzmanların kendilerine ait olan bu bilgi ve deneyimleri bilgisayar yazılımları tarafından anlaşılabılır olması ve daha sonraki gereksinimlerde kullanılabilmesi için yapısal bir şekilde bilgisayara girilmesi ve saklanması gerekmektedir. Uzman sistemler, saklanan bu bilgileri kullanarak insan karar verme sürecine benzer bir süreç ile problemlere çözümler üretirler. İnsanlar bir konu hakkındaki bilgilerini ve tecrübelerini uzun süreli belleklerinde saklarlar. O konu ile ilgili bir sorunla uğrassırken karşılaştıkları bir durumu ise o an için kısa süreli belleklerine alırlar. Daha sonra bu iki bellekteki bilgiyi kullanarak problemin çözümü hakkında bir çıkarım gerçekleştirirler. Böyle uğrassılan alandaki sorun ile ilgili bir sonuca ulaşılmış olur. Bu yapı uzman sistemlerinde temel yapısını oluşturmada örnek olmuştur. Kural tabanlı bir uzman sistem, Bilgi tabanı (knowledge base), Çıkarım mekanizması (inference engine) ve Kullanıcı Arayüz Birimi (user interface) olmak üzere üç temel modülden oluşur.

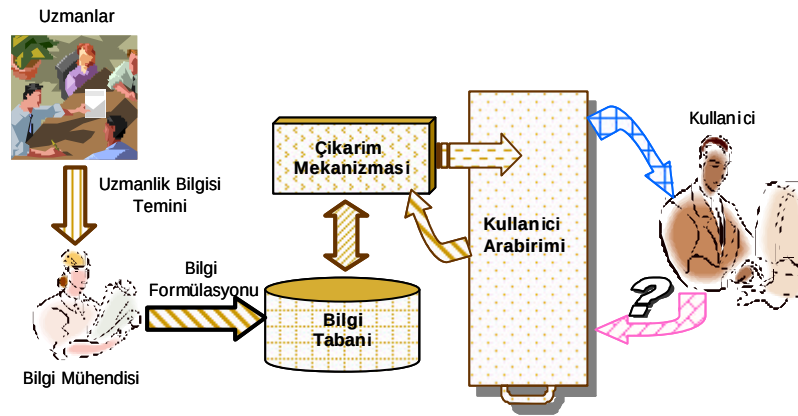
Bilgi Tabanı , problem çözmede kullanılacak olan alan ile ilgili bilgiyi içerir. Kural tabanlı uzman sistemlerde bilgi kuralları ile temsil edilir (Negnevitsky, 2005). Kural özellikleri yukarıdaki bölümde anlatılmıştır.

Çıkarım Mekanizması, bir problem üzerinde çalışılırken bilgi tabanındaki kuralları kullanarak o andaki durum ile ilgili bir sonuca ulaşılmasını sağlayan modüldür. Bu modül o andaki durum ile bilgi tabanındaki kuralları birbirine bağlar (Negnevitsky, 2005).

Kullanıcı Arayüz Birimi, kullanıcı ile uzman sistem yazılımının etkileşiminin sağlandığı bölümdür. Bu iletişim için kullanılan ekranlar mümkün olduğunca

kullanıcı dostu olmalıdır. Gereğinden az yada gereğinden fazla bilgi ekranda gösterilmemelidir.

Bu modüller uzman sistem içerisinde mutlaka bulunması gerekli olanlardır. Bunların yanında yardımcı modüllerde uzman sistemler içerisinde bulunabilir. Bunlar Bilgi Tabanına bilgi girişini sağlayan Bilgi Tabanı Editörü, yada uzman sistemin farklı programlar ve veri tabanları ile iletişimi sağlayabilecek olan dış arayüz programlarıdır. Bir uzman sistemin elemanları ve süreci Şekil II-23’de görülmektedir.



Şekil II- 24. Uzman sistem elemanları.

II.3.3 Kural Tabanlı Uzman Sistem Süreci

Kural tabanlı uzman sistem süreci içerisinde dört temel adım vardır:

a) **Bilginin temin edilmesi:** Uzman sistemin uzmanlık alanı ile ilgili bilgilerin toplanması, derlenmesi ve bilgisayarın anlayacağı şekle dönüştürülmesi çalışmalarını kapsar.

b) **Bilgi tabanının oluşturulması:** Uzman sisteminin uzmanlık alanı ile ilgili toplanan bilgilerin, bilgisayar ortamında saklanacağı veri tabanına aktarılmasını kapsar. Bu bilgiler veri tabanında genellikle Eger..İse (If...Then) kuralları veya bilgi sınıfları şeklinde olurlar. Veri tabanındaki bu bilgiler ilgili uzmanlık alanı hakkında uzmanların bildiği ve belirlediği gerçeklere dayanmaktadır.

c) **Çıkarım mekanizmasının oluşturulması :** Bilgi tabanında bulunan bilgileri arayan, filtreleyen, yorumlayan ve sonuçlar çıkaran yani; çözüm üreten bir mekanizmanın geliştirilmesini kapsar. Bu yapıda kullanılan üç tür çıkarım mekanizması vardır:

i) **Ileri dogru zincirleme**:İlgili problem hakkındaki gerçeklerden hareket edilerek sonuca gidilir.

ii) **Geriye dogru zincirleme**: Bir sonuç alınarak geriye dogru, “o sonucu destekleyen gerçekler var midir?” sorusunun cevabi aranir.

ii) **Hibrit zincirleme**: Geriye dogru zincirlemede ile Ileriye Dogru zincirleme tekniklerinin bir arada kullanilmasi ile olusturan bir yapidir

d) **Kullanici arabiriminin olusturulmasi** : Gelistirilen uzman sistemini kullanan kisiler ile uzman sistemin iletisimini saglar. Problemlere üretilen sonuçların nasıl üretildiği ve niçin o sonuçlara ulasildigini açıklar. Uzman sistemin bir uzman gibi görünmesi bu ara birimi ve açıklama yeteneginin güçlü olmasına baglidir (Öztemel, 2003).

II.3.4 Çıkarım Mekanizmasi Teknikleri

Daha önce belirtildiği gibi Çıkarım Mekanizmasi, bilgi tabanında bulunan bilgileri arayan, filtreleyen, yorumlayan ve sonuçlar çıkaran yani; çözüm üreten bir mekanizmadir. Çıkarım mekanizmasi çözüm üretirken, o anki durum ile bilgi tabanındaki her bir kurali karsilastirir. Eger duruma ait uygun bir kural bulunmus ise o kural içerisinde yer alan eylem gerçekleştirilir. O anki durum ile kuralların *Eger* kısımlarının karsilastirilmesine **çıkarım zinciri** denir (Negnevitsky, 2005). Çıkarım zinciri, uzman sistemin sonuca ulasmak için kurallari nasıl kullandigini gösteren bir yapidir. Bu yapıda ileri dogru zincirleme (forward chaining), geri dogru zincirleme (backward chaining) ve hibrit (karma) stratejiler (Hybrid strategies) olmak üzere, kullanılan üç tür çıkarım mekanizmasi vardır.

II.3.4.1 Ileri Dogru Zincirleme

Bu zincirleme türünde, ilgili problem hakkındaki gerçeklerden hareket edilerek sonuca gidilir. Ileri dogru zincirleme bilinen bir veriden baslar ve ileri dogru diger verileri üretir. Her seferinde en üsteki kural isletilir. Her bir kural bir sefer isletilir. Örneğin :

Kurallarımız:

1) *EGER* $b=dogru$ VE $m=dogru$

ISE $s=dogru$

2) *EGER* $k=dogru$ VE $b=dogru$

ISE $h=dogru$

- 3) *EGER i=dogru*
 ISE j=dogru
- 4) *EGER j=dogru*
 ISE f=dogru VE b=dogru
- 5) *EGER f=dogru*
 ISE c=dogru VE k=dogru

Bu kuralların bulunduğu bir veri tabanında ileri doğru zincirlemeyi çalıştiralım.

i=dogru durumu gerçekleştiğinde *h*'nin durumu ne olacaktır. İlk anda 3 numaralı kuralın *ISE* kısmi aktif olacaktır. Bu durumda *j=dogru* olacaktır.

j=dogru olduğunda 4 numaralı kuralın *ISE* kısmi aktif olacaktır. Bu durumda *f=dogru* VE *b=dogru* olacaktır.

f=dogru olduğunda 5 numaralı kuralın *ISE* kısmi aktif olacaktır. Bu durumda *c=dogru* VE *k=dogru* olacaktır.

c=dogru ve *k=dogru* olması kural tabanında yer alan diğer kuralları tetikleyecektir. Eger kurallar için gerekli şartlar sağlanmış ise. İşte 2 numaralı kuralda *k=dogru* olması ve daha önceden de *b=dogru* olması ile aktif olacaktır. Böylece *i=dogru* ile başlayan zincirin son halkası olarak 2 numaralı kural çalışacaktır. Böylece *i=dogru* olduğu bir durumda *h=dogru* olacaktır.

II.3.4.2 Geriye Doğru Zincirleme:

Geriye doğru zincirlemede, bir sonuç alınarak geriye doğru, “o sonucu destekleyen gerçekler var midir?” sorusunun cevabı aranır. Geriye doğru zincirleme amaç sürümlüdür. Yani uzman sistem bir amaca sahiptir. Çıkarım mekanizması bu amacı destekleyen deliller arar. Öncelikle aranan sonuçla ilgili bilgi tabanındaki kurallar aranır. Bu kuralların “*İse*” kısmında o kurallar ait amaçlar bulunur. Böylece ana amaca ait alt bir amaca ulaşılmış olur. Bundan sonra çıkarım mekanizması yeni ulaşılan bu alt amaç ile ilgili aramaya devam eder. Bu işlem ulaşılan alt amaçlara ait hiçbir kural bulunamayınca kadar devam eder.

II.3.4.3 Hibrit (Karma) Stratejiler:

Geriye doğru zincirlemede ile İleriye Doğru zincirleme tekniklerinin bir arada kullanılması ile oluşturan bir yapıdır.

Yapay zeka’da bu üç çıkarım mekanizması tekniği yanında kuralların araştırılması ve kullanımı ile ilgili değişik stratejiler de mevcuttur. Bu stratejilerin

detaylari ile ilgili bilgiler Russell ve Norvig. 1995 ve Luger, ve Stubblefield 1989 çalismalarından bulunabilir.

II.4 Zeki Kavramsal Modelleme'nin Üstünlüğü

Teknolojideki gelismeler, günümüz problemlerini ve bu problemlere ait çözümleri oldukça karmasik hale getirmiştir. Problemler içerisindeki aktör sayilari artmis, sistemler birden fazla alt sistemden, her alt sistem de kendine ait aktör, durum, ve olaydan oluşur hale gelmiştir. Tüm bunlar arasi iliskiler ise çok daha fazla kurala ve kisitta bağlanmış, aynı zamanda da karmasiklaşmıştır. Problemleri tüm bu yönleri ile görmek ve problemlerin çözümünü tam olarak ortaya koymak ise geniş uzmanlik bilgisi istemektedir. Bunlardan dolayidir ki, bu tür problemlerin çözümünde ya da sistemlerin geliştirilmesinde uzmana (insana) bağlı hatalarin ortaya çıkma ihtimali artmıştır. Yapay zeka bilimi, bu bilim içerisinde de özellikle uzman sistemler, bu tür hatari ve uzmanlara bağımliligi azaltmak için kullanılan bir yapıdır. Yapay zeka günümüz problemlerine sunmuş olduğu çözümler ile is hayatatimizdan günlük yasantimiz içerisine bir çok alanda karsimiza çıkmaktadır. Rich ve Knight 1991'de yapay zekayi, insanların günümüzde daha iyi yaptıkları isleri bilgisayarlara nasıl yaptırılabilir çalismalari olarak tanımlar. Yapay zekada, insanların bir olay karsisinde karar vermeye ve olaya çözüm getirmeye çalışırken nasıl bir düşünce yolu izledikleri taklit edilir. Düşünce islemi basit adımlara bölünür ve bu adımlari kullanarak problemi çözen bir bilgisayar programi düzenlenir. Yapay zeka teknikleri ile, bu sayede, karar verme mekanizmasi karisik olan problemleri çözen programların basit ve yapısal bir şekilde tasarlanması sağlanır.

Kavramsal model kullanmak ile sistem geliştirmelerde hata oranı düşmekte, sistemler daha kısa sürede hızlı bir şekilde tasarlanıp, gerçekleştirilmeye geçilebilmekte, geliştirilen sistemlerin doğrulanması tam olarak yapılabilmekte, müşteri sonuçta istediği ürüne tam olarak kavusabilmektedir. Kavramsal model geliştirmeye, yapay zekanın destek vermesi ile sistem geliştirmede, hata oranı daha da azalması, daha güvenilir ve doğru sistemlerin geliştirilmesi sağlanabilmektedir. Bu nedenle bu çalışmada kural tabanlı bir uzman sistem tanımlanmıştır.

II.5 KM Teknikleri ve Mevcut Çalışmaların İrdelenmesi

Geleneksel ve gelişmiş kavramsal modelleme teknikleri incelendiğinde aşağıdaki benzerliklere ulaşılmaktadır:

- Tüm kavramsal modelleme tekniklerinin ontolojisinde (gösterim tip kütüphanesi) ilişki, eleman, gibi sınıf kavramları yer almaktadır (Dieste ve dgr., 2001).
- Kavramsal modellerde ihtiyaç duyulan aktör, amaç, kural gibi kavramlar ontolojinin genişletilmesi ile gerçekleştirilmektedir. (Hayy, 1999)
- Ontoloji genişletilmesinin temelinde problem alanını tam ifade etmek yatmaktadır. (Hayy,1999, Dieste ve dgr. 2000)
- Tek bir kavramsal model problem alanı tam olarak ifade edilememektedir (Hayy,1999, Dieste ve dgr. 2000)

Sistem geliştirmede, başarılı ve sorunsuz bir sisteme ulaşmak için, basta amaçlanan sistemin iyi analiz edilip, problem alanının tüm ayrıntılarını belirlemek gerekliliği vardır. Ancak mevcut kavramsal modelleme teknikleri ile problem alanı yukarıdaki maddelerde bahsedilen nedenlerden dolayı objektif bir şekilde tüm yönleri ile tek bir kavramsal modelle gösterilememektedir.

Bunun yanında kavramsal model geliştirme için kullanılan tekniklerin çoğu “problem yönelimli” olmaktan çok, icrâ edilecek olan hedef yönelimlidir (Dieste ve dgr. 2000). Yani problemin anlaşılmasından daha çok, problemin nasıl çözüme ulaştırılacağı, yada çözümün yapılacağı ortamda bu işin nasıl yapılacağı ifade edilmektedir. Diğer bir ifade ile, mevcut kavramsal model geliştirme teknikleri “*problemin ne olduğundan*” çok “*problemin nasıl çözümleneceğini*” içermektedir.

Buna ilaveten, kavramsal model teknikleri içerisinde kullanılan kavramsal modellerin çoğu gösterim notasyonları olarak nesne yönelimli analiz (OOA) tekniklerinin gösterim sembolojilerini kullanmaktadırlar. Bu da, kullanılan nesne yönelimli analiz gösterim sembolojileri “*problem yönelimli*” olmaktan çok “*hedef yönelimli*” olmasından dolayı, kavramsal modele kısıtlama getirmektedir (Dieste ve dgr. 2000, Jackson, 1995). Örneğin, veri akışı diyagramları sorunun çözümü ile ilgili

bir çok gösterim sekline sahip olmasına ragmen, problemin ne olduguna dair bir gösterim yaklasimi içermemektedir (Jackson, 1998).

Bir önceki bölümde anlatılan TCM yazilimi, sadece yazilim sistemlerinde kullanılan bir kavramsal model gelistirme aracidir. Bu yazilimin zekilik özelligi yoktur. Yazilim sistemlerinin gelistirilmesinde kullanılan ER, Veri ve Olay akis diyagramlari, durum geçis diyagramlari ve UML diyagramlarini desteklemektedir.

ICOM ise, bu çalışma kapsamında gelistirilen yazilim ile isim benzerligine sahiptir. Her iki yazilimda da yapay zeka destegi bulunmaktadır. Ancak, ICOM sadece veri tabani modellemesinde ER diyagramlarini desteklemektedir. Bu diyagramlar çizilirken ortaya çıkan mantiksal hatalarda kullanıcıyi bilgilendirmektedir. ICOM'un UML destegi için çalışmalar devam etmektedir.

Tüm bu incelemelerin sonucunda:

Mevcut kavramsal modelleme tekniklerinin sahip olduklari kisitlari ortadan kaldırilmasi ve uygulama alanindan bagimsiz olarak sistemlerin basarili bir sekilde gelistirilebilmesi için **“disiplinler arasi bir kavramsal model gelistirme yaklasimina”** ihtiyaç vardır. Bu teknik asagidaki kriterleri saglamalidir:

- Öncelik kullanıcı ihtiyaçlarinin dogru bir sekilde anlasilmasidir. Bu nedenle, sistemin gelistirim yaklasimi düşünülmeden, kullanıcı ihtiyaçlarinin anlasilmasi saglanmalidir. Yani, problemin nasıl çözümlenmesi düşünülmeden, problemin ne oldugunun anlasilmasina odaklanilmalidir.
- Problemin yada kullanıcı ihtiyaçlarinin anlasilmasi, seçilen problem çözme yada yazilim gelistirme yaklasimindan bagimsiz olmalidir.

Disiplinler arasi bir yaklasimin gelistirilerek, bu kriterlerin saglanmasi da ancak **“KM gelistirim sürecinin güncellenmesi”** ile gerçekleştirilebilir. Bu çalışma kapsamında öncelikle bu gerçekleştirilmistir. Daha sonra güncellenen süreç temel alınarak disiplinler bir kavramsal model yazilimi gelistirilmistir. Bu yazilima “In-CM Kavramsal Model Gelistirme Yazilimi” adi verilmistir. Kisaca In-CM ile bu yazilim ifade edilmektedir.

BÖLÜM III

KAVRAMSAL MODEL GELİSTİRME SÜRECİ

III.1 Bilinen Kavramsal Modelleme Süreci

Dehne ve dgr. 2000’de yapılan çalışmada kavramsal modelleme “bir sistemin davranisini ya da ayristirilmasini gösteren yapidir” seklinde tanimlanmistir. Literatür çalışmasında ve kavramsal modelleme tekniklerinin incelenmesinde görüldü ki, sistemlerin davranislarini tek bir kavramsal modelleme teknigi ile gösterilmesi oldukça zordur. Böyle bir kavramsal modelin ortaya konulmasi için genis bir gösterim kütüphanesine sahip bir modelleme teknigini gerektirmektedir. Bu özelliklere sahip bir kavramsal modelleme teknigi hazirlansa dahi bu teknikle çizilen kavramsal modeller oldukça karmasik bir yapıda olacaktır. Çünkü bu tür bir kavramsal model içerisinde verilerin durumu, elemanların özellikleri ve ilişkileri, olayların zaman sirasi, aktörler ve görevleri gösterilecektir. Oysaki, yine Bölüm II’de kavramsal modellerin kullanım amaçlarında belirtilen ortak anlasilirligin saglanmasi, kullanıcı ve analist açısından kavramsal modelin grafiksel gösteriminin basitligi ile saglanabilir. Çünkü kullanıcı teknik terimlerden ve bu yöndeki karmasiklikten uzakta bir bilgiye sahiptir. Oysaki sistemin tüm yönleri ile tek bir kavramsal model üzerinde gösterilmesi oldukça genis ve karmasik bir model çizilmesine sebep olacaktır.

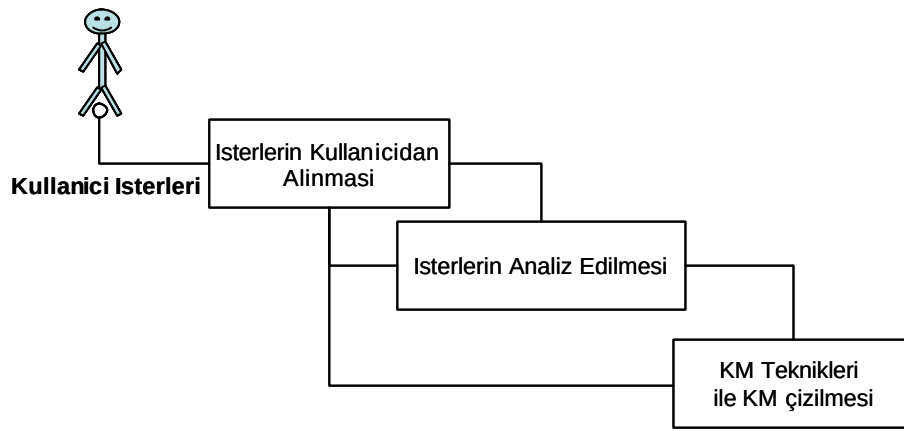
Bilinen kavramsal modelleme sürecinde kavramsal modelin geliştirilmesi müşteri (kullanıcı) isteklerinin belirlenmesi ile baslar. Bu istekler üzerine bir tasarım ve bu tasarım temel alınarak sistemin gerçekleştirilmesi saglanır. Daha sonra sistem gerekli doğrulama ve test işlemlerinden geçirilerek müşteriye teslim edilir. Ancak bu süreç, kavramsal modelin geliştirilmesinde problemin çözüm yöntemi ve gösterimine odaklanmış bir yapıdadır. Bu nedenle problemin ne olduğunun anlasilmasi ile ilgili bir tanımlama yoktur.

Bunun yanında mevcut süreç içinde problemin anlasilmasi için sistem veya problemin elemanları ile bunların ilişkilerinin detaylarını açık olarak irdelemeyi gösterememektedir. Mevcut süreç içerisinde eleman ilişki modelleri kullanilsada bu modeller sistemin ya da problemin çözümü sırasındaki elemanları ve niteliklerini

gösterir. Mevcut süreç Şekil III.1’de gösterilmiştir. Bu iki durum kavramsal modellemenin temel amaçlarından olan

- Sistemin veya problemin elemanlarını, verilerini, durumlarını ve ilişkilerini göstermek,
- Bir sistemin analiz edilirken anlaşılabilirliğini arttırmak,

ters düşmektedir. Kavramsal modellemenin bu amaçlarını da gerçekleştirecek, bahsedilen sorunları çözebilecek nitelikte sürecin güncellenmesine ihtiyaç vardır. Bu çalışmada bu ihtiyacı giderecek güncellemeler yapılmış aşağıda detaylı olarak güncellenmiş süreç açıklanmıştır.



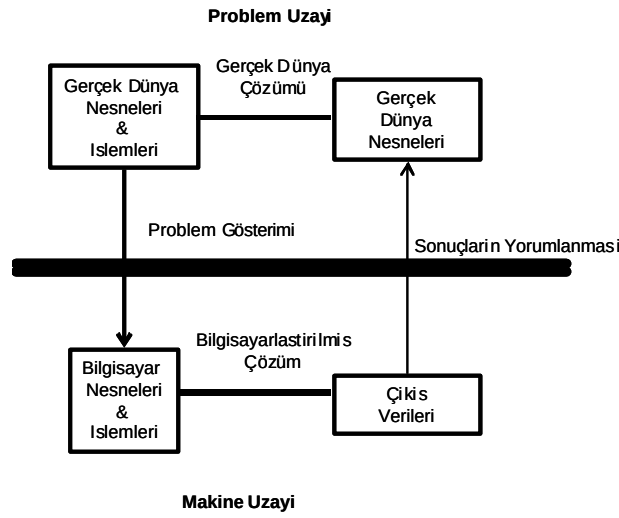
Şekil III- 1. Bilinen Kavramsal Modelleme Süreci.

III.2 In-CM Kavramsal Model Gelistirme Süreci

Bu tezin amaçlarından biri de “*disiplinler arası ortak kavramsal model yapısının*” kullanılması ise yine geniş bir gösterim kütüphanesini gerektirmektedir. Çünkü her disiplin kendine özgü yapı ve gereksinimlere sahiptir. Bu yapı ve gereksinimlerin kavramsal model içerisinde gösterilmesini gerekecektir. Bu nedenle “*sistem davranışını*” disiplinler arası tek bir kavramsal model ile gösterilmesi mümkün olarak gözükse dahi, kavramsal modeli kullanım amacından uzaklaştıracaktır.

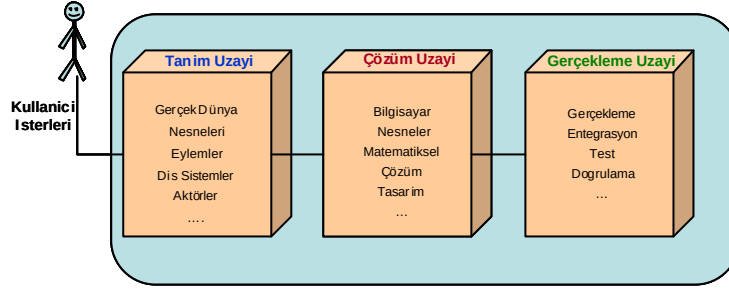
Bu çalışma kapsamında hedeflenen amaca ulaşmak için kavramsal model tanımında yer alan “*sistemin ayrıştırılması*” üzerine çalışılmıştır. Böylece, kural tabanlı uzman sistemlerin kullanıldığı, disiplinler arası model geliştirebilen ve anlaşılabilirliği yüksek olan bir sistem geliştirilmiştir. Bu sistem kısaca In-CM olarak

isimlendirilmiştir. Dieste ve arkadaşları 2000’de gerçekleştirilen teorik bir çalışmada, Şekil III-2’deki yaklaşım önerilmiştir. Bu yaklaşıma göre problem çözüm uzayı, problem uzayı ve makine uzayı olmak üzere iki alana ayrılmıştır. Problem Uzayı’nda gerçek dünya nesneleri üzerinde çalışılır. Burada çözülmesi gereken problem için kavramsal çözüm gerçekleştirilir. Daha sonra, Makine Uzayında, sayısallaştırılmış bilgisayar nesneleri üzerinde çalışılır. Yani, problem için amaçlanan çözüm gerçekleştirilir. Problem uzayında elde edilen kavramsal çözüm ile makine uzayından elde edilen çözüm karşılaştırılarak problem için amaçlanan çözüme ulaşıp ulaşılmadığı yorumlanır. Gerekirse düzeltme işlemleri gerçekleştirilerek ideal çözüme ulaşmaya çalışılır.



Sekil III- 2. “Problem” uzayına karşılık “Makine” uzayı (Dieste ve dgr.,2000).

Sistemler arası kavramsal model geliştirme süreci tanımlanması için ise, bu tez çalışması kapsamında, Dieste ve diğerlerinin gerçekleştirmiş oldukları bu teorik yaklaşım temel alınarak Şekil III-3’de görülen problem çözüm uzayı tanımlanmıştır. Bu yaklaşıma göre, problem: tanım, çözüm ve gerçekleştirme uzayı olmak üzere üç alan içerisinde düşünülür.



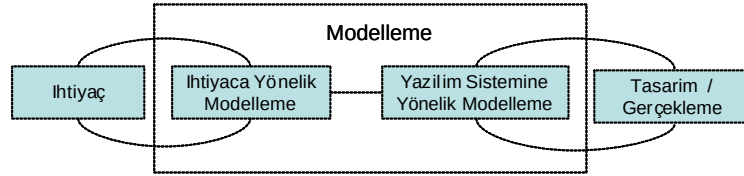
Sekil III- 3.In-CM Problem çözüm uzaı.

Bu uzaılar içerisinde :

- *Tanım uzaı*nda çalışılırken, amaçlanan sadece problemin anlaşılmasıdır. Bu nedenle odakta sadece kullanıcı istekleri vardır. Bu ihtiyaçların nasıl çözüme kavuşturulacağına dair herhangi bir yapı üzerinde düşünülmez. Sadece problemin ne olduğu, hangi alt problemleri kapsadığı, problem içerisinde genel amaç ve alt amaçların ne olduğu, bu amaçlara ait problem içerisinde hangi eylemlerin gerçekleştiği ve hangi dis sistem ile aktörlerin yer aldığı anlaşılmasına odaklanılır.
- *Çözüm uzaı* tarafında ise, tanımlanmış gerçekleştirilen problemin ne şekilde çözümleneceği odak noktası olarak alınır. İhtiyaçların anlaşılmasında, gerçekleşen eylemlere bu odak noktasından bakılarak bu probleme uygun olan çözüm yapısı seçilir.
- *Gerçekleme uzaı*nda ise tanımlanmış gerçekleştirilen ve çözüm yapısı tespit edilen problem son olarak tespit edilen yapıya uygun bir şekilde gerçekleşir. Bu uzaı içerisinde seçilen çözüm yapısına uygun programlama dili ve donanımı seçilerek problem çözümü bilgisayarlaştırılır. Daha sonra geliştirilen sistem üzerinde test ve doğrulama işlemleri gerçekleştirilerek nihai hedefe ulaşılır.

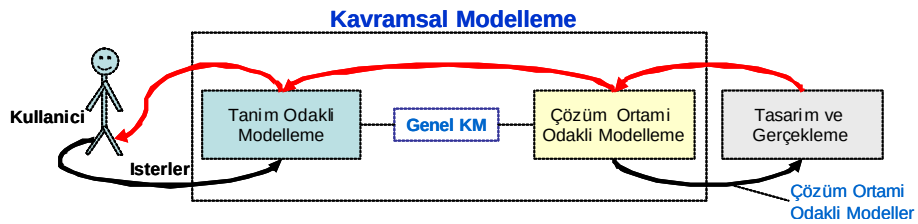
Bu şekilde üç katmanlı bir yaklaşımın avantajı, kesin olarak “tanım, çözüm ve gerçekleştirme”, gereksinim ve eylemlerinin birbirinden bağımsız olarak düşünülmesidir. Bu yaklaşımın amacı, geliştirilecek olan sistemlerin son hali göz önüne alınmadan sisteme ait “kavramsal modelin” gerçekleştirilmesi ve bu yapılırken de çözüm uzaına ait tasarım ile ilgili kavramların geliştirilecek olan “kavramsal modeli” etkilememesidir.

Yine Dieste ve arkadaşları 2000’de gerçekleştirilen teorik çalışmada, yazılım sistemlerinde tasarım ve uygulama öncesinde gerçekleştirilen modelleme eylemlerini iki adım olarak Sekil III-4’deki gibi önermişlerdir.



Sekil III- 4. Yazılım Gelistirim Sürecinde Modelleme (Dieste ve dgr. 2000).

Bu yaklaşımdan hareket ile, disiplinler arası bir kavramsal modelleme gerçekleştirilebilmesi için bu çalışmada Sekil III-5’deki yaklaşım geliştirilmiştir. Buradaki kavramsal modelleme eylemleri yukarıda anlatılan “*Tanım Uzayı*” içerisinde yer almaktadır.

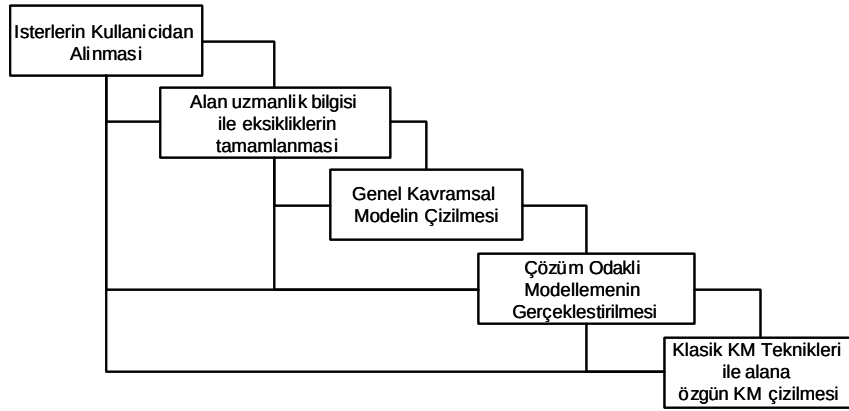


Sekil III- 5. In-CM Sistem Gelistirim Sürecinde Modelleme.

Bu yapıda, kavramsal model, “tanım odaklı modelleme” ve “çözüm ortamı odaklı modelleme” olmak üzere iki adımdan oluşmaktadır. Bu süreçte, “Tanım Odaklı Modelleme (TOM)”, direkt olarak kullanıcı ihtiyaçları ile ilgilidir. Bu adımda, çözümde kullanılacak olan problem çözme yada sistem geliştirme yaklaşımı düşünülmez. Bu adimin çıktısı “Genel Kavramsal Model”dir.

“Çözüm Ortamı Odaklı Modelleme (ÇOM)” ‘de sistem mühendisleri, tespit ettikleri ihtiyaçları karşılamada ve istenilen sonuca ulaşmada en uygun olacak olan yapısal, nesne yönelimli, bilgi tabanlı veya disiplin içerisinde yer alan ya da ileride geliştirilecek olan herhangi bir “sistem geliştirme yaklaşımını” belirlerler. Bu adimin çıktısı, belirlenen çözüm yaklaşımına göre klasik kavramsal modelleme teknikleri kullanılarak oluşturulan yapısal, nesne yönelimli, bilgi tabanlı yada gerçek zamanlı modellerdir. Tasarım ve Gerçeklemede ise kavramsal modeli belirlenmiş ve tasarımı yapılmış olan sistem geliştirilir, test edilir ve doğrulanır.

Sistemler arası kullanılacak olan genel bir kavramsal model geliştirilmesi için, problem çözüm uzayının bu şekilde tanımlanmasından sonra, Şekil III-6'da görülen In-CM Kavramsal Model geliştirme süreci tanımlandı. Bu sürecin ilk adımı kullanıcı isteklerinin yapısal bir şekilde kullanıcılardan alınmasıdır. Daha sonra bu istekler alan uzmanlık bilgisi kullanılarak tamamlanır ve sistem ayrıştırılması gerçekleştirilir. Bu şekilde tanım odaklı modellemenin çıktısı olan “Genel Kavramsal Model” geliştirilir ve grafiksel olarak çizilir. Bu model kullanıcı için basit ve anlaşılır yapıdadır. Daha sonra “Genel Kavramsal Model” kullanılarak çözüm odaklı modelleme gerçekleştirilir. Bu adımda sistemin çözüm yapısına uygun geleneksel kavramsal modelleme tekniklerine karar verilir. Karar verilen geleneksel kavramsal modelleme tekniklerinden biri ya da bir kaç kullanılarak sisteme ait kavramsal modeller geliştirilir.



Şekil III- 6. In-CM Kavramsal Model Geliştirim Süreci.

Tanımlanan bu süreç Bölüm II’de anlatılan ‘*Değişik uygulama alanlarında kullanılmasına rağmen kavramsal model geliştirme yaklaşımları genelde benzerlik gösterirler. Kavramsal model geliştirme temel olarak üç adımı sahiptir:*

- *Dogal dil ile ifade edilen isteklerinin toplanması,*
- *Alan uzmanlık bilgisi ile eksik ya da yanlış noktaların tamamlanması,*
- *Kavramsal model gösterim tekniklerinden biri ya da birkaçının kullanılması ile kavramsal modelin geliştirilmesidir.”*

bilgilerini aynen kapsamaktadır. Bu süreç literatür içerisinde anlatılan kavramsal modelleme sürecinin disiplinler arası hale getirilebilmesi için düzenlenmiş

halidir. Bir sonraki bölüm içerisinde bu güncellenen süreç ile kavramsal modelleme yaklaşımı ve referans kavramsal model anlatılacaktır.

BÖLÜM IV

KAVRAMSAL MODELLEME YAKLASIMI VE REFERANS MODEL

IV.1 Kavramsal Modelleme Yaklasimi

Bilinen kavramsal model geliştirme yaklasimda problemin çözümüne ait bir yaklasim var. Yani sistemin veya problemin ne olduğu üzerinde çalışılmaz. Müsterinin belirttiği isteklerin eksiklikleri tamamalanarak geleneksel kavramsal modelleme teknikleri ile kavramsal modeller oluşturulur. Bölüm II’de belirtilen mevcut tekniklerin kısıtlarından dolayı tek bir kavramsal model ile sistem ya da problem çözümlemesi gösterilemez. Budan dolayı herhangi bir sistem veya çözülecek herhangi bir problem kavramsal açıdan modellenirken, anlasilirligin arttirilmesi için su açılardan bakilmesi gerekir :

- Aktörler (eylemleri gerçeklestirenler),
- Süreçler (islemler dizisi),
- Görevler (temel amaç),
- Alt yapı (yazılım, donanım vs.)
- Amaç
- Alan ve altalanlar

Bu tür bir yaklasim ile sistemleri kavramsal olarak modellemenin en önemli faydasi bir referans model olusturulmasini mümkün kilmasidir. Gelistirilen referans model bir sonraki bölüm içerisinde detayli olarak açıklanmistir.

IV.2 Referans Model

Referans model, herhangi bir alanda o alana özgü değişik özelliklerde kavramsal modelleme yapmayı kolaylastiracak bir yaklasimi göstermektedir. En önemli özelliği herhangi bir alanda referans olabilmesidir. O nedenle Referans Kavramsal Model (RKM) olarak adlandırılmaktadır. RKM problem alaninin tamamini kapsayan kapsamlı bir meta model olarak görülebilir. Bu meta model

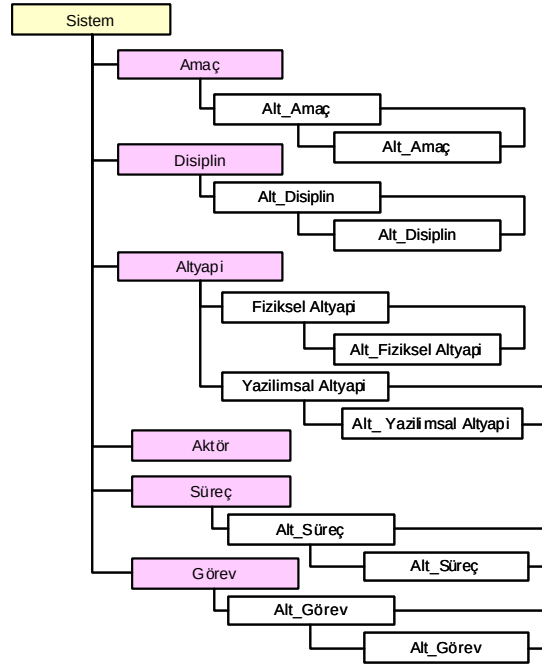
kullanarak kullanıcı kendi istek ve ihtiyacı doğrultusunda kendine özgü daha dar kapsamlı kavramsal modeller oluşturabilir.

Önerilen kavramsal model geliştirme sürecinin girdisi kullanıcının istekleridir. Ancak, kullanıcı problemini belirtirken, yani isteklerini ifade ederken, problem alanını tüm ayrıntıları ile ortaya koyamayabilir. Diğer bir ifade ile problemi tam tanımlayamaz ya da eksik istek bildirebilir. Bunun yanında kullanıcının dikkate almadığı fakat geliştirilecek olan sistem için bir girdi oluşturan dış sistem ya da aktörler de olabilir. Tüm bunların eksiksiz bir şekilde tespit edilmesi için kullanıcının probleminin analiz edilmesi ve varsa eksikliklerin de tamamlanması gereklidir. Bu tamamlama ancak problem alanına ait daha kapsamlı bir modelin varlığı ile mümkündür. Bu model bu çalışmada Referans Kavramsal Model olarak isimlendirilmiştir.

Bu çalışmada eksiklikleri tamamlanan ve RKM'den oluşturulan kullanıcı kavramsal modeline “Genel Kavramsal Model (GKM)” adı verilmiştir. GKM, TOM kapsamında gerçekleştirilir. TOM ve ÇOM Bölüm III’de detaylı şekilde anlatılmıştır. Bir sistemin, RKM yardımıyla GKM oluşturulduktan sonra, problemin ya da sistemin yapısına ve uzmanının yönlendirmesine göre “çözüm ortamı odaklı” bir yaklaşım ile bir geliştirim ortamı seçilir ve problemin kavramsal model gösterimi bu ortam içerisinde kullanılabilen klasik ya da gelişmiş kavramsal model gösterim metodlarından biri ya da bir kaç kullanılarak gerçekleştirilir. Bu amaçla gerçekleştirilen işlemler ÇOM kapsamındadır.

Referans Kavramsal model kullanıcı isteklerinin yapısal halde alınması ve eksikliklerin tamamlanması için kullanılan bir araçtır. Disiplinlere ait alan bilgisinin problem analizi için bir kalıp(pattern) haline getirilmesidir.

Bu yapıya göre tüm alanlarda yer alan “Sistem”ler **altı** yönden çözümlenirler. Şekil IV.1’de referans kavramsal modelin genel yapısı görülmektedir.



Sekil IV -1. In-CM Referans Kavramsal Modelin genel yapısı .

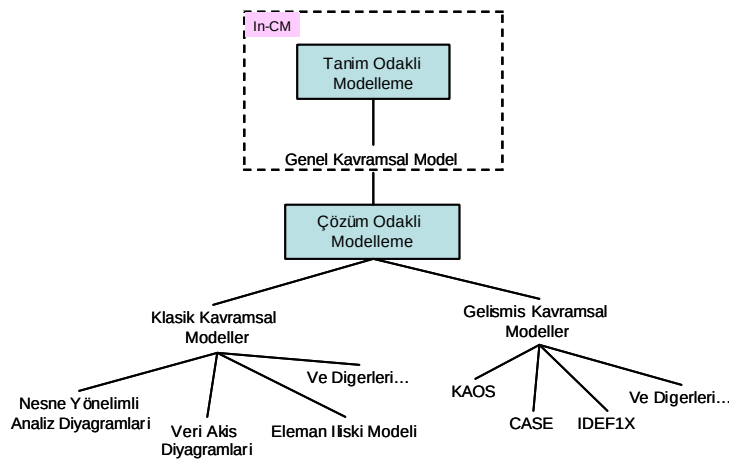
Referans kavramsal modelde yer alan adımların açıklamaları şu şekildedir:

- 1) **Amaç:** Her sistem bir amaca ve her amaçta kendine ait alt amaçlara sahiptir. Kullanıcı eğer bir alan içerisinde bir sistem geliştirmek istiyor ise o sistem ile amacının ne olduğunu belirtmelidir. Bu amaç kendine ait alt amaçlara sahip olabileceği gibi, üst amaçlara da sahip olabilir.
- 2) **Disiplin:** Sistem’ler belirli disiplinler içerisinde yer alır. Bu disiplinlerde kendilerine ait alt disiplinlere sahiptir.
- 3) **Altyapi:** Sistemler belirli alt yapılar üzerine insâ edilir. Bu alt yapıların genellemesi yapıldığında her sistem bina, makine, teçhizat gibi “fiziksel altyapi”ya ya da işletim sistemi, grafik programları, test programları gibi “yazılımsal altyapiya” ya da bunların ikisine birden sahip olmaktadır. Bu nedenle sisteme ait kullanıcı istekleri belirlenir iken bu iki başlık altında sistem altyapısı tanımlaması gerçekleştirilir.
- 4) **Aktör:** Sistemler içerisinde yer alan ve sisteme girdi veren ya da sistemden etkilenen elemanlardır.

- 5) **Süreç:** Sistem içerisinde yer alan süreç ve bu süreçlere ait alt süreçlerin tanımlamaları gerçekleştirilir.
- 6) **Görev:** Sistem süreçleri içerisinde, sistem aktörleri tarafından gerçekleştirilecek olan eylemlerdir.

Referans kavramsal model, problemin ayrıştırılmasının tam yapılması için kullanılan ve bu çalışma kapsamında önerilen bir kalıptır. Bu kalıp eğer bir yazılım içerisinde kullanılacak ise, esnek bir yapıda, yazılımı kullananlarca alt seviyede genişletilebilecek bir şekilde tanımlanmalıdır. Ayrıca, referans kavramsal modelin tanımlanma derinliği uzmandan uzmana göre değişiklik gösterir. Normalde bu gerçek hayat da böyledir. Bir alan konusunda uzman olan iki kişi birbirlerine göre farklı derinlikte bilgilere sahiptirler. Aynı zamanda aynı konunun uzmanları o konu hakkında detaylı bilgiye sahip olsalar dahi, ikisi de o konunun farklı bir alanında diğerinden daha derin bilgiye sahiptir. Bu da referans kavramsal modelin hazırlanmasında uzmandan uzmana göre farklılık göstermesine sebep olur. Bu nedenle farklı uzmanların görüşleri alınarak geliştirilen RKM daha zengin bir yapıya sahip olur.

Bu çalışmada geliştirilen In-CM Zeki Kavramsal Model Geliştirme yazılımı önerilen kavramsal model geliştirme yaklaşımının ilk adımını kapsar. Şekil IV-2’de bu durum gösterilmektedir.



Şekil IV -2. In-CM Zeki KM Geliştirme Yazılımı Kapsamı.

TOM’da amaç problemin anlasilmasinin saglanmasidir. Bunun için bir problem analizi gereklestirilmelidir. Bu safhadaki problem analizi, kullanicinin isteklerini anlama sürecidir (Dieste ve dgr. 2000). Amaci ise, isteklerin nasıl yerine getirilebileceginin ortaya konulmasidir. Bu asamada kulanici istekleri yapisal bir hale getirilir. Bir problem, “seylerin” nasıl oldugu ile kullanicinin onu nasıl istedigini arasında bir haritadir (Booker, 2003).

Amaçlanan problem analizinde ilk olarak problemin uygun bir yapıda yazılarak problem üzerinde herkesin ortak anlasilirligi saglanmalıdır. Daha sonra problem içerisinde yer alan aktörler, alt sistemler, araştırılır. Problem için uygun bir kalip (patern) bulunursa problemin çözümüne daha kolay ulasilabilir (Booker, 2003). Bu çalışmadaki yaklasim kapsamında, kullanicinin isterlerini yapisal hale getirilebilmesi için RKM ayni zamanda bir kalip olarak önerilmiştir.

Önerilen yaklasim, güncellenen süreç ve olusturulan RKM uzman sistem teknoloji ile bilgisayar ortamına tasinarak In-CM isimli yazilim kütüphanesi olusturulmustur. Bir sonraki bölümde bu kütüphane içerisinde yer alan In-CM Bilgi Tabani Editörü, Çikarim Mekanizmasi ve Grafik Editörü detayli olarak açıklanmıştır.

BÖLÜM V

IN-CM :GENEL YAPI

V.1 In-CM Zeki KM Gelistirme Yazilimi'nin Amaci

Kavramsal modelleme gelistirilirken yazilim sistemlerinden faydalanmak mümkündür. Özellikle yapay zeka teknolojisinin kavramsal model gelistirme konusunda önemli katkiları olacağı bu çalışmada öngörülmektedir. Yapay Zeka (YZ) günümüz problemlerine sunmuş olduğu çözümler ile iş hayatımızdan günlük yaşantımız içerisine bir çok alanda karşımıza çıkmaktadır. Yapay zeka çalışmaları sonucunda ortaya çıkan ürünlerin, özellikle gelişmiş ülkelerde, günlük hayatta kullanımları, insanlara bir çok fayda sağlamakta, işlemlerin maliyetlerini azaltmakta, kârliliklerde artışlar olmakta, işlem zamanları kısaltılabilmekte ve problemlere daha iyi çözümler üretilebilmektedir. Diğer yünden, kavramsal model kullanmak ile sistem geliştirmelerde hata oranı düşmekte, sistemler daha kısa sürede hızlı bir şekilde tasarlanıp, gerçekleştirilmeye geçilebilmekte, geliştirilen sistemlerin doğrulanması tam olarak yapılabilmekte, müşteri sonuçta istediği ürüne mümkün olduğunca tam olarak kavuşabilmektedir. Yapay zeka ve kavramsal modelleme sistem geliştirmede birbirine benzer faydaları sunmaktadırlar.

Kavramsal model geliştirmeye, yapay zekanın destek vermesi ile sistem geliştirmede, hata oranı daha da azaltılarak, daha güvenilir ve doğru sistemlerin geliştirilmesi sağlanabilir. Böyle bir sistemin tasarlanabilmesi için genel bir çerçeveye ve örnek uygulamalara ihtiyaç vardır.

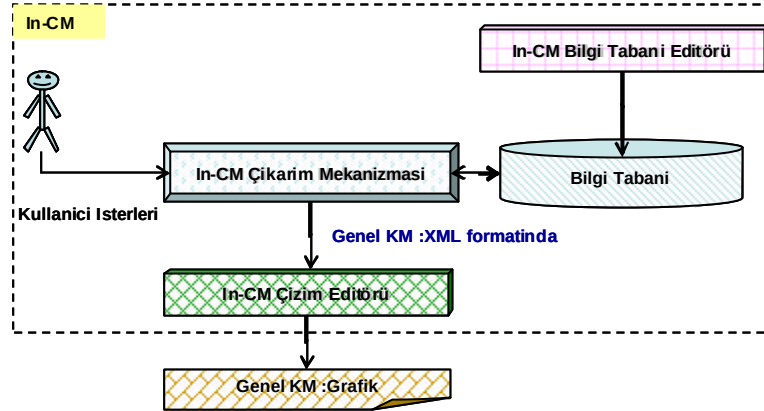
Yukardaki gerekçe ve ihtiyaçlardan dolayı bu tez çalışmasında, hem kavramsal modelleme alanında önemli sorunlara çözümler üretilmiş hem de yapay zeka tekniklerinden uzman sistemler kullanarak “Zeki Kavramsal Model Gelistirme Aracı” yazılımı geliştirilmiştir. Bu yazılımın modülleri ve özellikleri aşağıdaki bölümlerde açıklanmıştır.

V.2 In-CM Zeki KM Gelistirme Yazilimi

Bu tez çalışması kapsamında önerilen kavramsal model geliştirme süreci temel alınarak disiplinler arası bir kavramsal modelleme yazılımı hedeflenmiştir. Ayrıca bu

yazılım Bölüm 2’de anlatılan yapay zeka tekniklerinden kural tabanlı uzman sistemler ile desteklenerek sisteme zekilik yeteneği katılmıştır. Bu yazılımın uygulama alanı olarak “Eğitim” seçilmiştir.

In-CM yazılımı kural tabanlı bir uzman sistemdir. Geliştirilen programın genel yapısı Sekil V-1’de görülmektedir.



Sekil V-3. In-CM Genel Program Yapısı.

Bu yazılım üç alt yazılım ve veritabanı modülünden oluşmaktadır. Bu yazılım modülleri:

1. **In-CM Bilgi Tabanı Editörü (Knowledge Base Editor):** Bu yazılım, In-CM Çıkarım Mekanizması’nda kullanılacak olan Referans Kavramsal Model’in ve Kuralların tanımlanması veya çıkarılmasını sağlar.
2. **In-CM Çıkarım Mekanizması (Inference Engine):** Bu yazılım ile kullanıcı kavramsal model geliştirir.
3. **In-CM Grafik Editörü (Graphic Editor):** In-CM Çıkarım Mekanizması ile geliştirilen kavramsal modelin grafiksel gösterimi yapılır.

Bu üç program tüm modüllere kullanıcının kolay ulaşımı ve yazılım bütünlüğü açısından tek bir çatı altında toplanmıştır. Her bir modülün detaylı açıklaması ileri bölümlerde yapılacaktır. Programların ekran görüntüleri Ek-A’da verilmiştir.

V.3 In-CM Bilgi Tabanı

In-CM Çıkarım mekanizması yazılımında kullanılacak olan referans kavramsal model ile bu modele ilişkin alana ait kuralların depolandığı XML formatında bir veri tabanıdır. Bu yapı içerisinde üç XML dosyası bulunur. Bunlar

- Referans Kavramsal model tanımını içeren “ReferenceCM.xml”,
- Alan ait kuralları içeren “RuleBase.xml”
- Kullanıcının oluşturmaya çalıştığı kavramsal modeli saklayan “UserGoal.xml”’dir. Kullanıcı üzerinde çalıştığı kavramsal modeli farklı bir isim ile de kaydedip çalışabilir.

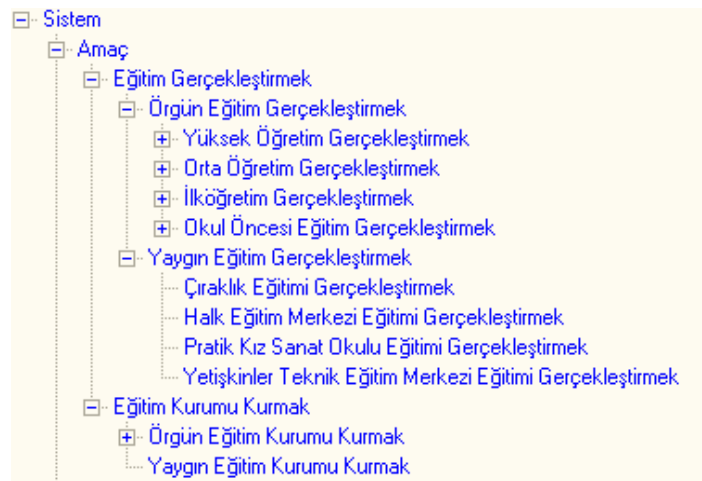
Veri tabanında bulunan bu XML dosyalarından ilk ikisi “*Bilgi Tabanı Editörü*” üzerinden oluşturulup, şekillendirilirken, kullanıcı kavramsal modelini içeren XML dosyası ise “*Çıkartım Mekanizması*” yazılımından oluşturulup değiştirilebilmektedir.

V.3.1 In-CM KM Geliştirme Sürecinde Referans KM

Referans Kavramsal model kullanıcı isteklerinin yapısal halde alınması ve eksikliklerin tamamlanması için kullanılan bir araçtır. Disiplinlere ait alan bilgisinin problem analizi için bir kalıp(pattern) haline getirilmesidir. Bu yapıya göre tüm alanlarda yer alan “Sistem”ler **altı** yönden çözümlenirler. Bu yönlerin detaylandırılması Bölüm IV içerisinde anlatılmıştır.

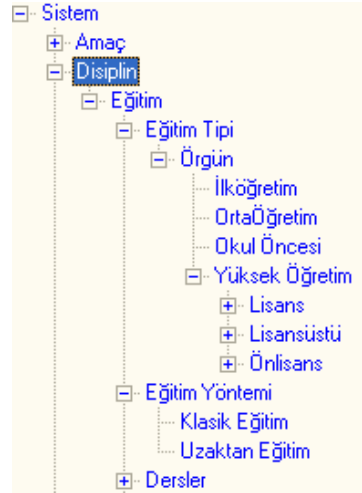
In-CM Bilgi tabanında kullanılan Referans kavramsal modelde yer alan adımların açıklamaları şu şekildedir:

- 1) **Amaç:** Her sistem bir amaca ve her amaçta kendine ait alt amaçlara sahiptir. Bu tez çalışmasına disiplin alanı olarak Eğitim seçilmiştir. Bu nedenle Şekil V-1’de bu disipline göre amaç örneklendirilmiştir.



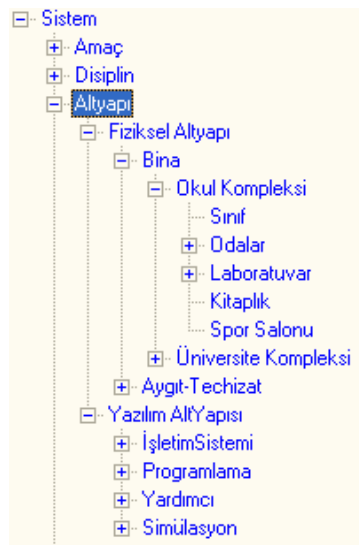
Şekil V- 1. In-CM Referans KM’de Amaç detaylandırılması örneği.

- 2) **Disiplin:** Sistem’ler belirli disiplinler içerisinde yer alır. Sekil V-2’de bu eğitim’e göre disiplin örneklendirilmiştir.



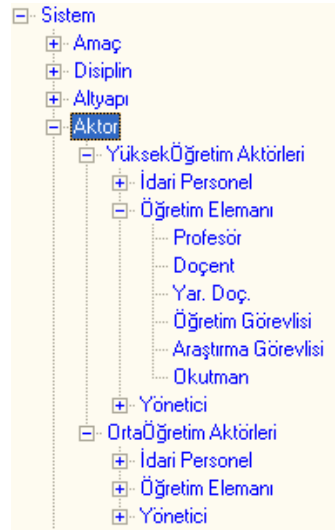
Sekil V- 2. In-CM Referans KM’de Disiplin detaylandırılması örneği.

- 3) **Altyapi:** Sistemler belirli alt yapılar üzerine insâ edilir. Sisteme ait kullanıcı istekleri “fiziksel altyapi” ve “yazılımsal altyapi” olarak iki baslık altında tanımlaması gerçekleştirilir. Sekil V-3’de eğitime göre altyapi örneklendirilmiştir.



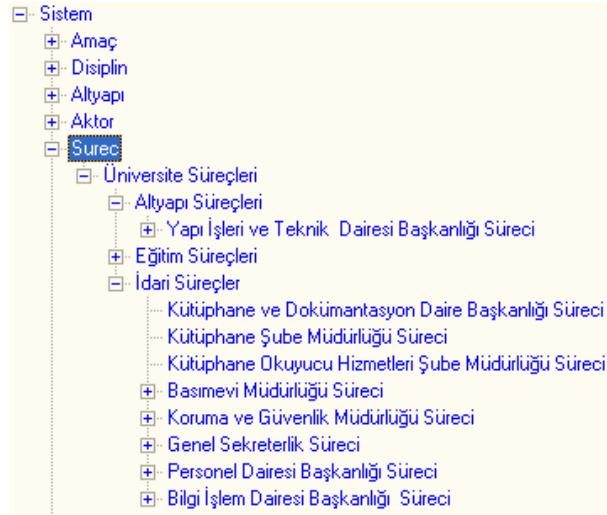
Sekil V- 3. In-CM Referans KM’de Altyapi detaylandırılması örneği.

- 4) **Aktör:** Sistemler içerisinde yer alan ve sisteme girdi veren ya da sistemden etkilenen elemanlardır. Sekil V-4’de eğitime göre aktör örneklendirilmiştir.



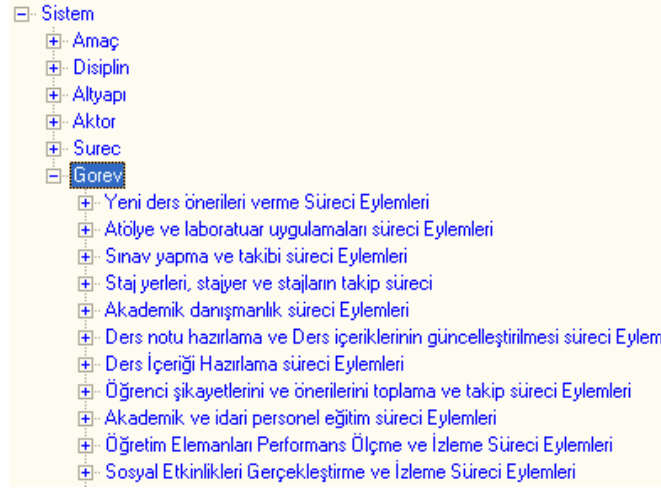
Sekil V- 4. In-CM Referans KM’de Aktör detaylandırılması örneği.

- 5) **Süreç:** Sistem içerisinde yer alan süreç ve bu süreçlere ait alt süreçlerin tanımlamaları Sekil V-5’de eğitime göre örneklendirilmiştir.



Sekil V- 5. In-CM Referans KM’de Süreç detaylandırılması örneği.

- 6) **Görev:** Sistem süreçleri içerisinde, sistem aktörleri tarafından gerçekleştirilecek olan eylemler Sekil V-6’da eğitime göre örneklendirilmiştir.



Sekil V- 6. In-CM Referans KM’de Görev detaylandırılması örneği.

In-CM kapsamında tanımlanan RKM’deki her bir eleman için aşağıdaki tanımlamalardan oluşur:.

1. **Tanım:** RKM içerisinde elemanın gösterilecek olan ismini içerir.
2. **ID:** Kuralların tanımlanması ve işleminde elemanları birbirinden ayırt eden sayısal bir değerdir. RKM içerisinde aynı ID nurasına sahip iki eleman olamaz.
3. **Baglanti Tipi:** RKM içerisinde alt ve üst elemanlar arası bağlantı ilişkisini gösteren tanımlamadır. Bu tanımlama “VE-VEYA” ifadelerinden birini içerir. Eğer bir eleman ile o elemana ait alt eleman arasında “VE” ilişkisi var ise, o eleman seçildiğinde alt elemanda mutlaka seçilmelidir. Alt eleman eğer üst elemana “VEYA” ile bağlanmış ise üst eleman seçildiğinde alt elemanın kavramsal modele dahil edilmesi kullanıcıya bırakılmıştır.

V.3.2 Kurallar

Bilgi Tabanında yer alan ikinci eleman “Kurallar”dır. Kullanıcı isteklerindeki eksikliklerin RKM üzerinden tamamlanabilmesi için RKM’den çıkartılan bazı kurallar vardır. Geliştirilen yapı içerisinde iki tip kural bulunmaktadır.

1. Genel Kurallar
2. Alana Özgü (Uzman tarafından tanımlanan) kurallar.

Genel Kurallar, geliştirilen sistemi kullanacak olanların mutlaka uymak zorunda oldukları kuralları içerir. Bu kurallar kullanıcı tarafından değiştirilemezler. Bunlar:

- Çalışılan alan için **mutlaka** bir “Referans Kavramsal Model” tanımlanmalıdır. Bunun aksi durumunda bu sistem ile herhangi bir kavramsal model geliştirilemeyecektir.
- Tanımlanacak olan Referans Model **mutlaka** 6 yönden tanımlanmalıdır.
- Referans Kavramsal Model içerisinde **mutlaka** alana özgü kurallar tanımlanmalıdır.

Alana Özgü kurallar, nicel ve niteliksel olmak üzere ikiye ayrılır.

Niteliksel Kurallar: RKM üzerinde bulunan elemanların birbirleri ile olan birliktelik ilişkisinin alan uzmanı tarafından tanımlanması ile oluşurlar. Bu kurallar RKM üzerindeki elemanların hangilerinin “birlikte kullanılıp / kullanılmayacağını” belirtirler. Örneğin:

“Niteliksel” Kural

[Üniversite Kompleksi], [Yüksek Öğretim Gerçekleştirme] ile birlikte kullanılmalıdır.

[Okul], [Sınıf] ve [Öğretmen] birlikte kullanılmalıdır.

Niteliksel kurallarda en fazla 6 elemanın birlikte kullanılması ya da kullanılmaması ile ilgili tanımlama yapılabilir.

Nicel Kurallar : RKM üzerinde bulunan elemanların birbirleri ile olan sayısal ilişkisinin alan uzmanı tarafından tanımlanması ile oluşurlar. Bu kurallar RKM üzerindeki elemanların hangilerinin, hangilerini “en az”, “en çok” ya da “tam tamına” kaç adet gerektirdiğini belirtirler. Örneğin:

“Nicel” Kurallar

[Üniversite Kompleksi] en az <1> [Üniversite Aktörleri] ile birlikte kullanılmalıdır.

[Okul] en az <2> [Sınıf] ile birlikte kullanılmalıdır.

[Sınıf] en az <1> [Öğretmen] ile birlikte kullanılmalıdır.

Kurallar sadece metinsel tanımlamalardan oluşmamaktadır. Aynı zamanda kuralla ilgili uzman tarafından girilen açıklamaları da içermektedir. Bu kurallar referans modeli hazırlayan uzman tarafından oluşturulur. Bu kuralların sayısı, kalitesi ve doğruluğu uzman bilgisine bağlıdır. Kuralların doğruluğu ve referans modelin tamlığı, geliştirilecek olan kavramsal modelin kalitesini ve güvenilirliğini arttıracaktır.

In-CM kapsamında tanımlanan Kurallar aşağıdaki tanımlamalardan oluşur:.

1. **Kural Tipi:** Kuralın Niteliksel mi, Nicel mi olduğunu belirtir.

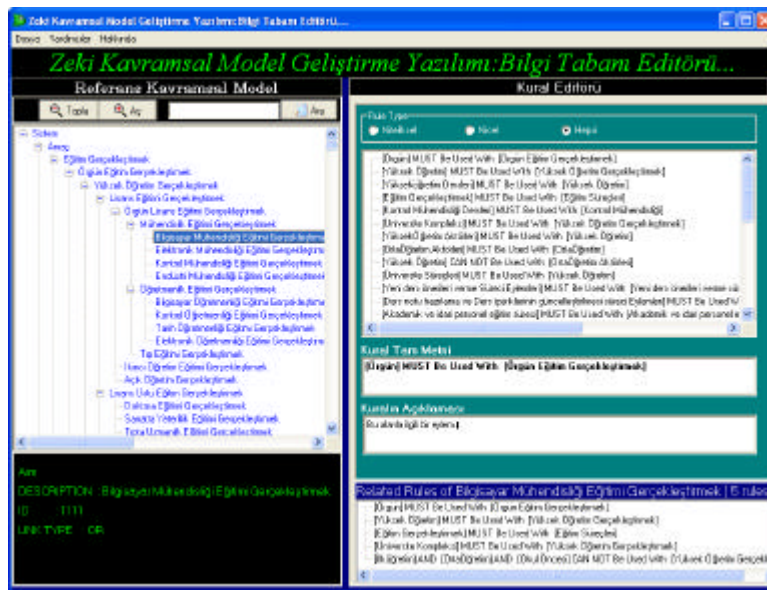
2. **Elemanlar:** Kural içerisinde yer alan elemanların Tanım ve ID'lerini kapsar.
3. **Açıklama:** Kurallar ile ilgili uzman tarafından girilen açıklamaları içerir.

V.3.3 Kullanıcı Kavramsal Modeli

Bilgi Tabanı'nın son elemanı Çıkarım Mekanizması üzerinden oluşturulan "Kullanıcı Kavramsal Modeli"dir. Kullanıcı Kavramsal Model, RKM içerisinde belirli kurallar çerçevesinde ve kullanıcı isterleri doğrultusunda çıkarılan bir alt modeldir. Bu model içerisinde yer alan her bir eleman, RKM olduğu gibi, Tanım, ID, ve Bağlantı Tipi tanımlamalarını içerir.

V.4 In-CM Bilgi Tabanı Editörü

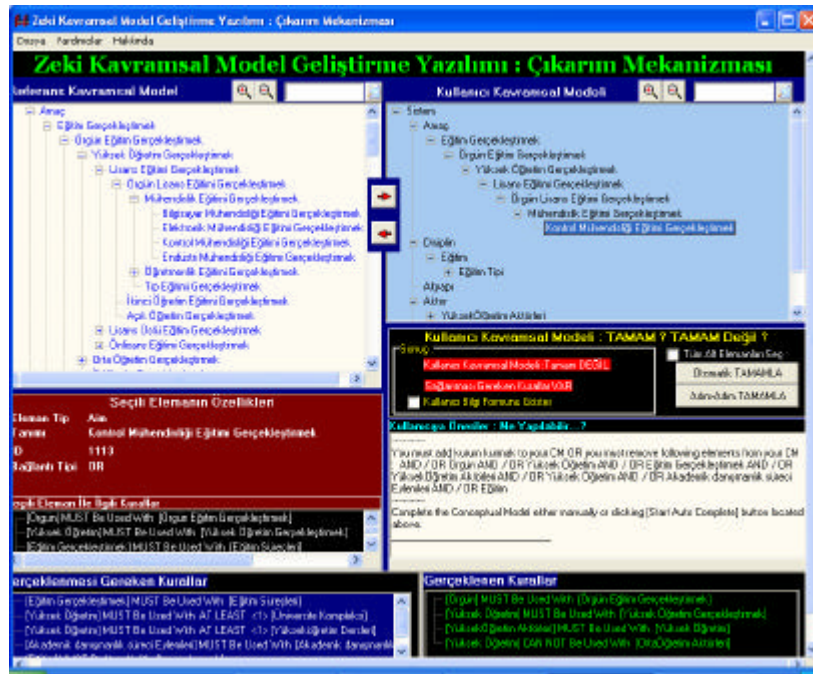
Bu yazılım ile referans kavramsal model ve bu modelle ilgili alana ait kurallar tanımlanır. Bu yazılım kendi içerisinde Niteliksel ve Nicel kural düzenleyicilerine sahiptir. Öncelikle, In-CM Zeki Kavramsal Model yazılımının çalıştırılması için bu editör kullanılarak bir Referans Kavramsal Model oluşturulmalıdır. Daha sonra bu kural düzenleyiciler kullanılarak alana özgü kurallar tanımlanmalıdır. Programın genel görünümü Şekil V-7'de gösterilmektedir.



Şekil V- 7. In-CM Bilgi Tabanı Editörü genel görünümü.

V.5 In-CM Çıkarım Mekanizması

Bu yazılım kullanıcının, kural tabanında referans kavramsal modeli tanımlı herhangi bir alanda kendi kavramsal modelini geliştirmesini sağlar. Öncelikle kullanıcı kendi isteklerini tanımlı olan yapıyı kullanarak yazılıma girer. Her bir istegi ile alana özgü kurallar isler. Böylece kullanıcıni eksik bıraktığı bilgilerin tamamlanması istenir. Kullanıcı bu tamamlamayı kendisi tek tek yapabildiği gibi yazılıma da otomatik olarak tamamlatabilir. Programın genel görünümü Sekil V-8’de gösterilmektedir.



Sekil V- 8. In-CM Çıkarım Mekanizması genel görünümü.

In-CM Çıkarım mekanizması içerisinde ileri doğru zincirleme tekniği kullanılmıştır. Verilen bir durum ile o duruma ait olan kurallar araştırılmış ve kavramsal modelin tamlığı için kullanıcıya tamamlatılmaya çalışılmıştır. In-CM Çıkarım mekanizmasında, kullanıcı bir kavramsal model oluşturmak istediğinde öncelikle, RKM üzerinden kendi isteklerine ait olan alanları “Kullanıcı Kavramsal Modeli (KKM)”’ne ekler. Bu ekleme ile her bir elemana ait kural Çıkarım Mekanizması tarafından bulunur ve kullanıcıya gösterilir. Bunun yanında In-CM çıkarım mekanizması KKM ’nin tamlığını da araştırır. In-CM Çıkarım mekanizması

bu işlemler için Derinlik-ilk Arastırma (Depth-First Search) algoritmasını kullanmıştır. Bu algoritma ile ilgili detaylı bilgi (Lafore, 2003)'de bulunabilir.

In-CM Çıkarım Mekanizması'nda her bir kullanıcı girişi gerçekleştirildikten sonra, bu kullanıcı girişi ile ilgili iki işlem gerçekleştirilir :

- Kullanıcı girişi ile ilgili kuralların bulunup bulunmadığı araştırılır.
- Bu girişim Kullanıcı Kavramsal Modeli üzerine olan etkisi araştırılır.

In-CM Çıkarım mekanizması üzerinde KKM ile ilgili gerçekleştirilmeyen kuralların bulunup bulunmadığına dair bir gösterim vardır. Kullanıcıdan modelle ilgili tüm kuralların tamamlanması istenir. Gerçeklenen kurallarda kullanıcıya ayrı bir alan içerisinde gösterilir. Kullanıcı tüm kuralları gerçekleştirildiği halde kavramsal modeli tam olmayabilir. Bu RKM içerisinde tanımlanan genel kurallardan kaynaklanır. Böyle bir durumda kullanıcı kavramsal modelin tamamlanmasını kendisi yapabildiği gibi bunu çıkarım mekanizmasından da isteyebilir. Otomatik tamamlama da çıkarım mekanizması ikilemde kaldığı durumları kullanıcıya sorar ve tamamlamaya kullanıcının seçeneğine göre devam eder.

V.6 In-CM Grafik Editörü

Bu program “In-CM : Çıkarım Mekanizması” tarafından oluşturulan genel kavramsal modelin tanımlanan grafik notasyona göre gösteren yazılımdır. Bu amaç için bu çalışma kapsamında bir grafiksel gösterim kütüphanesi (notasyon) tanımlanmıştır. Bu notasyonun tanımlanmasında, genel olarak literatür içerisinde yer alan şekiller ile uyumluluk gözetilmesine rağmen, literatür içerisinde yer almayan gösterimler için de bu çalışmaya özgü şekiller tanımlanmıştır. Tablo V-1’de bu gösterim görülmektedir.

“Kullanıcı ve sistem tasarımcısı açısından kavramsal modelin teknik yönden “okunabilirliği”nin tam olması, modelin grafiksel gösterimine bağlıdır. Bu nedenle, kavramsal model, iyi bir grafiksel tasarım prensibine sahip olmalıdır. Bir modelin iyi bir grafiksel tasarıma sahip olması :

- Her sembolün sadece tek bir anlamı olması
- Her kavram için sadece bir tek sembolün kullanılması ile sağlanır. (Hay, 1999)”

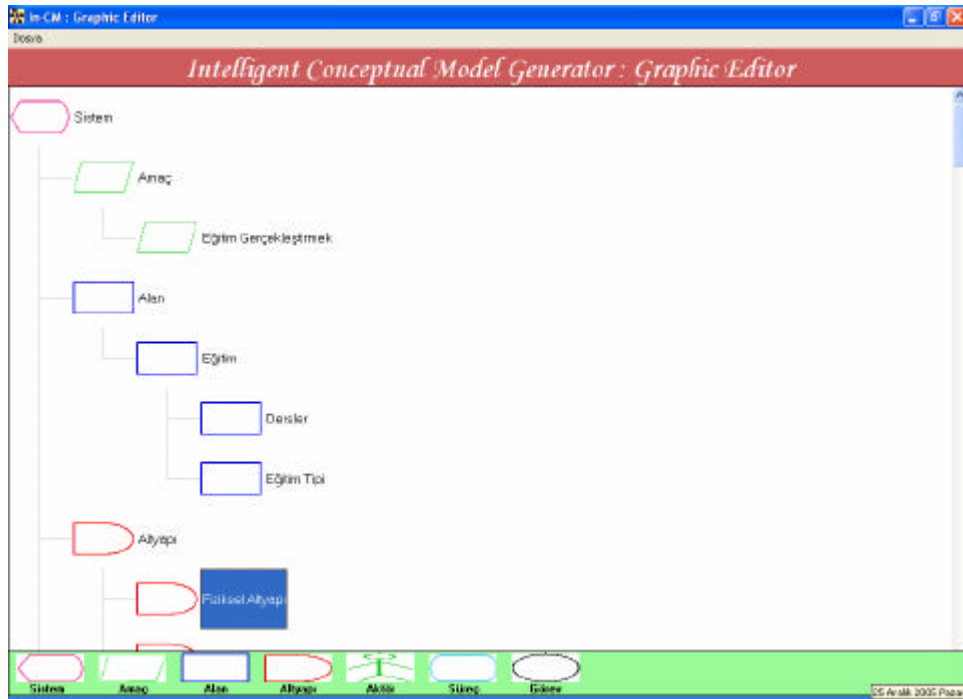
Bu bilgiler Bölüm II’de detaylı olarak anlatılmıştır. Bu tez çalışması kapsamında tanımlanan grafiksel gösterim notasyonunda da bu kriterlere uyulmuştur.

Tanımlanan her şekil sadece bir anlamı ifade eder. Aynı zamanda her kavramda sadece bir şekil ile gösterilmektedir.

Tablo V. 1 In-CM Kavramsal Model notasyonu.

Notasyon	Anlamı
	Sistem
	Amaç
	Alan
	Altyapı
	Aktör
	Süreç
	Görev

Programın genel görünümü Şekil V-9’da gösterilmektedir.



Şekil V- 9. In-CM Grafik Modülünün genel görünümü.

Bir sonraki bölümde, bu bölüm içerisinde anlatılan In-CM yazılımı ile örnek bir uygulama gerçekleştirilmesi anlatılacaktır.

BÖLÜM VI

ÖRNEK UYGULAMA

VI.1 Gelistirilecek olan “Kontrol Mühendisliği Eğitimi” KM ‘in Amacı

Bu çalışmada Bölüm V içerisinde açıklanan özelliklerde geliştirilmiş olan “In-CM Zeki Kavramsal Model Geliştirme” yazılımı gerçek bir problem üzerinde çalıştırılarak önerilen yaklaşım ve yöntemlerin (RKM gibi) uygulanabilirliği gösterilmiştir. Çalışmanın katkısını açık olarak gösterebilmek amacı ile mümkün olduğunca basit bir problem seçilmiştir. Ancak geliştirilen yöntem problemin karmaşıklığından olumsuz yönden etkilenmeyecek durumdadır ve her türlü probleme uygulanabilir niteliktedir. **Bu çalışmada örnek olarak herhangi bir üniversitede kontrol mühendisliği eğitimi verebilecek bir bölüm açılabilmesi için önceden kavramsal olarak bu bölümü açıp sistem elemanları arasındaki ilişkileri izleyebilecek bir model geliştirilmiştir.** Ayrıca, “Kontrol Mühendisliği Eğitimi gerçekleştirme” problemi uzmanlık bilgisinin rahatlıkla temin edileceği, birden fazla uzmanın bulunabileceği ve eksik kalan yönlerinin rahatlıkla görülebileceği bir alan olmasından dolayı da seçilmiştir.

Bu kapsamda öncelikle bir üniversitede bölüm açma konusunda özellikle Kontrol Mühendisliği konusunda gerekli olan tüm uzmanlık bilgilerini içerecek şekilde bir referans model oluşturulmuştur. Oluşturulan RKM yapısı Bölüm IV’de anlatılan formattadır. Daha sonra kontrol mühendisliği alanındaki ilişkileri düzenleyen kurallar geliştirilmiş ve In-CM bilgi tabanı oluşturulmuştur.

Seçilen alan basit olarak görülebilir. Ancak geliştirilen yazılımın anlaşılabilirliğinin sağlanması açısından önemlidir. Uzmanın az ve uzmanlık bilgisinin pahalı olduğu alanlara bakıldığında In-CM gibi bir yapının kullanılması hem maliyet, hem zaman hem de doğruluk açısından önemlidir. Örneğin, askeri yada sivil amaçlı büyük ihalelerin teknik şartname hazırlamalarında ya da ön analizlerinde oldukça maliyetli çalışmalar gerçekleştirilmektedir. Çünkü istenilen sistemin, sistemi gerçekleştireceklere en doğru şekilde tarif edilmesi gereklidir. En doğru şekilde tarif edilme yeterli de değildir. Aynı zamanda tarifi doğru anlaşıldığının da kontrolünün yapılması gereklidir. Kavramsal modelleme bu amaç için kullanılan bir yapıdır.

Emek yoğunluk bir çalışma olan kavramsal modelleme, kişi uzmanlığına bağlıdır. Uzman kişilerin elde tutulamaması ya da uzmanlık bilgilerinin çok yüksek maliyete temin edilmesi karşılaşılan sorunlardandır. Oysaki, In-CM gibi bir yazılım ile birden fazla uzman bilgisi tek bir bilgi tabanı içerisine alınabilmekte ve yıllarca istenilen amaçlar doğrultusunda kullanılabilir.

VI.2 Geleneksel Yollarla KM Geliştirme

Geleneksel olarak eğitim alanında uzman biri tarafından “Kontrol Mühendisliği eğitimi” gerçekleştirmek için gerekli yapının kavramsal modeli geliştirilmek istenirse, öncelikle kullanıcının isteginin analiz edilmesi ve daha sonra istegin doğru anlaşıldığının kavramsal olarak gösterilmesi gerekir. Bu işin gerçekleştirilmesi için uzman tarafından kullanıcının istediği sistem ile ilgili belirtileri unsurların eksik kalan kısımlarının ortaya konulması ve tamamlanması beklenir. Genel olarak burada anlatılan örnek için şu adımlar izlenir:

1. “Kontrol Mühendisliği Eğitimi”nin gerçekleştirilmesi için bir “Üniversite” yapısına ihtiyaç olduğu ortadadır. Aynı zamanda bu yapı içerisinde yer alması gereken “Fakülte”, fakülte içerisindeki sınıflar, laboratuvarlar istek doğrultusunda istenilen sisteme gereksinim olarak uzmanlık bilgisinde eklenmesi gerekir. Bununla birlikte herhangi bir üniversitede olduğu gibi, Rektörlük, Kütüphane, Spor kompleksi vb. fiziksel alt yapıların da belirlenmiş olmasına ihtiyaç vardır. Aksi takdirde Kontrol Mühendisliği Eğitiminin verildiği bir sistem bir bütün halinde tanımlanmamış olur.
2. Kontrol Mühendisliği Eğitimi veren bir sistemin diğer önemli bir unsuru da fonksiyonel işlevlerinin eğitim alanında gerçekleştiriliyor olmasıdır. Bir yükseköğretim kurumunda uygulanan eğitim altyapısı genel olarak okutulacak olan seçmeli ve zorunlu dersler, laboratuvar dersleri gibi konuların açıklığa kavuşturulması demektir. Bu derslerin içeriklerinin de uzman tarafından tanımlanması eğitim kalitesi açısından önem arz eder.
3. Açılacak olan kontrol mühendisliği eğitimi programında eğitim fonksiyonunu yerine getirebilmek için gerekli olan idari ve yönetim süreçlerinin de açık olarak belirlenmiş olması da başarılı bir sistem geliştirme bakımından diğer önemli bir unsurdur. Süreçleri

tanımlamanın en önemli faydası ilgili aktiviteler sorumluklar, görev tanımları ve etkin yönetim unsurlarının dikkate alınmasıdır.

4. Yukarıdakilerinde ötesinde herhangi bir eğitim sisteminin gerçekleştirilmesi için gerekli olan bina, laboratuvar, teçhizat vs. gibi unsurlar gözardı edilemez. Bu durum kontrol mühendisli için oluşturulacak kavramsal model içerisinde de ihmal edilmemesi gerekir.
5. Son olarak her türlü gereksinim tanımlansa bile oluşturulmuş olan sistemin ve alt sistemlerinin en etkin şekilde çalıştırılabilmesi amacı ile ilgili aktörlerin (bu örnekte rektör, yardımcısı, genel sekreteri veya dekan, yardımcısı, fakülte sekreteri, öğrenci işleri daire başkanı gibi) tanımlanması sistemin bütünlü ve tamlığı açısından kaçınılmaz bir unsurdur.

Yukarıda açıklanan unsurların bir araya getirilebilmesi bir çok sistem için kolay değildir. Çünkü sistemin her elemanı diğerleri üzerinde çok fazla sayıda etki üretebilmekte ve bu etkiler karışık sistemlerde rahatça tanımlanamamaktadır. Sistem unsurlarını alt sistemleri ve bu unsurların birbirleri ile ilişkilerini birbirleri üzerindeki etkilerini iyi bilen konu uzmanlarının kavramsal modellerin hazırlanmasında aktif olarak çalışması sistemlerin başarısı açısından önemlidir. Çünkü uzmanlar bilgi ve deneyimleri ile sahip oldukları tecrübelerle benzeri kavramsal çalışmaları sürekli yaptıkları ve kavramsal çalışmaları gerçek sistemlere dönüştürdükleri için sistem unsurlarının birbirleri üzerindeki etkilerini çok daha rahat öngörebilmektedirler. Bu sayede geliştirilecek olan sistemlerde özellikle olmazsa olmazlar mutlaka düşünülmüş olacak, herhangi bir eksiklik söz konusu olmayacaktır. Benzeri şekilde Uzmanlar bu sistemlerde alternatif unsurları da sistem kullanıcılarının istek ve beklentileri doğrultusunda ortaya koyabilir ve opsiyonel çözümleri oluşturabilirler.

Bu açıklamalar ışığında şöyle bir yorum yapmak yanlış olmaz. Eğer kavramsal modelleme çalışmaları mümkün olan yerlerde otomasyona bağlanacak ve otomatik olarak gerçekleştirilecek ise sistemin bir bütün olarak düşünülmesi gerekli olan tüm sistem elemanlarının, bunların birbirleri ile ilişkileri ve etkileşimlerinin dikkate alınması bunlarında ötesinde gerçek hayatta bu işi yapan uzmanların kullandıkları bilgi ve tecrübeleri dikkate alan bir otomasyon sistemi oluşturulmalıdır. İşin doğası gereği bu unsurlar ve etkileşimleri mevcut bilgi ve deneyimleri matematiksel formüller ile ifade etmek mümkün olmadığından geleneksel bilgi

otomasyonu yöntemleri yetersiz kalacaktır. YZ ve uzman sistemler gibi sezgisel bilgiyi isleyen yöntemler kullanılmalıdır. Bu tezde ortaya atılan yaklaşımlar ve geliştirilen yazılım (In-CM) bu teknikler ile donatılmış olup kavramsal modelleme sürecinin otomasyonunun da yapılacağını göstermektedir.

VI.3 In-CM Yazılımı İle Kavramsal Model Geliştirme

Geliştirilmiş olan In-CM yazılımın kavramsal modelleme geliştirebilme yeteneği ve kullandığı tekniklerin olurlulukları “kontrol mühendisliği eğitimi geliştiren” bir sistemin kavramsal modelinin oluşturulması ile gösterilmiştir. Bu bölümde bu örneğin uygulaması detaylı olarak açıklanacaktır.

In-CM ile herhangi bir alanda kavramsal model geliştirilmesi için öncelikle ilgilenilen alan ile ilgili disiplini temsil eden meta model olan Referans Kavramsal Modelin tanımlanması gerekir. Eğer ilgilenilen alan ile ilgili olarak bir referans model tanımlanmamış ise o alan için In-CM yazılımı bir kavramsal model oluşturulması mümkün değildir. Çünkü kavramsal modelleme konusunda sezgisel bilgilerin işlenebilmesinde RKM In-CM için çok önemli bir araçtır. Sistem elemanları ve bunların birbirleri ile ilişkileri ve etkileşimleri RKM içerisinde tanımlanmış olduğundan bu ilişkiler içerisinde kullanıcı kendi beklenti ve ihtiyaçlarına özgü ilişkileri çekip alabilir ve dar kapsamlı kullanıcı KM oluşturabilir. Bu da basit basına yeterli değildir. In-CM, RKM’in yanında Eğitim sistemlerine özgü kuralları da kullanmaktadır. O nedenle Kontrol Mühendisliği eğitimine özgü olan kurallarında tanımlanmış olması gerekmektedir. In-CM bu gereklilikleri karşılayabilmek için kullanıcıya hem referans modelini oluşturma hem de eğitim sistemine özgü kuralları bilgi tabanına girmesini sağlayan olanaklar sunar. Diğer bir ifade ile In-CM’i kullanan kullanıcılar öncelikle In-CM aracılığı ile RKM oluşturmali ve Kuralları sisteme girmelidirler. In-CM tarafından geliştirilecek olan kavramsal modelin bütünlüğü açısından önemlidir. Bu amaçla sistem içerisinde Bilgi Tabanı Editörü Geliştirilmiştir. Bu Editörün en önemli özelliği içindeki kuralların ve RKM’nin çok rahatlıkla değiştirilmesi ve güncellenmesidir.

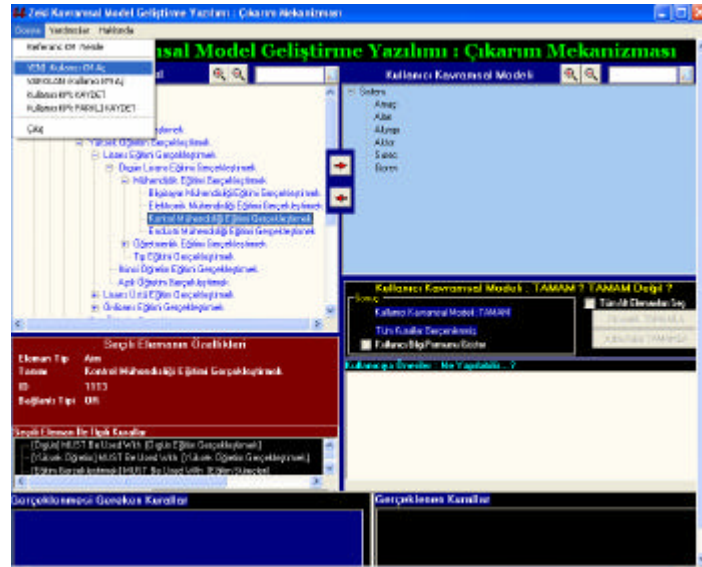
Bu çalışma kapsamında incelenen örnek problem için öncelikle, In-CM sistemi Bilgi Tabanında, eğitim alanına ait bir referans kavramsal model ve eğitim alanına ait kurallar tanımlanmıştır. Kullanıcı bunları kullanarak RKM’den kendi

kullanici kavramsal modellerinin olusturabilir. Bu modeli saklayabilir, üzerinde degisklik yapabilir ve gerekirse yeni kavramsal modeller türetebilir.

Bu tezde sunulan uygulamanin gerçekteştirilmesi asagida örneklerle detayli olarak açıklanmistir. Kullanicinin temel amacinin su sekilde oldugu varsayilmistir :

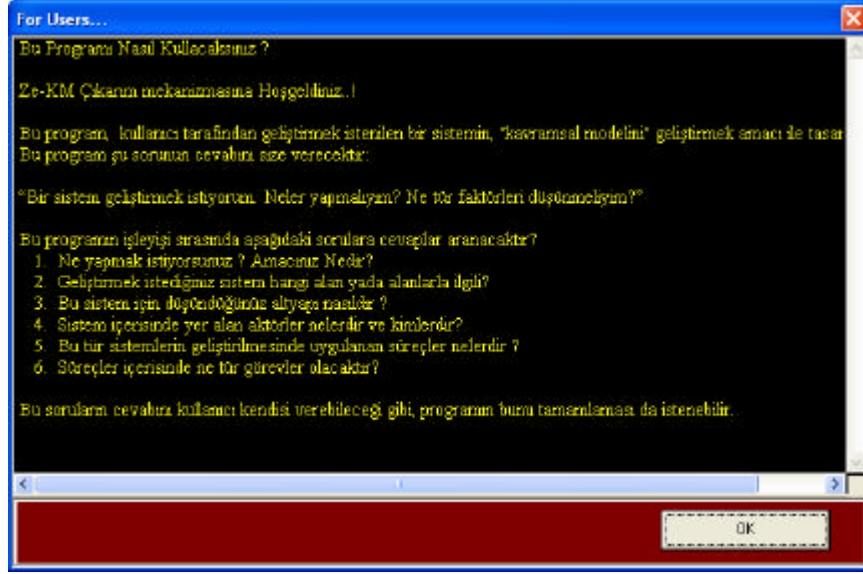
*“Kurulacak olan bir üniversite içerisinde **“Kontrol Mühendisligi Eğitimi”** gerçekteştirecek bir bölüm açmaktır”*

Kullanici bu amaci gerçekteştirecek bir kavramsal model olusturmak istemektedir. Bunun için ya mevcut bir kavramsal modeli günceller veya kavramsal modeli yeni bastan olusturur (ilgili In-CM menu seçenegi : “YENİ:Kullanici KM Aç”). Bu uygulamada yukarıdaki amaci gerçekteştirmek için önerilen kavramsal model In-CM’in tüm özelliklerini gösterebilmek için bastan itibaren tasarlanmistir. Yapilmasi gereken ilk is öncelikle bos bir kullanici kavramsal modeli olusturmaktır. Burada alti açıdan içi bos bir baslangıç modeli olusturmak kastedilmektedir. Sekil VI-1 bunu göstermektedir. Sekilden de görüldüğü gibi kullanici kavramsal modeli RKM’in ana basliklarini içermektedir.



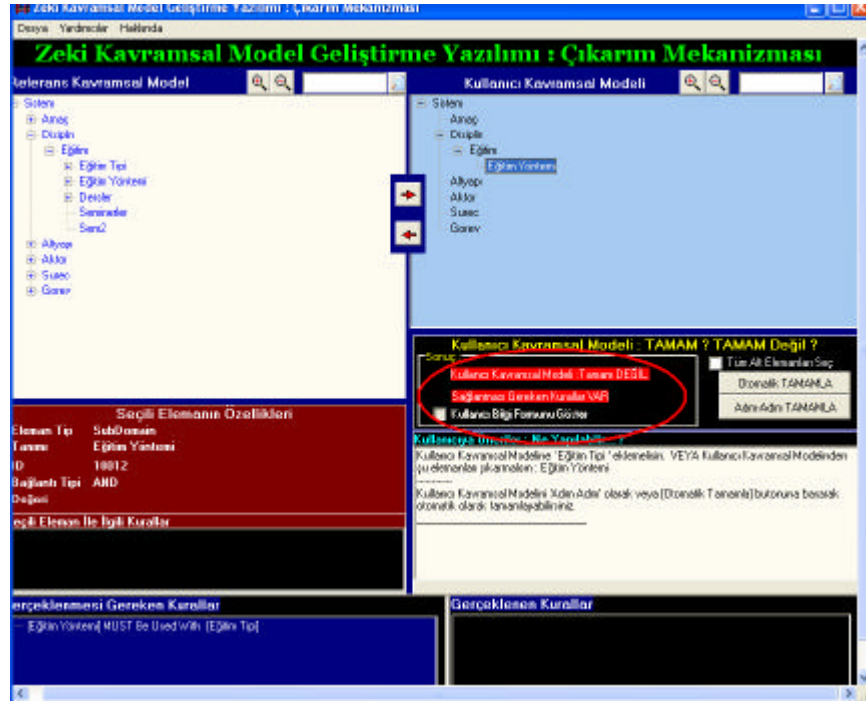
Sekil VI- 1. In-CM’de Yeni Kullanici KM olusturulmasi.

Sistemin en önemli özelliği sürekli olarak kullanici kavramsal modelinin bütünlüğünü izlemek ve bu bütünlüğün sağlanamaması durumunda gerekli ikaz ve önerileri yapmatadır. Bu örnekte su ana kadar bos bir model açıldığından (üzerinde çalışılmak üzere) kullaniciya sistgemi nasıl kullanilmasi gerektiğine dair bilgiler Sekil VI-2’deki gibi sunulur. Kullanici yardımlari kullanici kavramsal modelin tamamlanmasına kadar devam eder.



Sekil VI- 2. In-CM’de Kullanıcı KM olusturulmasında Kullanıcı bilgi ekranı.

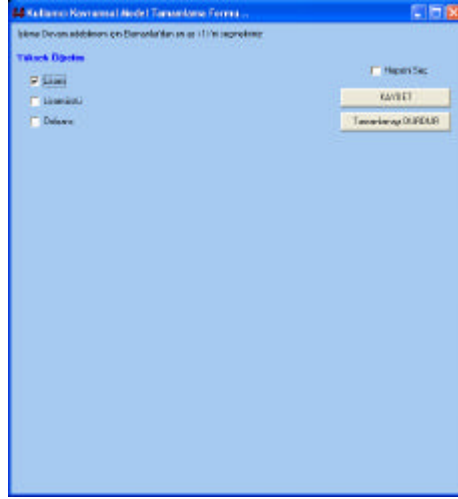
Daha önce açıklandığı gibi RKM belli bir hiyerarsik yapıdadır. Bu yapı içerisinde kullanıcı kendi kavramsal modelini geliştirmek için Referans Kavramsal Modelin herhangi bir noktasından girerek kavramsal modelini tamamlayabilir. Çünkü In-CM sistemi KM üzerinde aşağı ve yukarı ilişkileri kontrol edebilme özeliğine sahiptir. Sistemi zeki yapan özelliklerinden biri budur. Bu uygulamada kullanıcı Eğitim alanında bir model oluşturmak istediğinden RKM’den Disiplin altından “Eğitim” altından “Eğitim Yöntemi” seçmiş olduğunu varsayarsak sistem kullanıcı kavramsal modeline Sekil VI-3’deki gibi bunu ekler ve aşağı-yukarı ilişkileri kontrol ederek kullanıcı kavramsal modelin bütünlüğünün bozulduğunu ve gerçekleşmesi gereken kuralların olduğunu dair yardımlar sistem tarafından sunulur (bkz. kırmızı ile işaretli alan). Bu özelliklik sistemin kullanıcı dostu yapısından kaynaklanmaktadır. In-CM sistemi kavramsal model oluşturma her adımında sistemi kullanana yol gösterilir. Sistem içerisinde yapması ya da yapmaması gerekenler hakkında bilgiler verir. Bu gerçek hayatta kullanıcı ile uzman etkileşiminin karşılığı olarak sisteme kazandırılan yeteneklerdendir.



Sekil VI- 3. In-CM’de KKM’e RKM’den istegin eklenmesi.

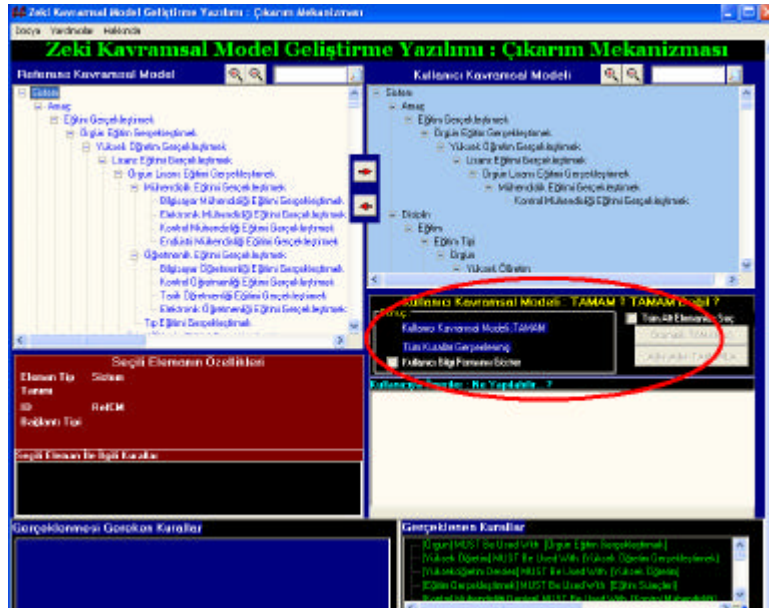
In-CM sisteminde kullanıcı isteklerini, beklentilerini karşılayacak olan özellikleri RKM üzerinde seçerek kullanıcı kavramsal modeli üzerine ekler. Bu eklemelerin KKM üzerindeki etkileri sistem tarafından kullanıcıya bildirilir. kullanıcı kavramsal modeli tamamlanincaya kadar kullanıcıyı bilgilendirir ve yönlendirir. Kullanıcının istegini karşılayan ve tutarlı bir kavramsal model oluşturulduğunda sistem bunu kullanıcıya bildirir.

Kullanıcı oluşturduğu kavramsal modelin sistem tarafından otomatik olarak oluşturulmasını isterse, sistem bunu kullanıcı adına yapabilir durumdadır (ilgili In-CM seçeneği : “Otomatik TAMAMLA”). . Otomatik olarak tamamlamada iste kullanıcı isterlerinin tümü bir bütün olarak ele alınmakta ve tüm isterlerin eksiklikleri sıra ile tamamlanmaktadır. Bu asamada eğer kullanıcı tarafından karar verilmesi gereken alt isterler söz konusu ise bu durum In-CM tarafından kullanıcıya sorulmakta ve kullanıcının görüşü alındıktan sonra kavramsal model geliştirmeye devam edilmektedir. Örneğin, “Kontrol Mühendisli Eğitimi” gerçekleştirir iken “Yüksek Öğretimde” gerçekleştirilecek olan bu eylem içerisinde Lisans, Lisansüstü ve Önlisans yapılarından hangilerinin bulunacağı sistem tarafından Sekil VI-4’deki gibi kullanıcıdan istenecektir. Bu durumda kullanıcı sadece “Lisans” i seçer ise, bu seferde kullanıcıdan “Açık Öğretim”, “Örgün Lisans” ve “İkinci Öğretim”den hangilerinin gerçekleştirileceği sorulacaktır.



Sekil VI- 4. In-CM’de Kullanıcı KM Tamamlama Formu.

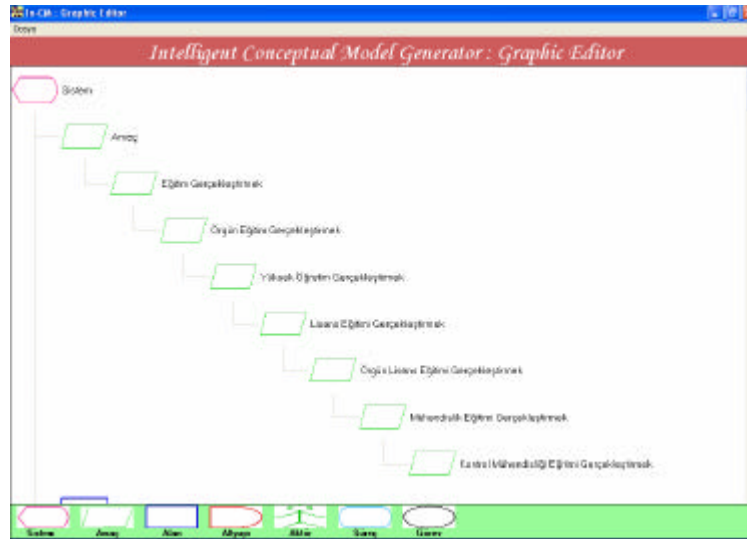
Gerçek hayata da böyle bir sistem geliştirilse idi bu süreç benzer bir şekilde işleyecekti. Bu sefer uzman, kullanıcıya sözlü olarak In-CM sisteminin sorduğu soruları soracak ve bu soruların cevabına göre problemi anlayıp eksik istekleri tamamlamaya çalışacak idi. Kullanıcı tarafından seçim yapılması gereken noktalarda ister gerçek hayatta isterse de In-CM sisteminde olsun eğer kullanıcı tarafından hiçbir seçim yapılmazsa bu durumda kavramsal model oluşturulmaya devam edilemez. Kavramsal Modele devam kullanıcının isteginin ne olduğunu belirtmesi ile ancak gerçekleşir. Tüm eksikler tamamlandıktan sonra KKM tamamlandığı Sekil IV-5’deki gibi In-CM sistemi tarafından kullanıcıya bildirilir.



Sekil VI- 5. In-CM’de KKM Tamamlandıktan sonraki görünüm.

Sistem kullanılarak KKM otomatik oluşturulabileceği gibi KKM üzerinden adım adım tamamlamada gerçekleştirilebilir (ilgili In-CM seçeneği : “Adım-Adım TAMAMLA”). Adım adım tamamlamada kullanıcı isterleri tek tek ele alınmakta ve sadece o isteklerin eksikleri tamamlanmaktadır. Bu işlem otomatik tamamlama da olduğu gibi sistemin karar veremediği ve kullanıcı tarafından yönlendirilmesi gereken tüm isteklerin sonlandırılmasına kadar devam eder.

In-CM sistemi tarafından oluşturulan KKM XML formatındadır. Kullanıcı kavramsal modelinin XML çıktısı Tablo M-1’de verilmistir. Kullanıcı kavramsal modeli aynı zamanda sistem tarafından Bölüm V’de tanımlana grafi notasyonu ile In-CM Grafik Editöründe Şekil VI-6’daki gibi grafiksel olarak da gösterilebilir. Bu şekilde KKM’nin sadece amaç bölümü görülmektedir.



Şekil VI- 6. In-CM’de Grafik Editöründe KKM Amaç bölümü.

Tablo VI. 2 XML Formatındaki Kullanıcı Kavramsal Modeli.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- edited by Ali GÜRBÜZ -->
<Entry>
  <EntryNodeName>Marmara</EntryNodeName>
  <EntryName>UserConceptualModel</EntryName>
  <EntryVersion>2.1</EntryVersion>
  <EntrySequenceNumber>1</EntrySequenceNumber>
  <EntryTypeName>UserConceptualModelEntryType</EntryTypeName>
  <EntryTypeVersion>2.1</EntryTypeVersion>
  <EntryDescription>This is an Example for User Conceptual Model Entry</EntryDescription>
  <EntryCharacterization />
  <EntryDetails>
    <Sistem>
      <Aim description="Amaç" id="45" type_link="">
      <Aim description="Eğitim Gerçekleştirmek" id="1" type_link="OR">
      <Aim description="Örgün Eğitim Gerçekleştirmek" id="100231" type_link="OR">
      <Aim description="Yüksek Öğretim Gerçekleştirmek" id="11" type_link="OR">
```

```
<Aim description="Lisans Egitimi Gerçekleştirmek" id="110321" type_link="OR">
<Aim description="Örgün Lisans Egitimi Gerçekleştirmek" id="1103321" type_link="OR">
<Aim description="Mühendislik Egitimi Gerçekleştirmek" id="111" type_link="OR">
  <Aim description="Kontrol Mühendisliği Egitimi Gerçekleştirmek" id="1113" type_link="OR" />
</Aim >
</Aim >
</Aim >
</Aim >
</Aim >
</Aim >
</Aim >
<Domain description="Disiplin" id="1000" type_link="">
<SubDomain description="Egitim" id="1001" type_link="OR" value="">
<SubDomain description="Egitim Tipi" id="10011" type_link="AND" value="">
<SubDomain description="Örgün" id="101011" type_link="OR" value="">
<SubDomain description="Yüksek Öğretim" id="120091" type_link="OR" value="">
<SubDomain description="Lisans" id="1021601" type_link="OR" value="">
<SubDomain description="Örgün Lisans" id="120910" type_link="OR" value="">
```

```
<SubDomain description="Mühendislik Eğitimi" id="102190" type_link="OR" value="">
  <SubDomain description="Kontrol Mühendisligi" id="10290" type_link="OR" value="" />
</SubDomain>
</SubDomain>
</SubDomain>
</SubDomain>
</SubDomain>
</SubDomain>
</SubDomain>
<SubDomain description="Dersler" id="10013" type_link="AND" value="">
  <SubDomain description="Yükseköğretim Dersleri" id="100131" type_link="OR" value="">
    <SubDomain description="Mühendislik Dersleri" id="1001311" type_link="OR" value="">
      <SubDomain description="Kontrol Mühendisligi Dersleri" id="101811" type_link="OR" value="">
        <SubDomain description="Zorunlu Dersler" id="1001811" type_link="AND" value="">
          <SubDomain description="Mühendislik Bölümü Dersleri" id="126312" type_link="AND" value="">
            <SubDomain description="Kontrol Mühendisligine Giriş" id="6430" type_link="OR" value="" />
            <SubDomain description="Olasılık ve İstatistik" id="6431" type_link="OR" value="" />
            <SubDomain description="Bilgi Sistemlerine Giriş" id="6432" type_link="OR" value="" />
            <SubDomain description="Elektronige Giriş" id="643" type_link="OR" value="" />
          </SubDomain>
        </SubDomain>
      </SubDomain>
    </SubDomain>
  </SubDomain>
</SubDomain>
```

```
<SubDomain description="Intro. to Comp. and Info. System" id="323" type_link="OR" value="" />
<SubDomain description="Elektrik Devre Temelleri" id="235" type_link="OR" value="" />
<SubDomain description="Elektromagnetik Alanlara Giris" id="237" type_link="OR" value="" />
<SubDomain description="Isaretler ve Sistemler" id="238" type_link="OR" value="" />
<SubDomain description="Sistem Modelleme ve Simülasyonu" id="239" type_link="OR" value="" />
<SubDomain description="Ölçme Teknigi ve Algilayicilar" id="240" type_link="OR" value="" />
<SubDomain description="Elektromekanik Enerji Dönüşümü" id="241" type_link="OR" value="" />
<SubDomain description="Sayisal Sistemlere Giris" id="242" type_link="OR" value="" />
<SubDomain description="Güç Elektronik Devreleri" id="243" type_link="OR" value="" />
<SubDomain description="Geribeslemeli Kontrol Sistemleri" id="244" type_link="OR" value="" />
<SubDomain description="Elektronik Enstrümantasyon" id="245" type_link="OR" value="" />
<SubDomain description="Mikrokontrolör Tabanlı Sistemler" id="246" type_link="OR" value="" />
<SubDomain description="Kontrol Sistem Tasarımı" id="247" type_link="OR" value="" />
<SubDomain description="Servo Motorlar" id="248" type_link="OR" value="" />
<SubDomain description="Endüstriyel Kumanda Sistemleri" id="249" type_link="OR" value="" />
<SubDomain description="Robotige Giris" id="250" type_link="OR" value="" />
<SubDomain description="Bilgisayar Kontrollü Sistemler" id="251" type_link="OR" value="" />
<SubDomain description="Mühendislik Bilimine Giris" id="2514" type_link="OR" value="" />
```

```

</SubDomain>
<SubDomain description="Laboratuvarlar" id="6541" type_link="AND" value="">
  <SubDomain description="Fizik Lab.1" id="342" type_link="OR" value="" />
  <SubDomain description="Fizik Lab.2" id="3420" type_link="OR" value="" />
  <SubDomain description="Kontrol Lab." id="34221" type_link="OR" value="" />
  <SubDomain description="Elektronige Giris Lab." id="3422" type_link="OR" value="" />
  <SubDomain description="Genel Kimya I Lab" id="343" type_link="OR" value="" />
  <SubDomain description="Güç Elektronigi Lab." id="345" type_link="OR" value="" />
  <SubDomain description="CAD /CAM Lab." id="263" type_link="OR" value="" />
  <SubDomain description="Bilgisayar Lab." id="262" type_link="OR" value="" />
  <SubDomain description="Telekominikasyon Lab." id="264" type_link="OR" value="" />
</SubDomain>
</SubDomain>
<SubDomain description="Seçmeli Dersler" id="1004811" type_link="AND" value="">
<SubDomain description="Mühendislik Bölümü Dersleri" id="1236312" type_link="AND" value="">
  <SubDomain description="Kontrol ve Otomasyon Sistemlerinin Tarihçesi" id="64130" type_link="OR" value="" />
  <SubDomain description="Sayısal Yöntemler" id="64131" type_link="OR" value="" />
  <SubDomain description="Yöneylem Arastirmasi" id="64132" type_link="OR" value="" />

```

```
<SubDomain description="Mühendislikte Proje Yönetimi" id="6413" type_link="OR" value="" />
<SubDomain description="Kalite Yönetimi" id="5312" type_link="OR" value="" />
<SubDomain description="Uzman Sistemlere Giriş" id="3213" type_link="OR" value="" />
<SubDomain description="Elektrik Makinalarında İleri Teknikler ile Hiz Ayar Projesi" id="2314" type_link="OR" value="" />
<SubDomain description="Elektrik Tesislerinde Güvenlik" id="2315" type_link="OR" value="" />
<SubDomain description="Güç Elektroniginin Endüstriyel Uygulamaları" id="2316" type_link="OR" value="" />
<SubDomain description="Bina Otomasyonuna Giriş" id="2317" type_link="OR" value="" />
<SubDomain description="Bilgisayar Destekli Tasarım ve üretim" id="2318" type_link="OR" value="" />
<SubDomain description="Optimal Kontrola Giriş" id="2319" type_link="OR" value="" />
<SubDomain description="Zeki Kontrol Sistemleri" id="2410" type_link="OR" value="" />
<SubDomain description="Biyolojik Sistemlerin Modellenmesi ve Kontrolü" id="2411" type_link="OR" value="" />
<SubDomain description="Elektrikli Ulaşım Sistemleri" id="2412" type_link="OR" value="" />
<SubDomain description="Elektrik Enerji Sistemlerinin Kontrolüne Giriş" id="2413" type_link="OR" value="" />
<SubDomain description="Power System Control and Communication" id="2414" type_link="OR" value="" />
<SubDomain description="Modern Kontrol Teorisi" id="2415" type_link="OR" value="" />
<SubDomain description="Bilgisayar Haberleşmesi" id="2416" type_link="OR" value="" />
<SubDomain description="Robot Kontrolü" id="2417" type_link="OR" value="" />
```

```
<SubDomain description="Elektrik Enerji Sistemleri" id="2418" type_link="OR" value="" />
<SubDomain description="Elektrik Makinalarının Dinamisinin Temelleri" id="2419" type_link="OR" value="" />
<SubDomain description="Endüstriyel Kumanda Sistemleri" id="2510" type_link="OR" value="" />
<SubDomain description="Nesneye Dayalı Programlama" id="2511" type_link="OR" value="" />
<SubDomain description="Termodinamik" id="25114" type_link="OR" value="" />
<SubDomain description="Kompleks Değişkenli Fonksiyonlar Teorisi" id="24168" type_link="OR" value="" />
<SubDomain description="Kismi Türevli Diferansiyel Denklemler" id="24169" type_link="OR" value="" />
<SubDomain description="Optik" id="25150" type_link="OR" value="" />
<SubDomain description="Dinamik" id="25161" type_link="OR" value="" />
<SubDomain description="Statik ve Mukavemet" id="251154" type_link="OR" value="" />
</SubDomain>
</SubDomain>
</SubDomain>
</SubDomain>
</SubDomain>
</SubDomain>
</SubDomain>
</SubDomain>
</Domain>
```

```

<Infrastructure description="Altyapi" id="2000" type_link="">
<InfType description="Fiziksel Altyapi" id="20003" type_link="AND">
<InfType description="Bina" id="20001" type_link="OR">
<InfType description="Üniversite Kompleksi" id="3773" type_link="AND">
<InfType description="Fakülte" id="373" type_link="AND">
  <InfType description="Derslik" id="2365" type_link="AND" />
<InfType description="Odalar" id="3180" type_link="AND">
  <InfType description="Dekan Odasi" id="34180" type_link="AND" />
  <InfType description="Genel Sekreter Odasi" id="314480" type_link="AND" />
  <InfType description="Dekan Yardimcisi Odasi" id="342380" type_link="AND" />
  <InfType description="Hizmetliler Odasi" id="348220" type_link="AND" />
</InfType>
<InfType description="Laboratuvar" id="3174" type_link="AND">
  <InfType description="Robotik Laboratuvari" id="317201" type_link="OR" />
  <InfType description="Fizik Laboratuvari" id="311701" type_link="OR" />
  <InfType description="Kimya Laboratuvari" id="373201" type_link="OR" />
  <InfType description="Biyoloji Laboratuvari" id="3173301" type_link="OR" />
  <InfType description="Bilgisayar Laboratuvari" id="374201" type_link="OR" />

```



```
<InfType description="Kontrol Laboratuvarı" id="3172201" type_link="OR" />
<InfType description="Elektronik Laboratuvarı" id="3733301" type_link="OR" />
<InfType description="Güç Elektronigi Laboratuvarı" id="3714401" type_link="OR" />
<InfType description="CAD /CAM Laboratuvarı" id="341701" type_link="OR" />
<InfType description="Telekominikasyon Laboratuvarı" id="3214701" type_link="OR" />

</InfType>

</InfType>

<InfType description="Kütüphane" id="30274" type_link="AND" />
<InfType description="Rektörlük" id="330274" type_link="AND" />
<InfType description="Spor Kompleksi" id="203012" type_link="OR">
  <InfType description="Spor Salonu" id="32701" type_link="OR" />
  <InfType description="Yüzme Havuzu" id="332701" type_link="OR" />
  <InfType description="Kondisyon Salonu" id="3324701" type_link="OR" />
  <InfType description="Sauna" id="424701" type_link="OR" />
</InfType>

</InfType>

</InfType>

<InfType description="AygıtTechizat" id="20002" type_link="AND">
```

```
<InfType description="Bilgisayar" id="3451" type_link="OR" />
<InfType description="Server" id="365" type_link="OR" />
<InfType description="Is Istasyonu" id="366" type_link="OR" />
<InfType description="Masa" id="367" type_link="OR" />
<InfType description="Sandalye" id="368" type_link="OR" />
<InfType description="Sira" id="369" type_link="OR" />
<InfType description="Kablo" id="370" type_link="OR" />
</InfType>
</InfType>
</Infrastructure>
<Actor description="Aktör" id="4000" type_link="">
<AcType description="YüksekÖğretim Aktörleri" id="4700" type_link="OR">
<AcType description="Yönetici" id="900" type_link="AND">
  <AcType description="Rektör" id="760" type_link="OR" />
  <AcType description="Rektör Yardimcisi" id="762" type_link="OR" />
  <AcType description="Dekan" id="761" type_link="OR" />
  <AcType description="Dekan Yardimcisi" id="763" type_link="OR" />
  <AcType description="Bölüm Baskani" id="764" type_link="OR" />
```

```
<AcType description="Bölüm Başkan Yardimcisi" id="765" type_link="OR" />
<AcType description="Anabilim Dalı Başkanı" id="768" type_link="OR" />
<AcType description="Anabilim Dalı Başkan Yardimcisi" id="769" type_link="OR" />
<AcType description="Müdür" id="766" type_link="OR" />
<AcType description="Müdür Yardimcisi" id="767" type_link="OR" />
</AcType>
<AcType description="Öğretim Elemanı" id="800" type_link="AND">
  <AcType description="Profesör" id="732" type_link="OR" />
  <AcType description="Doçent" id="731" type_link="OR" />
  <AcType description="Yar. Doç." id="730" type_link="OR" />
  <AcType description="Öğretim Görevlisi" id="734" type_link="OR" />
  <AcType description="Araştırma Görevlisi" id="733" type_link="OR" />
  <AcType description="Okutman" id="735" type_link="OR" />
</AcType>
<AcType description="İdari Personel" id="700" type_link="AND">
  <AcType description="Sekreter" id="701" type_link="OR" />
  <AcType description="Hizmetli" id="702" type_link="OR" />
  <AcType description="Mübasir" id="703" type_link="OR" />
```

```
<AcType description="Özel Kalem" id="713" type_link="OR" />
<AcType description="Fakülte Sekreteri" id="724" type_link="OR" />
</AcType>
</AcType>
</Actor>
<Process description="Surec" id="6000" type_link="">
  <SubProcess description="Üniversite Süreçleri" id="19032" type_link="OR">
    <SubProcess description="Altyapi Süreçleri" id="9032" type_link="AND">
      <SubProcess description="Yapi Isleri ve Teknik Dairesi Baskanligi Süreci" id="9145" type_link="OR">
        <SubProcess description="Yapi Isleri Sube Müdürlüğü Süreci" id="9146" type_link="OR" />
        <SubProcess description="Teknik Bakim Ve Onarim Sube Müdürlüğü Süreci" id="9147" type_link="OR" />
        <SubProcess description="Peyzaj Ve Planlama Sube Müdürlüğü Süreci" id="9148" type_link="OR" />
        <SubProcess description="EtüdProje Sube Müdürlüğü Süreci" id="9149" type_link="OR" />
      </SubProcess>
    </SubProcess>
    <SubProcess description="Egitim Süreçleri" id="9084" type_link="AND">
      <SubProcess description="Ders notu hazirlama ve Ders içeriklerinin güncelleştirilmesi süreci" id="9086"
type_link="OR" />
    </SubProcess>
  </SubProcess>
</Process>
```

```
<SubProcess description="Ders Programi Hazirlama Süreci" id="9087" type_link="OR" />
<SubProcess description="Ders Asistani Atama Süreci" id="9088" type_link="OR" />
<SubProcess description="Degerlendirme Süreci" id="9089" type_link="OR" />
<SubProcess description="Yil içi projesi ve Bitirme Ödevi Süreci" id="9090" type_link="OR" />
<SubProcess description="Ders Anlatma Süreci" id="9091" type_link="OR" />
<SubProcess description="Staj yerleri, stajyer ve stajların takip süreci" id="9092" type_link="OR" />
<SubProcess description="Lab. Derslerini Hazirlama Süreci" id="9093" type_link="OR" />
<SubProcess description="Derse Devam ve Takip Süreci" id="9094" type_link="OR" />
<SubProcess description="Sinav yapma ve takibi süreci" id="9095" type_link="OR" />
<SubProcess description="Stratejik Planlama Süreci" id="9096" type_link="OR" />
<SubProcess description="Yeni ders önerileri verme süreci" id="9097" type_link="OR" />
<SubProcess description="Atölye ve laboratuvar uygulamaları süreci" id="9098" type_link="OR" />
<SubProcess description="Akademik danışmanlık süreci" id="9099" type_link="OR" />
<SubProcess description="Öğrenci şikayetlerini ve önerilerini toplama ve takip süreci" id="9100" type_link="OR" />
<SubProcess description="Akademik ve idari personel eğitim süreci" id="9101" type_link="OR" />
<SubProcess description="Öğretim Elemanları Performans Ölçme ve İzleme Süreci" id="9102" type_link="OR" />
<SubProcess description="Sosyal Etkinlikleri Gerçekleştirme ve İzleme Süreci" id="9103" type_link="OR" />
```

```
<SubProcess description="Öğrenci İşleri Süreci" id="9104" type_link="OR">
  <SubProcess description="Öğrenci İşleri Dairesi Başkanlığı Süreci" id="9105" type_link="OR" />
  <SubProcess description="Öğrenci İşleri Sube Müdürlüğü" id="9106" type_link="OR" />
  <SubProcess description="İstatistik ve Disiplin Sube Müdürlüğü" id="9107" type_link="OR" />
</SubProcess>

<SubProcess description="Mezunlar, Belgeler ve Burslar Sube Müdürlüğü Süreci" id="9108" type_link="OR">
  <SubProcess description="Harçlar Sube Müdürlüğü Süreci" id="9109" type_link="OR" />
</SubProcess>

<SubProcess description="Proje Basvurusunda Bulunma ve Değerlendirme Süreci" id="9110" type_link="OR" />
<SubProcess description="Üniversite Sanayi İşbirliği Geliştirme Süreci" id="9111" type_link="OR" />
<SubProcess description="Toplantı Yapma Süreci" id="9112" type_link="OR" />
<SubProcess description="Dersi veren ve ders yükü belirlenmesi süreci" id="9113" type_link="OR" />
<SubProcess description="Proje / Danışmanlık Hizmetleri ve Sertifikalandırma Süreci" id="9114" type_link="OR" />
<SubProcess description="Teklif Verme Süreci" id="91105" type_link="OR" />
<SubProcess description="Performans İzleme Süreci" id="91106" type_link="OR" />
</SubProcess>

<SubProcess description="İdari Süreçler" id="9115" type_link="AND">
```

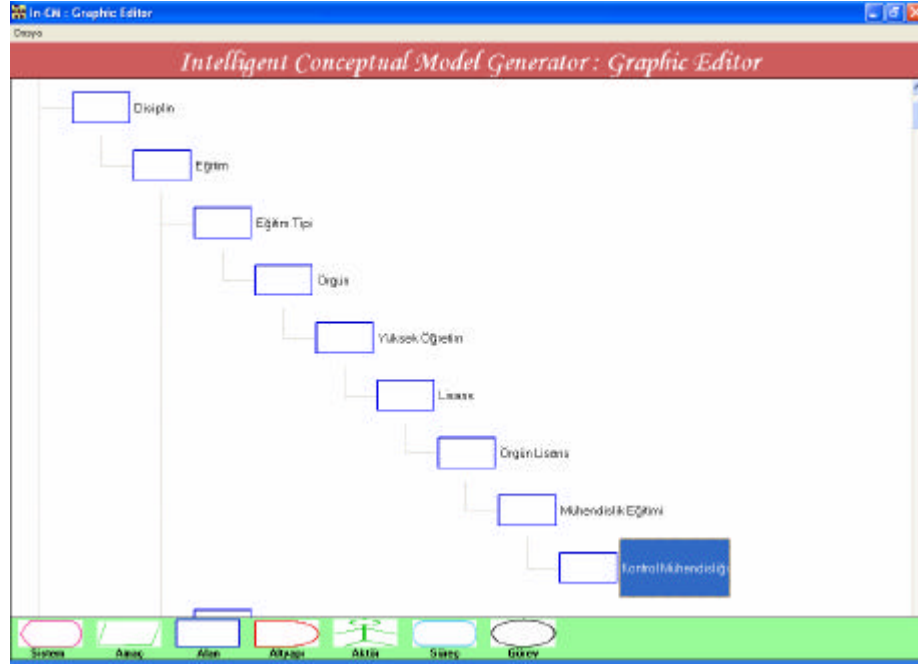
```
<SubProcess description="Kütüphane ve Dokümantasyon Daire Başkanlığı Süreci" id="91107" type_link="OR" />
<SubProcess description="Kütüphane Sube Müdürlüğü Süreci" id="91108" type_link="OR" />
<SubProcess description="Kütüphane Okuyucu Hizmetleri Sube Müdürlüğü Süreci" id="91109" type_link="OR" />
<SubProcess description="Basımevi Müdürlüğü Süreci" id="9121" type_link="OR">
  <SubProcess description="Basın ve Halkla İlişkiler Müsavırlığı Süreci" id="9122" type_link="OR" />
  <SubProcess description="Hukuk Müsavırlığı Süreci" id="9123" type_link="OR" />
  <SubProcess description="Genel Hizmetler Sube Müdürlüğü Süreci" id="9124" type_link="OR" />
  <SubProcess description="Bütçe Sube Müdürlüğü Süreci" id="9125" type_link="OR" />
  <SubProcess description="Satınalma Sube Müdürlüğü Süreci" id="9126" type_link="OR" />
  <SubProcess description="Ayniyat ve Levazım Sube Müdürlüğü Süreci" id="9127" type_link="OR" />
  <SubProcess description="Genel Evrak Sube Müdürlüğü Süreci" id="9128" type_link="OR" />
  <SubProcess description="Arsiv Hizmetleri Sube Müdürlüğü Süreci" id="9129" type_link="OR" />
</SubProcess>
<SubProcess description="Koruma ve Güvenlik Müdürlüğü Süreci" id="9130" type_link="OR">
  <SubProcess description="Sivil Savunma Uzmanlığı Süreci" id="9131" type_link="OR" />
  <SubProcess description="Sağlık , Kültür ve Spor Dairesi Başkanlığı Süreci" id="9132" type_link="OR" />
  <SubProcess description="Spor Sube Müdürlüğü Süreci" id="9133" type_link="OR" />
  <SubProcess description="Kültür İşleri Sube Müdürlüğü Süreci" id="9134" type_link="OR" />
```

```
<SubProcess description="Mediko Sosyal Merkezi Müdürlüğü Süreci" id="9135" type_link="OR" />
<SubProcess description="Birim Saymanligi Süreci" id="9136" type_link="OR" />
</SubProcess>
<SubProcess description="Genel Sekreterlik Süreci" id="9137" type_link="OR">
  <SubProcess description="Kalite Yönetim Koordinatörlüğü Süreci" id="9138" type_link="OR" />
</SubProcess>
<SubProcess description="Personel Dairesi Başkanligi Süreci" id="9139" type_link="OR">
  <SubProcess description="Idari Tayin Sube Müdürlüğü Süreci" id="9140" type_link="OR" />
  <SubProcess description="Özlük Haklari Tahakkuk Sube Müdürlüğü Süreci" id="9141" type_link="OR" />
  <SubProcess description="Sicil Sube Müdürlüğü Süreci" id="9142" type_link="OR" />
  <SubProcess description="Hizmetiçi Egitim Sube Müdürlüğü Süreci" id="9143" type_link="OR" />
  <SubProcess description="Akademik Tayin Sube Müdürlüğü Süreci" id="9144" type_link="OR" />
</SubProcess>
<SubProcess description="Bilgi Islem Dairesi Başkanligi Süreci" id="9150" type_link="OR">
  <SubProcess description="Bilgi Islem Sube Müdürlüğü Süreci" id="9151" type_link="OR" />
  <SubProcess description="Internet Hizmetleri Sube Müdürlüğü Süreci" id="9152" type_link="OR" />
  <SubProcess description="ArGe Sube Müdürlüğü Süreci" id="9153" type_link="OR" />
</SubProcess>
```



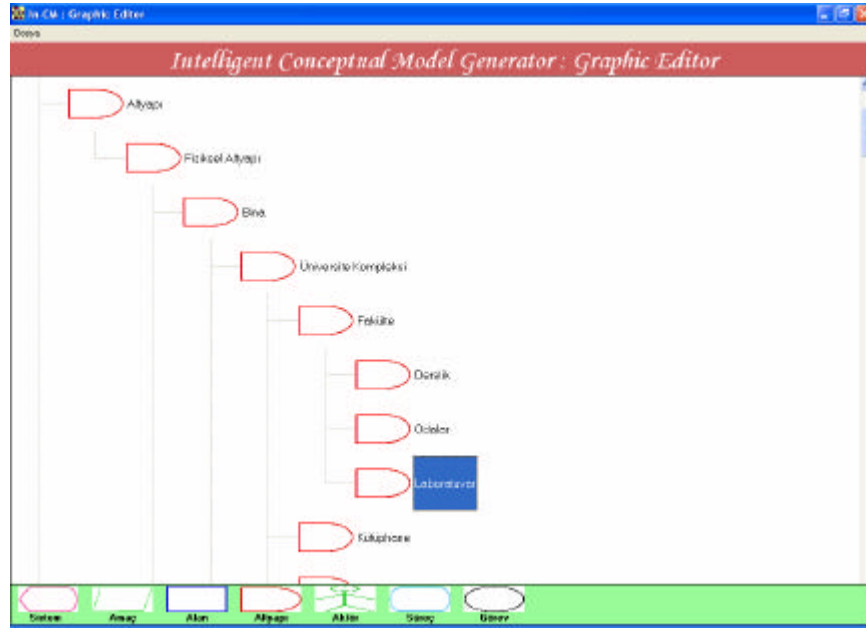
```
</SubProcess>
</SubProcess>
</Process>
<Task description="Gorev" id="7000" type_link="">
<Text description="Akademik ve idari personel eğitim süreci Eylemleri" id="90008" type_link="OR">
  <Text description="Eğitim verecek konu, zaman ve kişilerin tespiti" id="9043" type_link="OR" />
</Text>
</Task>
</Sistem>
</EntryDetails>
</Entry>
```

Tablo VI-1’de verilen dosyanın grafik gösteriminin tek bir ekran görünümüne sigması imkansızdır. Sekil VI-7’de isterin Disiplin ayrıştırılması verilmistir. Disiplin olarak istegin hangi alanda yer aldigi ve tüm diger alt gereksinimleri sistem tarafından ortaya konulmus ve grafiksel olarak kavramsal model içerisinde gösterilmiştir.



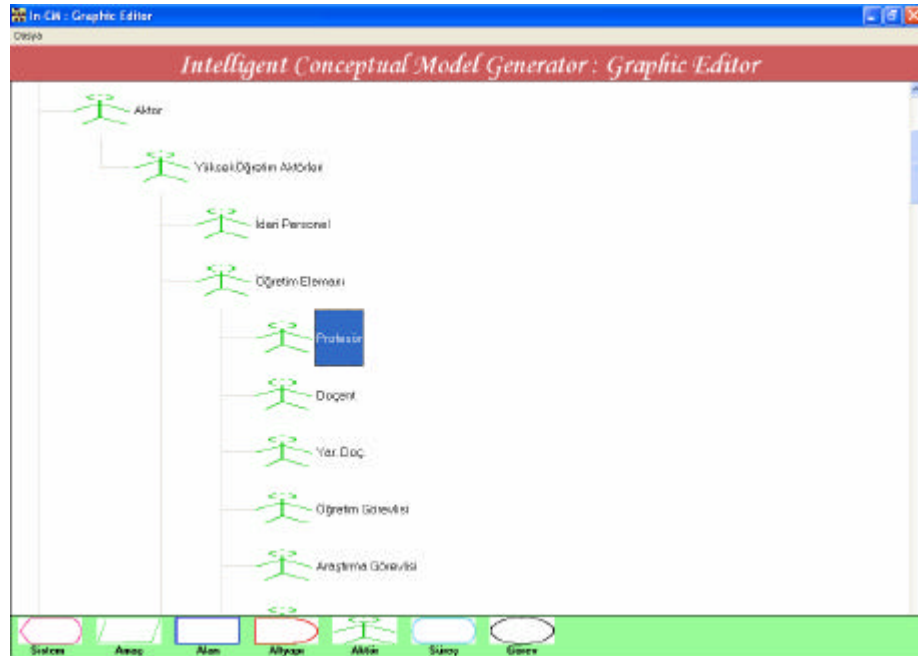
Sekil VI- 7. In-CM’de Grafik Editöründe KKM Disiplin bölümü.

Sekil VI-8’de istegin In-CM sistemi tarafından ortaya konulan Altyapi ayrıştırılmasının bir bölümü verilmistir. Altyapi ayrıştırılmasında isterin gerçekleşmesi için fiziksel olarak bina ve aygıt-teçhizat gereksinimleri tanımlanmaktadır. Istergn bir Üniversite kompleksi gerektirmesi In-CM sisteminin bu yönde tanımlamalar ortaya koyması ile sonuçlanmıştır.



Sekil VI- 8. In-CM’de Grafik Editöründe KKM Altyapi bölümü.

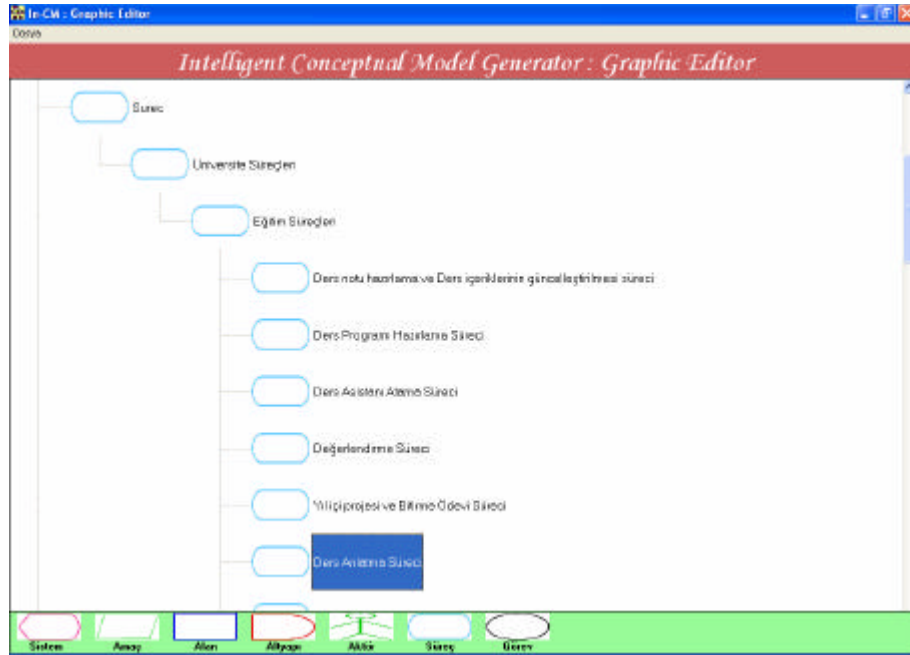
In-CM sistemi, kullanıcı isterinin gerektirdiği aktörler yönetici, öğretim elemanı ve hizmet personeli şeklinde gruplandırılarak tanımlanmıştır. Sekil VI-9’da bu tanımlamanın bir bölümü In-CM Grafik Editörü çıktısı olarak görülmektedir.



Sekil VI- 9. In-CM’de Grafik Editöründe KKM Aktör bölümü.

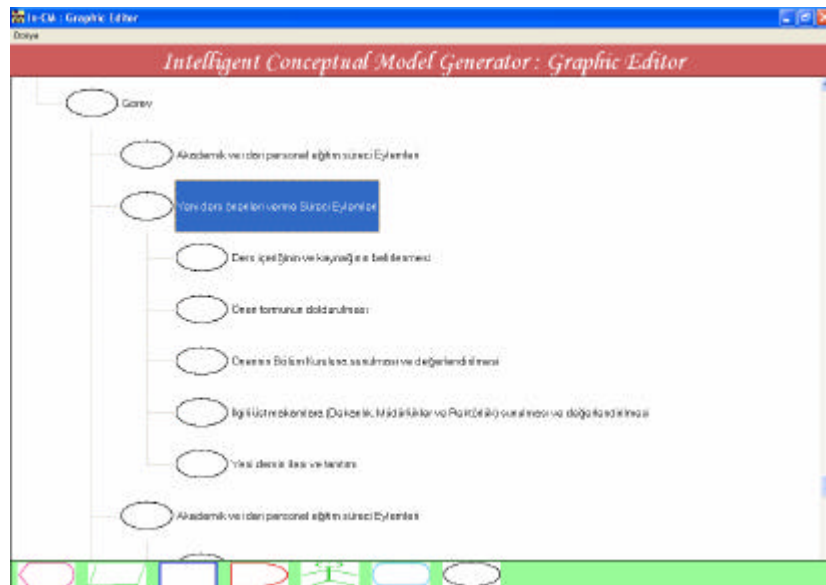
Bir eğitim yapısı içerisinde gerçekleşmesi gereken altyapı, eğitim ve idari süreçler vardır. Bu süreçler In-CM sistemi tarafından örnek istegin gereksinimi

olarak tanımlanmıştır. Şekil VI-10’da örnek istek için tanımlanan süreçlerin bir bölümü Grafik Editör çıktısı olarak gösterilmektedir.



Şekil VI- 10. In-CM’de Grafik Editöründe KKM Süreç bölümü.

Süreçler içerisinde yer alan görev tanımlamaları gereksinim olarak In-CM sistemi tarafından tanımlanmıştır. Tanımlanan görevlerden bir bölümü In-CM Grafik editörü çıktısı olarak Şekil VI-11’de görülmektedir.



Şekil VI- 11. In-CM’de Grafik Editöründe KKM Görev bölümü.

VI.4 Gelistirilen Sistemin Zeki Özellikleri

In-CM Kavramsal Model Gelistirme Araci uzman sistem olarak tasarlanmistir. Bir uzman, kendi uzmanlik alaninda, kendisine danisilan konularda rehberlik yapabilir ve yol gösterebilir. Ayni sekilde In-CM Kavramsal Model Gelistirme Araci da RKM'sine sahip oldugu disiplin içerisinde o disiplinin uzmani olmayan bir insana rehberlik yapiyor, o konu ile ilgili kavramsal model gelistirmesi için nelere bakmasi gerektigini gösteriyor. In-CM yazilimi bu islemi yaparken, bu isi gerçekte o okunun uzmani nasil gerçeklestiriyor ise aynen o sekilde yapiyor. Kullanici tarafından eksik birakilan noktalar sistem tarafından tamamlaniyor. Kullanici istekleri arasindaki tutarsizliklar belirlenerek kullaniciya düzeltmesi için bildiriyor. Kullaniciya alternatif seçenekler sunuyor.

Kullanici Km olusturmaya istedigii seviyenden basiyabiliyor. Çünkü In-CM sistemi KM üzerinde asagi ve yukari iliskileri kontrol edebilme özelligine sahiptir. Sistemi zeki yapan özelliklerinden biri de budur.

Birden fazla uzmanin bilgisi ile kavramsal model gelistirmeye yordimci oluyor. Böylece daha saglam ve tutarli bir uzmanlik bilgi ile kavramsal modeller gelistiriyor.

Uzmanin az oldugu ya da uzmanlik bilgisine ulasilmanin zor veya maliyetli oldugu alanlarda bu bilgiye bir defa ulasmak sarti ile o disiplin içerisinde farkli kavramsal modeller gelistirmeye imkan sagliyor.

Kavramsal modellerin olusturulmasinda önemli noktalardan biri de modellerin tamligi uzmanin uzmanlik bilgisi ile kavramsal modelleme tekniklerine ne kadar hakim olduguna baglidir. Yani uzman çalıştigi konu üzerine ne kadar hakim ise o sisteme ait olan sinirlarin belirlenmesi o derece net olacaktır. Uzmanin kavramsal modelleme tekniklerine hakim olmasi ile de sinirlari net olarak belirlenen problemin kavramsal modeli tam olarak çizilebilecektir. Bu nedenle, bu yolla gelistirilecek olan kavramsal modeller uzmandan uzmana farklılık gösterecektir. Oysaki In-CM birden fazla uzmanin uzmanlik bilgisini bilgi tabaninda bulundurabildigi gibi çizilen kavramsal modellerde de objektiflik getirmektedir.

VI.5 In-CM Zeki KM Gelistirme Yazilimi ve Incelenen Yazilimlarin Karsilastirilmesi

Literatür arastirmasinda incelenen kavramsal model gelistirme yazilimlari Bölüm II.2.3 içerisinde anlatildi. Bu yazilimler :

- ICOM : Intelligent Conceptual Modeling

- TCM : Toolkit for Conceptual Modeling

Ancak temel olarak bu yazılımlar ile bu tez çalışması kapsamında geliştirilen yazılım arasında amaç farkı vardır. ICOM veri tabanları tasarımı, TCM ise yazılım sistemleri tasarımı kullanılmayı amaçlamaktadır. In-CM kavramsal model geliştirmede disiplinler arası bir yazılım olmayı amaçlamaktadır.

ICOM, ER diyagramlarını sadece desteklemektedir. UML desteği için çalışmalar devam etmektedir. Ancak hali hazırda bu destek mevcut değildir. TCM ise ER diyagramları, Veri ve Olay akis diyagramları, durum geçiş diyagramları ve UML diyagramlarını desteklemektedir. In-CM yazılımında ise, “*disiplinler arası olmak*” amacıyla dolayı kendisine ait grafiksel gösterimleri kullanır. Ancak bu grafiksel gösterimlerde geçmişe bağlılık vardır. Yani literatür içerisinde kavramlar için kullanılan grafikler aynen alınmıştır.

ICOM ve In-CM yapay zeka’ya dayalı bir yapıya sahiptirler. TCM ise sadece kavramsal modellerin çizilebildiği bir grafik editördür. In-CM ve TCM tek bir uygulama olarak çalışabilir iken, ICOM ise sunucu-istemci (server-client) yapısında tasarlanmıştır. İstemci olan yazılım, veri tabanı tasarımı yapılan, TCM gibi, grafiksel bir editördür.

TCM, Unix platformlarında çalışır. In-CM Windows platformları için tasarlanmıştır. ICOM ise hem Unix hem de Windows platformlarında çalışan sürümlere sahiptir.

Yapay zeka’ya dayalı olan In-CM’de Bilgi tabanı editörü, Grafik editörü ve Çıkarım mekanizması ayrı ayrı uygulamalardır. ICOM’da ise kural tabanı server üzerindedir. İstemci’de geliştirilen ER diyagramına tutarlılık ya da uyumluluk kontrolü için sunucuya bağlanması gereklidir. Bu nedenle kullanıcıya ait bir kural tanımlaması yapılması zordur. Oysaki In-CM yazılımında Bilgi tabanı editörü sayesinde alandan bağımsız olarak kurallar kolayca tanımlanabilmektedir. Ayrıca, In-CM yazılımında bilginin yapısal hale getirilmesi için bir kalıp olarak referans kavramsal model’den faydalanılmaktadır. Diğer yazılımlar bu tür bir kalibi içermemektedir.

In-CM üzerinde yazılım kaynak kod değiştirilmeden kuralların güncellenmesi Bilgi Tabanı Editörü ile yapılabilmektedir. Bu güncelleme ister var olan kurallara ek olarak, istenirse yeni baştan tanımlama ile sağlanmaktadır. Ayrıca bir bilgi gösterim kalibi olarak kullanılan RKM’nin de güncellenmesi ve yeniden tanımlaması da bu editör yardımı ile kolayca gerçekleştirilmektedir. ICOM ‘da ise bu güncelleme

sunucu üzerinde gerçekleştirilmesi gereklidir. Sunucu ise İngiltere’de bir server üzerindedir. Bu nedenle istemciyi kullananlar tarafından kurallarda bir güncellenmenin yapılması mümkün değildir. Sunucu üzerinde ise bu tür bir güncellenmenin nasıl yapıldığına ya da yapılıp yapılmadığına dair ulaşılan kaynaklarda bir bilgi mevcut değildir. TCM ise tamamen kaynak kod tabanlı bir yazılımdır ve bu tür güncellemelerin yapılması yoktur.

Sonuç olarak literatür incelemesinde şu sonuçlara ulaşıldı :

- Disiplinler arası yapay zekaya dayalı bir kavramsal modelleme yazılımı bulunmamaktadır.
- Su ana kadar incelenen tüm teknik ve yazılımlar alan bağımlidir.
- Mevcut teknikler ilgilenilen alanın belirli bir bölümünü modelleyebilmektedirler.

Bu nedenle

- disiplinler arası,
- ilgilenilen alanı genel olarak gösterebilen,
- kullanıcıya yol gösteren bir kavramsal model yazılımına ihtiyaç vardır.

Bu tez çalışmasında bu amaçlara ulaşabilmek için mevcut kavramsal model geliştirme sürecin güncellenerek kavramsal modellemede iki aşamalı bir yaklaşım önerilmiştir. Bu yaklaşım temel alınarak yazılım geliştirilmiştir. Yazılımın ana yapısını genel alan bilgisinin tutulduğu referans model oluşturmaktadır. İleri bölümlerde bu konular sıra ile anlatılacaktır.

Tablo VI. 2 In-CM Zeki Kavramsal Model Gelistirme Yazilim Araci'nin incelenen çalismalar ile karsilastirilmesi.

	In-CM	Önceki Çalismalar		
		ICOM	TCM	
Yapay Zeka	Evet	Evet	Hayir	
Uygulama Alani	Disiplinler Arasi	Veri Tabanlari	Yazilim Sistemleri	
Amaç	Disiplinler arasi, yapay zekaya dayali bir kavramsal modelleme yapmak.	Veri tabani modellemesinde ER diyagramlarindaki kullanıcı belirtimlerini mantıksal olarak tamamlamak .	Yazilim sistemlerinin diyagram, ağaç yada tablolar seklinde kavramsal modellerinin olusturulmasini saglayan bir yazilim gelistirmek.	
Bilgi Gösterimi	Kural Tabanlı	Kural Tabanlı	-	
Yazilim Kaynak Kodu Degistirilmeden Bilgi Güncellemesi	Evet	Belli degil.	-	
Kullanici Yardimi	Var	Var	Var	
Bakim ve Güncelleme	Kolay	Zor.	Zor.	
Yapi	Bilgi Tabani Editörü, Çıkarım Mekanizmasi ve Grafik Editörü ayrı ayrı.	Istemci-Sunucu olarak çalışmakta. Olusturulan ER diyagramini uyumluluk kontrolü için İngiltere'deki Sunucu'ya baglanilmasi gereklidir.	Unix sistemler üzerinde tek bir uygulama olarak çalışmakta.	
Kullanabilirlik	Kolay	Orta	Zor	
Kendi Kendine öğrenim	Yok	Yok	Yok	
Grafiksel Gösterim	Kendisine ait grafiksel gösterimi var.	ER diyagramlarini kullanmakta.	Eleman Iliski Modellemesi, Veri ve Olay akis modellemesi, durum geçis modellemesi ve UML	
Referans Model Kullanimi	Var. Bilginin yapisal hale getirilmesi için Referans Kavramsal Model kullanilir.	Yok.	Yok.	

BÖLÜM VII

SONUÇLAR

Kavramsal modelleme, özellikle bilgi ve yazılım mühendisliğinde kullanılan ve gittikçe önemini arttıran bir eylemdir. Literatür incelendiğinde görüldü ki, kavramsal modellerin kullanımı sadece yazılım ve bilgi mühendisliği ile sınırlı değildir. Sosyal bilimlerden, fen bilimlerine kadar değişik disiplinler içerisinde kavramsal model uygulamalarına rastlamak mümkündür [örnekler için bkz.: Wiegand, 1999, ICOM/CIDOC, 1999, Wand ve Weber 2002, Berild, 2005, Heckel ve Guo 2004, Kunii ve dgr. 2003, URL.4, 2006, Storey, 2005, Guru & Savory, 2004, Bochicchio & Fiore, 2005]. Her disiplin, kavramsal modellemeyi kendi açısından ele alır ve kendilerine ait teknikler ile kavramsal modelleri geliştirir.

Bu çalışmada, yapay zekaya dayalı, disiplinler arası kavramsal model geliştiren bir yazılım geliştirilmiştir. Bu nedenle öncelikle kavramsal modellemenin ne olduğu, nasıl ve hangi teknikler ile gerçekleştirildiği incelenmiştir. Bu inceleme sonucunda kavramsal model geliştirme için kullanılan tekniklerin, icrâ edilecek olan hedef yönelim olduğu görülmüştür. Yani bu teknikler, problemin anlaşılmasından daha çok, problemin nasıl çözüme ulaştırılacağı, ya da çözümün yapılacağı ortamda bu işin nasıl yapılacağını ifade etmektedir. Tekniklerin bu kısıtından hareketle, bu çalışmada üç katmanlı bir problem çözüm uzayı önerilmiştir. Bu uzaya göre, bir problem tanımlanır, çözüm ve gerçekleştirme uzayları içerisinde sıra ile ele alınır. Tanım uzayında problemin anlaşılmasına odaklanılır ve problem analiz edilir. Bu analizin sonucunda problemin “genel kavramsal modeli” geliştirilir. Genel kavramsal model problemin ayrıştırılmasını gösteren genel bir yapıdır. Çözüm uzayında ise, problemin ne şekilde çözümleneceği odak noktası olarak alınır. Çözüm uzayının girdisi bu çalışma kapsamında tanımlanan genel kavramsal modeldir. Çıktısı ise problemin disiplinine uygun olan geleneksel ya da gelişmiş kavramsal model teknikleri kullanılarak oluşturulmuş olan kavramsal modellerdir. Daha sonra, problemin çözüme kavuşturulacağı sistem gerçekleştirme uzayında gerçekleştirilir, test edilir ve uygulamaya alınır. Bu çalışmanın amaçlarından biri olan disiplinler arası kavramsal model geliştirmek için, önerilen problem çözüm uzayı yaklaşımları temel alınarak mevcut kavramsal modelleme süreci yeniden güncellenmiştir. Bu süreç “In-CM Kavramsal Modelleme Süreci” olarak isimlendirilmiştir.

Literatür incelemesi sırasında ulaşılan bir diğer sonuç da, mevcut kavramsal modelleme teknikleri ile problem alanı tek bir kavramsal model ile ifade edilememesidir. Bu çalışmada önerilen “Tanım Odaklı Modelleme” ve “Çözüm Ortamı Odaklı Modelleme” yaklaşımları ile bu kısıt ortadan kaldırılmıştır. Tanım odaklı modelleme ile problemin ya da sistemin genel kavramsal modeli ortaya konuldu. Bu model ile ilgilenilen alanın tam ve doğru olarak anlaşılması amaçlandı. Bu adım, problem hangi disiplin içerisinde yer alırsa alsın genel olarak aynı yaklaşıma sahip olmasından dolayı tek bir kavramsal model ile problemin genel yapısı ve ayrıştırılması ortaya konulabildi. Bu amaç için problem alanının tüm detaylarını içerecek bir model oluşturuldu. Bu çalışmada bu amaçla önerilen referans kavramsal model (RKM) disipline ait en genel bilgileri içeren bir meta model olarak tanımlanmıştır. RKM yapısı, aynı zamanda, bir bilgi modeli kalibi (pattern) olarak da önerilmiştir.

Disiplinler arası kavramsal model olan “ Genel Kavramsal Model” bu çalışma kapsamında önerilmiştir. Sistemlerin genel bir ayrıştırmasını kavramsal model olarak gösteren bir yapıdır. Genel Kavramsal Model içerisinde sistemlerin kavramsal modeli, Amaç, Disiplin, Altyapı, Aktör, Süreç ve Görev olmak üzere altı alt alandan oluşmaktadır.

Diğer bir yönden, kavramsal model kullanmak ile sistem geliştirmelerde hata oranı düşmekte, sistemler daha kısa sürede hızlı bir şekilde tasarlanıp, gerçekleştirilmeye geçilebilmekte, geliştirilen sistemlerin doğrulanması tam olarak yapılabilmekte, müşteri sonuçta istediği ürüne tam olarak kavuşabilmektedir. Kavramsal model geliştirmeye, yapay zekanın destek vermesi ile sistem geliştirmede, hata oranı daha da azalması, daha güvenilir ve doğru sistemlerin geliştirilmesi sağlanabilmektedir.

Bu nedenle

- disiplinler arası,
- ilgilenilen alanı genel olarak gösterebilen,
- kullanıcıya yol gösteren bir kavramsal model oluşturma süreci bu çalışmada geliştirilmiştir.

Tanımlanan sürecin bilgisayar ortamında YZ teknikleri ile zenginleştirilmesi sağlanmış ve In-CM adı verilen uzman sistem geliştirilmiştir. In-CM genel olarak disiplinler arası kavramsal modelleme yapabilmektedir. Tezin bilimsel olarak bir

diğer katkisi da bu yazılımdır. Bu yazılım ile o alanlara ait RKM oluşturulduğu takdirde, farklı alanlarda kavramsal model geliştirilebilir. Bu yazılım üç alt modülden oluşmaktadır. In-CM Bilgi Tabanı Düzenleyici, Çıkarım Mekanizması ve In-CM Grafik Editörü geliştirilmiş olan üç modüldür.

Sonuç olarak, YZ ve KM teknolojileri birlikte kullanarak bu tez kapsamında:

- Bilinen kavramsal modelleme sürecini güncellenmiş,
- Problemin çözüm yönteminden çok içeriği ile ilgilenecek bir kavramsal modelleme yaklaşımı geliştirilmiş,
- Hızlı, doğru ve güvenilir kavramsal modeller oluşturmayı sağlayacak bir referans model tanımlanması gerçekleştirilmiş,
- Kavramsal modeli farklı açılardan gösterecek grafiksel gösterim yöntemi tanımlanmış,
- Yapay zeka teknikleri ile kavramsal modelleme sürecini mümkün olduğu kadar otomatik olarak geliştirecek bir sistem geliştirilmiştir.
- Geliştirilen sistem eğitim alanında örneklendirilerek sistemin işlevliği gösterilmiştir.

Gerçekleştirilen bu tez çalışması amaçından tanımlanan hedeflere ulaşarak tamamlanmıştır.

BÖLÜM VIII

TARTISMA VE DEGERLENDIRME

Gelistirilen In –CM: Zeki Kavramsal Model Gelistirme Araci ile sistem gelistirmelerinde kullanılan ve günümüzün önemli bir problemi olan, daha hizli,ve dogru bir kavramsal model gelistirimi saglanabilmektedir.

Ancak, In-CM Kavramsal Model Gelistirme Süreci’ne göre bu yazilimin çiktisi “Genel Kavramsal Model”’dir. Bu çalisma içerisinde “Çözüm Odakli Modelleme” üzerine çalisilmamistir. Bunu takiben, bu yazilimin çiktisi olan “Genel Kavramsal Model”, çözüm odakli modelleme ile alan bagimli kavramsal modellere çeviren bir çalisma gerçeklestirilebilir.

Çözüm Odakli Modelleme gerçeklestirmek üzere yapay zeka’ya dayali bir yazilim çalismasi gerçeklestirilebilir.

Tanimlanan süreç temel alinarak, farkli yapay zeka teknikleri kullanilarak, farkli kavramsal model gelistirme yazilimleri gelistirilebilir.

Bu çalismada gelistirilen ve Tanim Odakli Modelle yapan yazilim direk olarak kullanıcı isteklerini girdi Çikarim mekanizmasi üzerinden girdi olarak almaktadır. Grafik gösterimi olusan Genel Kavramsal Modeli göstermektedir. Grafik gösterim üzerinden kullanıcı girdi verememektedir. Bu çalisma kapsaminda önerilen süreç temel alinarak sadece “Grafik Gösterimi” ile kavramsal model olusturan ve yapay zekaya dayali, kullanıcıyi yönlendirebilen, tutarsizliklari tespit edebilen bir yazili çalismasi gerçeklestirilebilir.

Ayrica, Referans Kavramsal Model yapisi oldukça esnek bir bilgi modeli kalibidir. Bu kalip temel alinarak farkli alanlarda, uygulamalari gerçeklestirilebilir.

KAYNAKLAR

- [1] Andrade, J.; Ares, J.; Garcia, R.; Pazos, J.; Rodriguez, S.; Silva, A.; 2004, “A methodological framework for viewpoint-oriented conceptual modeling”, *Software Engineering, IEEE Transactions on* Volume 30, Issue 5, May 2004 Page(s):282 – 294,
- [2] Arrington, C.T., “Enterprise Java with UML”, Wiley Computer Publishing, 2001, S.2-3,.
- [3] Berild, S., “ Conceptual Modeling: Some thoughts 1”, *Technical Report*, Santa Anna IT Research Institute AB, Mayis **2005**.
- [4] Bochicchio, M.; Fiore, N.; 2005, “Supporting the conceptual modeling of Web applications: the MODE project, *Advanced Information Networking and Applications*, 2005. AINA 2005. 19th International Conference on Volume 1, 28-30 March 2005 Page(s):495 - 500 vol.1.
- [5] Booker, G., “Requirements Engineering and Management”, URL: <http://users.snip.net/~gbooker/INFO627/info627.htm> ,**2003**, Erisim Tarihi:Ocak 2006
- [6] Burg, J.F.M., van de Riet, R.P., “Analyzing Informal Requirements Specifications : A first step towards conceptual modeling”, *The proceedings of the Second International Workshop on Applications of Natural Language to Information Systems (NLDB'96)*, Amsterdam,**1996**.
- [7] Chen, P.P., March, “The Entity-Relationship Model-Toward a Unified View of Data”, *ACM Transactions on Database Systems*, Vol.1, No.1, **1976**, S. 9-36.
- [8] Coad, P. and Yourdon, E., “Object-Oriented Analysis”, Prentice Hall, **1991**.
- [9] Dachchour, M., Pirotte, A., Zimanyi, E., “Materialization and its Metaclass Impementation”, *IEEE Trans. on Knowledge and Data Engineering*, September **2002**, 14(5):1078-1094.
- [10] Dehne, F., Wieringa, R.J. and Zandschulp, H.R., “Toolkit for conceptual modeling, design and implementation.”, Technical report, Faculty of Computer Science, University of Twente, **2000**, E.Adresi: <http://www.cs.utwente.nl/~tcm>, E.Tarihi:Subat **2006**.
- [11] Dieste, O., Juristo, N., Moreno, A.N., Pazos, J., Sierra, A., “Conceptual Modeling in Software Engineering and Knowledge Engineering: Concepts,

- Techniques, and Trends”, *Handbook of Software Engineering and Knowledge Engineering*, Vol.0, No.0, World Scientific Publishing Company, **2000**.
- [12] Fowler, M., Scoot, K., “UML Distilled Second Edition, A brief guide to the standard object modelin kanguage”, Addison-Wesley, **2001**.
 - [13] Gomez, J, Cachero, C.; Pastor, O., “Extending a Conceptual Modelling Approach to Web Application Design”, 12 th International Conference on Advanced Information Systems. Springer-Verlag. Lecture Notes in Computer Science, 06 **2000**.
 - [14] Guru, A., Savory, P.,2004, “A template-based conceptual modeling infrastructure for simulation of physical security systems”, Simulation Conference, 2004. Proceedings of the 2004 Winter, Volume 1, 5-8 Dec. 2004.
 - [15] Gürbüz,A.,Can,B,Öztemel,E., In-CM Conceptual Modeling Process: A New Approach, INISTA 2005, **2005**.
 - [16] Hay, D.C., “A Comparison of Data Modeling Techniques”, Essential Strategies, Inc., October **1999**.
 - [17] Heckel, R., Guo, P. “Conceptual modeling of styles formobile systems: A layered approach based on graph trans-formation”, *In Working IFIP Conference on Mobile Information Systems(MOBIS)*, Oslo, Norway, **2004**.
 - [18] ICOM/CIDOC, “Definition of the CIDOC object-oriented Conceptual Reference Model”, ICOM/CIDOC Documentation Standards Group.**1999**, Erisim Adresi : <http://www.icom.org.br/cidoc2002/comunicacoes/htm/comunicacoes/liragouveiame nezes.htm>, Erisim Tarihi: Ocak 2006.
 - [19] Jackson,M., “Software Requirements and Specifications. A lexicon Practice”, Addison-Wesley, **1995**.
 - [20] Jackson,M., “Will there ever be software engineering”, IEEE Software, vol. 15, no.1, **1998**.
 - [21] Khatri, V.; Ram, S.; Snodgrass, R.T., 2004, “Augmenting a conceptual model with geospatiotemporal annotations”, Knowledge and Data Engineering, IEEE Transactions on Volume 16, Issue 11, Nov. 2004 Page(s):1324 – 1338.
 - [22] Kunii, T.L., Ibusuki, M., Pasko, G.I, Pasko, A.A., Terasaki, E., Hanaizumi, H., “Modeling of Conceptual Multiresolution Analysis by an Incrementally Modular Abstraction Hierarchy”, IEICE TRANS. INF. & SYST., VOL. E86- D, No.7, pp. 1181-1190, Temmuz **2003**.
 - [23] Lafore, R., Data Structures & Algorithms in JAVA, Second Edition, SAM

Publishing, **2003**.

- [24] Lamsweerde,A.,Dardenne,A.,Dubisy,F., “The KAOS Project:Knowledge Acquisition in Automated Specification of Software”, *Proceedings of AAAI Spring Symposium Series*, March **1991**.
- [25] Lommens,K., “Mapping Conceptual Schema’s to XML Document Type Definitions”, *Doktora Tezi*, Vrije Universiteit Brussel Faculteit Wetenschappen Departement Informatica, **2001**, ss.9-12.
- [26] Luger, G. F. and Stubblefield, W.A., Artificial Intelligence and the Design of Expert Systems, Benjamin Cummings Publishing Com. Inc, **1989**.
- [27] Manley, P., Zielinski, W.J., Stuart, M.C., ve dgr., “Monitoring Ecosystems in Nevada: Conceptual Model Foundation”, *Environmental Monitoring and Assessment* 64: 139-152, Kluwer Academic Publishers, Netherlands, **2000**.
- [28] Meersman , R.A., Weigand,H., “Conceptual Modeling And The Lexicon”, Jeroen Hoppenbrouwers, **1997**.
- [29] Muth, T., Modeling Telecom Networks and Systems Architec-ture: Conceptual Tools and Formal Methods, Springer-Verlag, **2001**.
- [30] Mylopoulos, J,”Conceptual Modeling for Knowledge Management: A Tutorial”, University of Toronto University of Toronto,City University of Hong Kong City University of Hong Kong, **2001**, Erisim Adresi: <http://www.comp.polyu.edu.hk/ds9/john.pdf> , Erisim Tarihi: Aralik **2005**.
- [31] Mylopoulos, J, Borgida, A., JArke, M., Koubarakis, M., “TELOS: Reprensting knowledge about information systems”, *ACM Transactions on Office Information Systems*, vol.8, no.4, **1990**.
- [32] Negnevitsky, M., “Artifial Intelligence: A guide to Intelligent Systems”, Second Edition, Addison Wesley, **2005**.
- [33] Öztemel, E., Istanbul, Yapay Sinir Aglari, Papatya Yayıncılık, **2003**.
- [34] Öztemel, E., Öztürk, V., Soyer, B. S., Öztürk, S., Detsis, G., Gonidakis, V., zu Drewer, P. M. and Greiwe, K., AI Support for SEDEP and Requirements for intelligent tool support, *RTP 11.13 Project Technical Report*, **2001**.
- [35] Pace, D.K., “Simulation conceptual model development”, *Simulation Interoperability Workshop (SIW)*, Spring **2000**.
- [36] Pace, D.K., Fall, “Development and documentation of a simulation conceptual

- model”, *Simulation Interoperability Workshop (SIW)*, Fall **1999**.
- [37] Presman, R. S., “Software Engineering A Practitioner’s Approach”, McGraw-Hill International Editions, **1992**, s.769.
 - [38] Protte,A., Zimanyi,E., Massart,D., Yakusheva,T., “Materialization :a powerful and ubiquitous abstraction pattern”, *Proceeding oh the 20th VLDB Conference Santiago, Chile*, **1994**.
 - [39] Rich E., Knight K., “Artificial Intelligence”, International Edition, McGraw Hill, Inc., Singapore, **1991**.
 - [40] Russell, S. and Norvig, P., *Artificial Intelligence: A Modern Approach*, Prentice Hall,**1995**.
 - [41] Shekhar, S., M. Coyle, B. Goyal, D. Liu, and S. Sarkar, “Data Models in Geographic Information Systems”, *Communication of ACM*, Vol. 40, No.4, April **1997**, pp: 103-111.
 - [42] Sommerville, I., 2001, “Software Engineering”,Addison Wesley, 6th Edition, **2001**, s.45,.
 - [43] Storey, V.C.; 2005, “Comparing relationships in conceptual modeling: mapping to semantic classifications”, *Knowledge and Data Engineering*, IEEE Transactions on Volume 17, Issue 11, Nov. 2005 s. 1478 - 1489
 - [44] TDK, Türk Dil Güncel Türkçe Sözlük, E.Adresi <http://www.tdk.gov.tr/tdksozluk/sozara.htm> , **2006**, E.Tarihi: Subat 2006
 - [45] URL.1, 2002, <http://www.esri.com/software/arcview/extensions/modelbuilder-whatis.html>, **2002**, Erisim Tarihi: Aralık 2002.
 - [46] URL.2, “Modeling Data, Three Level Architecture Enterprise Views, Where to begin data modelling”, Erisim Adresi: <http://www.cis.gsu.edu/~dtruex/courses/cis330/cis330PDF/wk3/DataMdlg.pdf>, **2006**, Erisim Tarihi: Subat 2006.
 - [47] URL.3, Erisim Adresi: http://www.ee.ualberta.ca/~wjoerg/SE/SE96/F_Note/F_What/F_reqa/F_smod/F_cm od/dfdm.htm#top, **2006**, Erisim Tarihi: Ocak, 2006.
 - [48] URL4, <http://www.spiritsoftworks.com/process/conceptual-modeling.htm>, Erisim Tarihi: Subat 2006.
 - [49] U.S. Department of Defense, December, "DoD Modeling and Simulation (M&S) Glossary", DoD 5000.59-M, **1998**.

- [50] Yu,E., Modelling strategic relationships for process reengineering”, *PhD Dissertation*, Dept. Of Computer Science, Univ. Of Toronto, **1995**.
- [51] Wand, Y., Weber, R., “Research Commentary: Information Systems and Conceptual Modeling—A Research Agenda”, ISSN: 1047-7047, Volume: 13, Issue: 4, December **2002**, Page(s): 363-376
- [52] Wiegand, N., “Advanced OODBMS Conceptual Modeling for GIS and other Applications”, Civil & Environmental Engineering Department & Land Information and Computer Graphics Facility University of Wisconsin-Madison, **1999**.
- [53] Worboy, M.F., “Object-Oriented Approaches to Geo-Referenced Information,” *Int. J. Geographical Information System*, Vol 8, No.4, **1994**.
- [54] Zhang, B., Chen, Y., 2005, “Enhancing UML Conceptual Modeling through the Use of Virtual Reality”, *System Sciences*, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on 03-06 Jan. 2005 s.11b - 11b.

EK-A

A.1 In-CM Ana Program Tanitimi

In-CM Zeki Kavramsal Model Gelistirme Araci bir ana programdan ulasilan üç programdan olusan bir yapidir. Ana program Sekil A-1’de görölmektedir. Bu program Microsoft .NET 2003 C# yazilimi kullanilarak yazilmistir.



Sekil A- 1. In-CM Ana Programi genel görünümü.

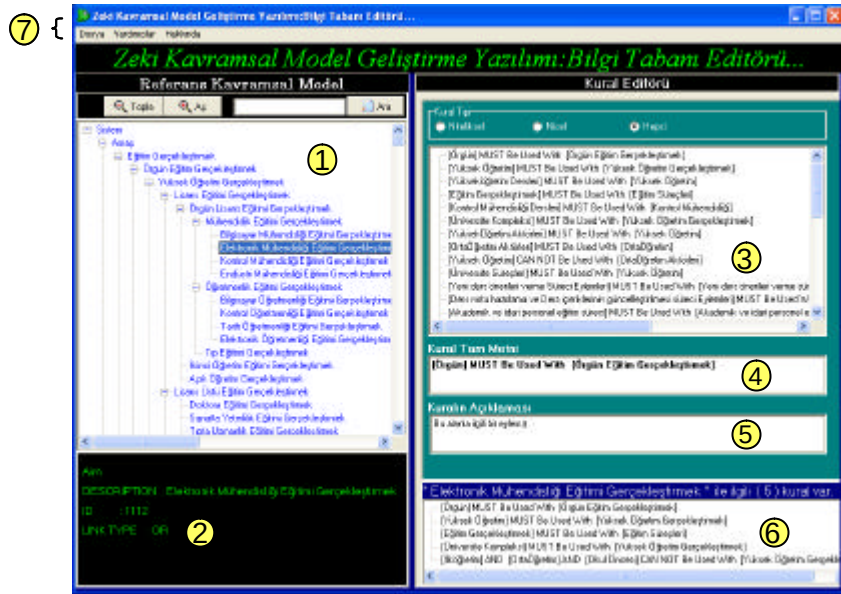
A.2 In-CM Bilgi Tabanı Editörü Tanitimi

Bilgi Tabanı Editörü (BTE) çalıştırıldığında Sekil A-2’de görülen bos ekran karsimiza gelecektir. Bu ekranda “Bilgi Tabanı Editörü”’nün hangi dilde çalıştırılacağı seçilir. In-CM yazılımı su an için İngilizce ve Türkçe dillerini desteklemektedir.



Sekil A- 2. In-CM’de BTE dil seçimi görünümü.

Dil seçildikten sonra “Tamam” butonuna basılarak Sekil A-3’deki ana ekran görüntülenir.



Sekil A- 3. In-CM BTE ana ekran görünümü.

Ana ekran 7 bölümden oluşmaktadır. Sekil A-3 üzerinde numaralandırılmış olan bu bölümler:

1. Referans Kavramsal Modelin (RKM) gösterildiği alandır. Bu alan üzerinden farenin sağ butonu ile açılan menüden RKM üzerine eleman ekleme ya da eleman çıkarma yapılır.
2. RKM üzerinde bulunulan elemanla ilgili bilgilerin gösterildiği alandır.
3. RKM ile ilgili tanımlanmış olan ve Kural Bilgi Tabanında bulunan niteliksel, nicel ya da tüm kuralların aynı anda gösterildiği bölümdür. Bu bölüm “Kurallar” etiketi altında gösterilmektedir. Bu bölüm üzerinden farenin sağ butonu ile açılan menüden yeni kural ekleme ya da çıkarma gerçekleştirilir.
4. Kurallar bölümünde seçilen kuralın tam metninin gösterildiği alandır.
5. Seçili kural ile ilgili varsa açıklama metninin gösterildiği alandır.
6. RKM üzerinde seçili olan eleman ile ilgili Kural Bilgi Tabanında bulunan kuralların gösterildiği alandır.
7. Menü çubuğu : Burası Dosya, Yardımcılar ve Hakkında seçeneklerinden oluşur. Her bir öğe kendisi ile ilgili alt menü öğelerine sahiptir.

A.2.1 BTE’de Gerçekleştirilen İşlemler

Bilgi Tabanı Editöründe gerçekleştirilen işlemler ya RKM ya da Kurallar ile ilgilidir. Bu işlemler:

- RKM’e yeni eleman ekleme,
- RKM üzerinde var olan elemanı yeniden düzenleme,
- RKM üzerinden bir elemanı silme,
- Niteliksel kural ekleme,
- Niteliksel Kuralı Düzenleme,
- Niteliksel Kuralı Silme,
- Nicel kural ekleme,
- Nicel Kuralı Düzenleme,
- Nicel Kuralı Silme’dır.

Sıra ile bu işlemlerin nasıl yapıldığı aşağıdaki bölümlerde ekran görüntüleri ile birlikte incelenmektedir.

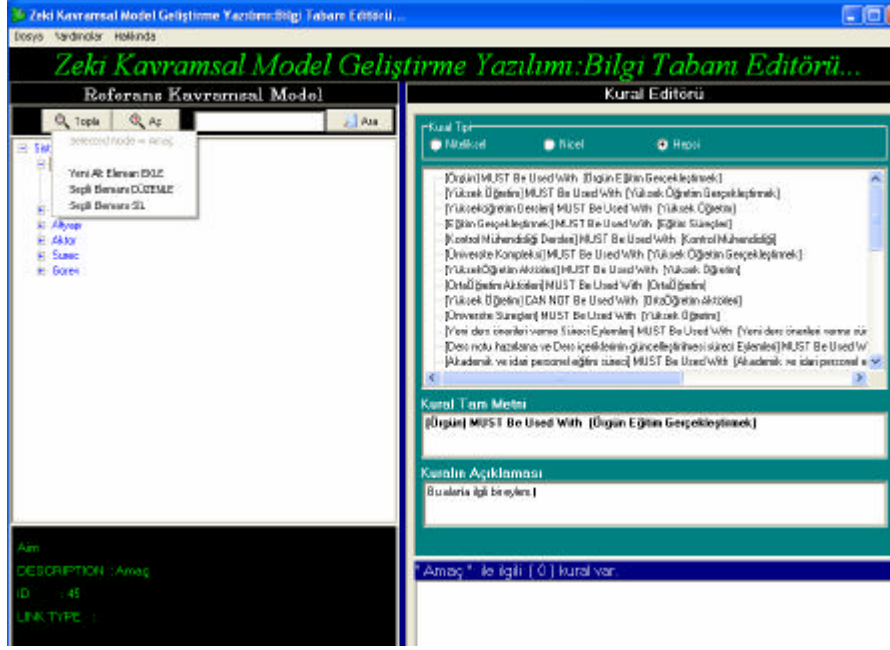
A.2.1.1 RKM’e Yeni Eleman Ekleme

RKM yüklendiğinde “Sistem” ve sistemin altı alt elemanı görülecektir. “Sistem”e alt eleman eklemek mümkün değildir. Ancak “Sistem” altında yer alan Amaç, Disiplin, AltYapı, Aktör, Süreç ve Görev ile bunların altında yer alan elemanlara alt eleman eklemek mümkündür. RKM içindeki bir elemana alt eleman eklemek için aşağıdaki adımlar sıra ile gerçekleştirilir :

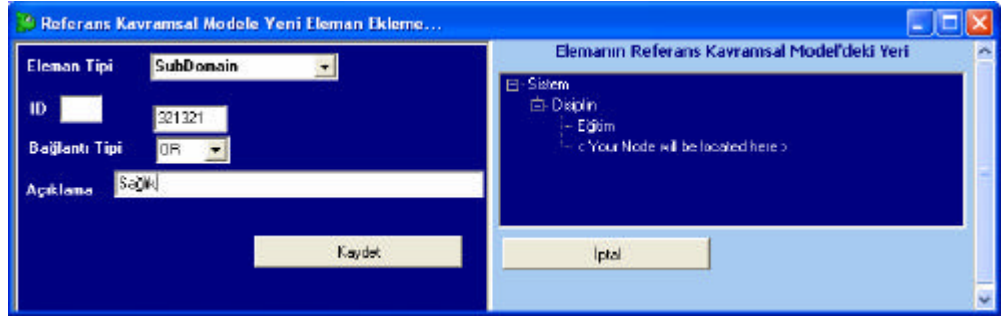
- Alt eleman eklemek istenilen eleman seçilir.
- Farenin sağ butonu tıklanır. Bu durum Şekil A-4’de görülmektedir.

Açılan menüden “Yeni Alt Eleman EKLE” seçeneği seçilir.

- Şekil A-5’de gösterilen diyalog kutusunda yer alan alanlar doldurularak “Ekle” butonuna basılır ise seçilen satıra yeni bir alt eleman eklenir.



Sekil A- 4. In-CMBTE’de RKM’e Yeni Alt Eleman Ekleme.



Sekil A- 5. RKM’e Yeni Alt Eleman Ekleme Diyalog kutusu görünümü.

“RKM’e Yeni Eleman Ekleme” diyalog kutusunda yer alan alanlar şunlardır:

- o **Eleman Tipi** : Bu alanın, eklenecek yere göre alabileceği değerler program tarafından çıkartılıyor. Örneğin, Amaç altına bir ekleme yapılacak ise bu alan “Amaç (Aim)” tipinde, Disiplin altına bir ekleme yapılıyor ise “Alt_Alan (SubDomain)” tipinde olacaktır.

- o **ID**: RKM içerisinde yer alan her bir eleman için tanımlanan ve birbirinden farklı olan bir ID (tanıtıcı, ayırt edici sayı) vardır. Bu alana bu amaçla bir sayısal değer girilir.

- o **Bağlantı Tipi** : Her bir eleman için tanımlanan bir bağlantı tipi vardır. Bu tip “ve” yada “veya” değerlerinden birini alır. Bunun anlamı, kullanıcı eğer “ve” tipine sahip bir elemanı seçer ise, çıkarım

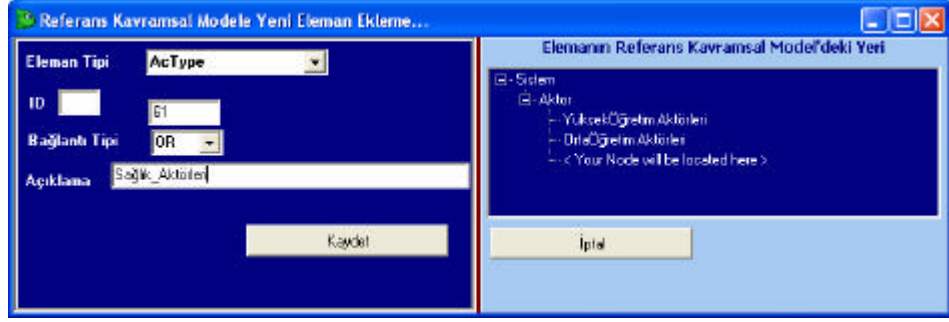
mekanizmasında o eleman ile birlikte o elemanın altında yer alan diğer tüm elemanların hepsi seçilecektir. “veya” ifadesine sahip bir eleman seçilir ise o ifadenin altında yer alan elemanlardan en az bir tanesinin seçili olması çıkarım için yeterli olacaktır.

- o **Açıklama:** Eleman için gösterilecek olan ifade buraya girilir.
- o **Eleman’ın RKM’deki Yeri :** Bu alan yeni elemanın RKM içerisinde nereye ekleneceğini göstermektedir.
- o **Kaydet :** Girilen elemanın RKM içerisine kaydeder ve diyalog kutusunu kapatır.
- o **İptal :** RKM’e Yeni Eleman Ekleme diyalog kutusunu girilen elemanı kaydetmeden kapatır.

Örnek:

“Aktör” altına “Sağlık_Aktörleri” alt elemanını ekleyelim.

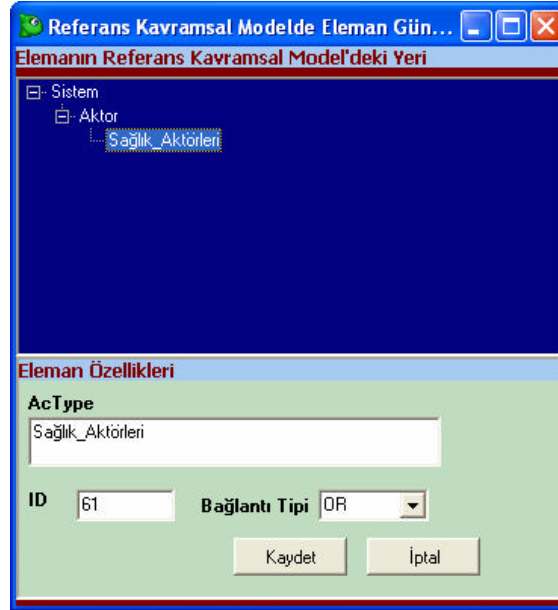
- o Öncelikle “Aktör” üzerine gelip, farenin sağ butonunu tıkladıktan sonra “Yeni Alt Eleman EKLE” ’yi seçiyoruz. Şekil A-5’de gösterilen diyalog kutusu karşımıza geliyor.
- o Bu diyalog kutusunda, “Eleman Tipi” olarak sadece “AcType” gelmektedir. Bunun sebebi “Aktör” tipi, eleman olarak sadece bu tipi kullanmaktadır. Bu nedenle bu alan içerisinde herhangi bir seçme işlemi gerçekleştirilmiyor.
- o “ID” ‘ye 61 değeri yazılır. Eğer bu sayı daha önceden girilmiş ise program tarafından bir uyarı mesajı gösterilecektir.
- o “Bağlantı Tipi” için “veya” tipini seçelim.
- o “Açıklama” olarak “Sağlık_Aktörleri” yazalım. Daha sonra “Kaydet” butonuna basalım. Yeni eleman “Eleman’ın RKM’deki Yeri” ’nde gösterilen yere kaydedilip, diyalog kutusu kapatılacaktır. Bu durum Şekil A-6’da görülmektedir.



Sekil A- 6. RKM’e Yeni Alt Eleman Ekleme örneği ekran görünümü .

A.2.1.2 RKM Üzerinde Mevcut Elemanı Yeniden Düzenleme

RKM’de var olan bir elemanın düzenlenmesinde, eleman seçilir ve farenin sağ butonu tıklanır. Sekil A-4’de görülen menüden “Seçilen Elemanı DÜZENLE” seçilir. Bu seçimden sonra Sekil A-7’de görülen diyalog kutusu açılacaktır.



Sekil A- 7. RKM’de Eleman Güncelleme diyalog kutusu görünümü .

Bu diyalog kutusu üzerinde bulunan alanların açıklaması şu şekildedir:

- o **Elemanın RKM’deki Yeri** : Bu alan düzenlenecek olan elemanın RKM içerisinde nerede yer aldığını göstermektedir.
- o **Tip** : Elemanın XML dosyası içerisinde hangi tipte olduğunu gösterir. Tek tip olduğunda bu alan değiştirilemez şekilde gelecektir.

- o **Açıklama** : Elemanın RKM içerisindeki ismidir.
- o **ID**: RKM içerisinde yer alan her bir eleman için tanımlanan ve birbirinden farklı olan tanımlayıcı sayı bu alanda görülecektir.
- o **Baglantı Tipi** : Her bir eleman için tanımlanan bir “ve” yada “veya” değerlerinden birini alan alandır.
- o **Kaydet** : Düzenlenen verileri RKM’ye kaydederek diyalog kutusunu kapatır.
- o **İptal** : Diyalog kutusunu RKM üzerinden hiçbir şey yapmadan kapatır.

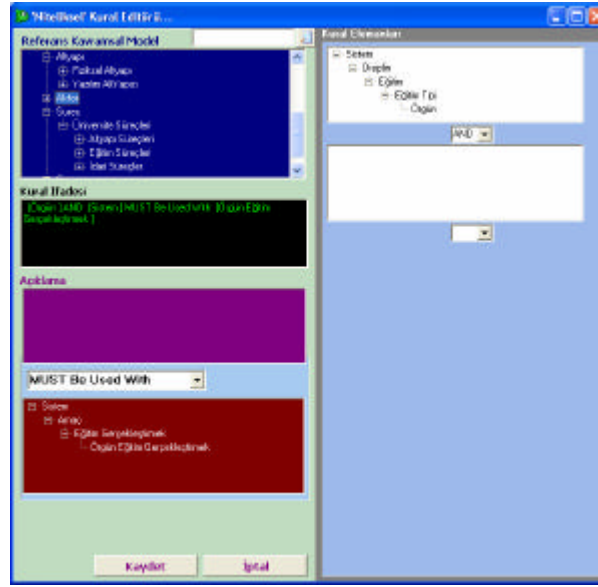
A.2.1.3 RKM Üzerinden Eleman Silme

RKM üzerinde silinmek istenen eleman seçilerek farenin sağ tusuna basılır. Şekil A-4’de görülen menüden “Seçili Elemanı SIL” seçilir. Bu seçimden sonra seçili eleman RKM üzerinden silinir. Eğer silinme sırasında bu elemana ait kurallar tabanında kurallar var ise bu kurallar da silinecektir.

A.2.1.4 Niteliksel Kural Ekleme

Niteliksel kurallar RKM içerisinde yer alan elemanların hangilerinin beraber kullanılması gerektiği ve hangilerinin de beraber kullanılmaması gerektiği ile ilgili kuralları kapsar. “Niteliksel” Kural Editörü”nü açmak için program içerisinde iki yol vardır.

- Birincisi, Şekil A-3’de 3 numara ile gösterilen alan üzerinde öncelikle “Niteliksel” Kurallar radio butonu seçildikten sonra, kuralların gösterildiği alan üzerine gelip farenin sağ butonu tıklatıldığında çıkan menüden “ Yeni Kural Ekle” seçilir ise Şekil A-8’de görülen “Yeni Kural Düzenleme” diyalog kutusu açılacaktır.
- İkinci yol ise, menü çubuğu üzerinde “Yardımcılar” tıklatılarak, buradan “Niteliksel Kural Editörü:BASLAT” seçildiğinde yine aynı editör açılacaktır.



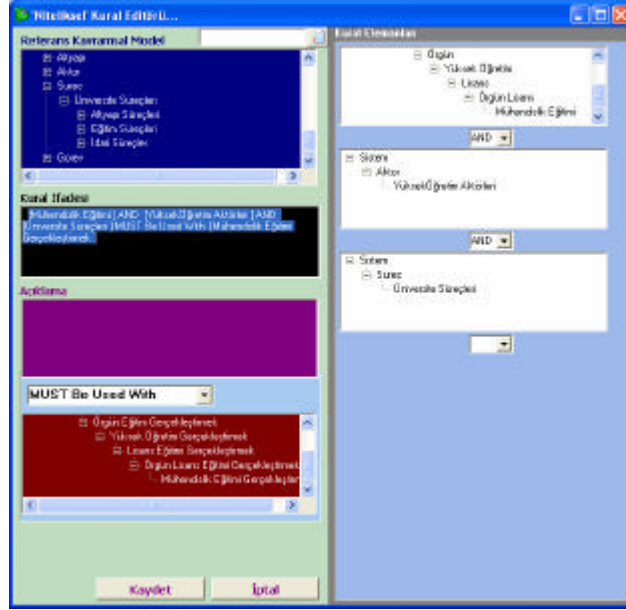
Sekil A- 8. Niteliksel Kural Editörü ekran görünümü.

Bu editör üzerinde yer alan alanları açıklamaları şu şekildedir:

- **Referans Kavramsal Model** : Bu alan içerisinde mevcut RKM gösterilir.
- **Kural Tanımı**: Sağ tarafta oluşturduğumuz kuralın metin halinde okunusu yer alır.
- **Açıklama** : Kural ile ilgili verilecek olan açıklamanın bulunduğu alandır. Bu alanın girilmesi zorunlu değildir.
- **Kural Elemanları** : Bu alanda kutulara referans kavramsal model üzerinde seçtiğimiz elemanları faremin sol butonunu basılı tutarak sürükleyip bırakarak kural içinde yer alacak olan elemanlar seçilir. Bu kutunun altında bulunan açılıp kapanır yerden de bu elemanın bir sonraki eleman ile olan “ve” yada “veya” ilişkisini belirlenir.
- **Kural Tanımı** : Bu alan içinde, kuralla ilgili olarak “birlikte kullanılmalıdır (Must be used with) ” ya da “ birlikte kullanılmamalıdır (cannot be used with)” tanımlaması gerçekleştirilir.
- **Kaydet** : Kural oluşturulduktan sonra kuralın kural bilgi tabanına eklenmesi sağlanır.
- **İptal** : Kural ile ilgili bir işlem yapılmadan bu editörün kapatılmasını sağlar.

Örneğin, Sekil A-9’da oluşturulan kural :

“ [Mühendislik Eğitimi] VE [YüksekÖğretim Aktörleri] VE [Üniversite Süreçleri] VE[Mühendislik Eğitimi Gerçekleştirmek] birlikte kullanılmalıdır”



Sekil A- 9. Niteliksel Kural Editörü ile örnek bir kural oluşturma.

A.2.1.5 Niteliksel Kurali Düzenleme

Tanımli Niteliksel kurallar’i düzenlemek için, düzenlenecek olan kural seçilir ve üzerinde farenin sag tusuna basilarak “Seçili Kurali DÜZENLE” tiklatilir. Bundan sonra Sekil A-9’da görülen Niteliksel Kural Editörü kural elemanlarini içerir sekilde açilir. Alanlar üzerinde gerekli düzenlemeler yapilarak Kaydet butonu ile kural güncellenmis olur.

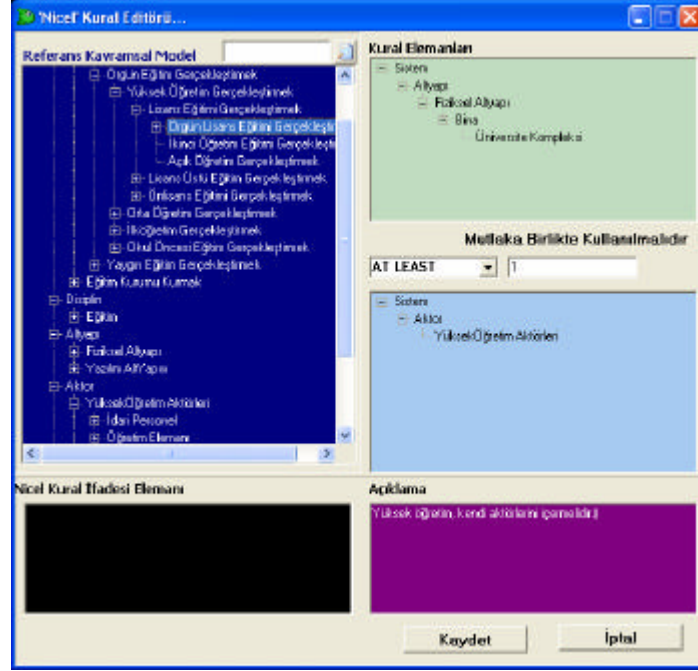
A.2.1.6 Niteliksel Kurali Silme

Tanımli Niteliksel kurallar’i silmek için, silinecek olan kural seçilir ve üzerinde farenin sag tusuna basilarak “Seçili Kurali SIL” tiklatilir. Kural, kural bilgi tabanindan silinir.

A.2.1.7 Nicel Kural Ekleme

Nicel kurallar RKM içerisinde yer alan elemanların “en az”, “en çok” yada “tamtamına” sahip olması gereken diğer elemanlar ile ilgili kuralların tanımlanması için kullanılmaktadır. “Nicel” Kural Editörü’nü başlatmak için program içerisinde iki yol vardır.

- Birincisi, A-3’de 3 numara ile gösterilen alan üzerinde öncelikle “Nicel” Kurallar radio butonu seçildikten sonra, kuralların gösterildiği alan üzerine gelip farenin sağ butonu tıklatıldığında çıkan menüden “ Yeni Kural Ekle” seçilir ise A-10’da görülen “Nicel Kural Editörü” açılacaktır.
- İkinci yol ise, menü çubuğu üzerinde “Yardimcılar” tıklatılarak, buradan “Nicel Kural Editörü: BASLAT” seçildiğinde yine editör açılacaktır.



Sekil A- 10. Nicel Kural Editörü ekran görünümü.

Bu editör üzerinde yer alan alanları açıklamaları şu şekildedir:

- **Referans Kavramsal Model** : Bu alan içerisinde mevcut RKM gösterilir.
- **Nicel Kural İfadesi**: Oluşturulan kuralın metin halinde okunusu yer alır.
- **Kural Elemanları** : Bu alanda kutulara referans kavramsal model üzerinde seçtiğimiz elemanları farenin sol butonunu basılı tutarak sürükleyip bırakarak kural içinde yer alacak olan elemanlar seçilir. Bu kutunun altında bulunan açılıp kapanır yerden de bu elemanlara ait kural tanımları gerçekleştirilir.

- **Kural Tanımı** : Bu kutunun altında bulunan açılıp kapanır yerden de bu elemanın bir sonraki eleman ile olan “en az”, “en çok” yada “tamtamına” ilişkisinden biri seçilir. Sayısal değer bu kutunun yanına girilir.

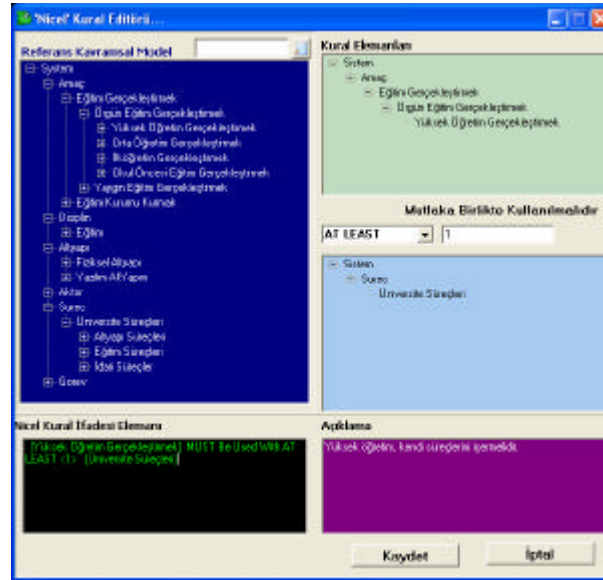
- **Açıklama** : Kural ile ilgili verilecek olan açıklamanın bulunduğu alandır. Bu alanın girilmesi zorunlu değildir.

- **Kaydet** : Kural oluşturulduktan sonra kuralın kural bilgi tabanına eklenmesi sağlanır.

- **İptal** : Kural ile ilgili bir işlem yapılmadan bu editörün kapatılmasını sağlar.

Örneğin, Şekil A-11’de oluşturulan kural :

“ [Yüksek Öğretim Gerçekleştirmek] En Az <1> [Üniversite Süreçleri] ile birlikte kullanılmalıdır.”



Şekil A- 11. Nicel Kural Editörü ile örnek bir kural oluşturma.

A.2.1.8 Nicel Kuralı Düzenleme

Tanımli Nicel kurallar’i düzenlemek için, düzenlenecek olan kural seçilir ve üzerinde farenin sağ tusuna basılarak “Seçili Kuralı DÜZENLE” tıklanılır. Bundan sonra Şekil A-10’da görülen Nicel Kural Editörü kural elemanlarını içerir şekilde açılır. Alanlar üzerinde gerekli düzenlemeler yapılarak Kaydet butonu ile kural güncellenmiş olur.

A.2.1.9 Nicel Kurali Silme

Tanimli Nicel Kurallari silmek için, silinecek olan kural seçilir ve üzerinde farenin sag tusuna basilarak “Seçili Kurali SIL” tiklatilir. Kural, kural bilgi tabanindan silinir.

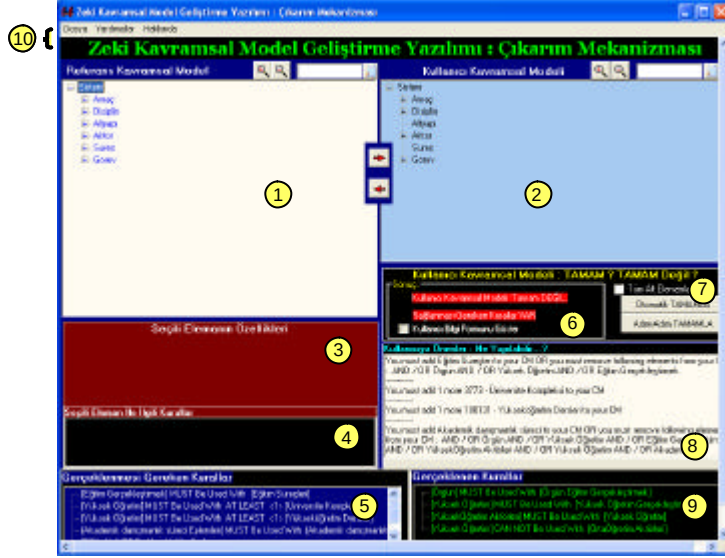
A.3 In-CM Çıkarım Mekanizmasi Tanitimi

In-CM Çıkarım Mekanizmasi çalistirildiginda Sekil A-12’de görülen bos ekran karsimiza gelecektir. Bu ekranda “In-CM Çıkarım Mekanizmasi”nin hangi dilde çalistirilacagi seçilir. In-CM Çıkarım Mekanizmasi yazilimi su an için Ingilizce ve Türkçe dillerini desteklemektedir.



Sekil A- 12. In-CM’de Çıkarım Mekanizmasi dil seçimi görünümü.

Dil seçildikten sonra “Tamam” butonuna basilarak Sekil A-13’deki ana ekran görüntülenir.



Sekil A- 13. In-CM Çıkarım Mekanizması ana ekran görünümü.

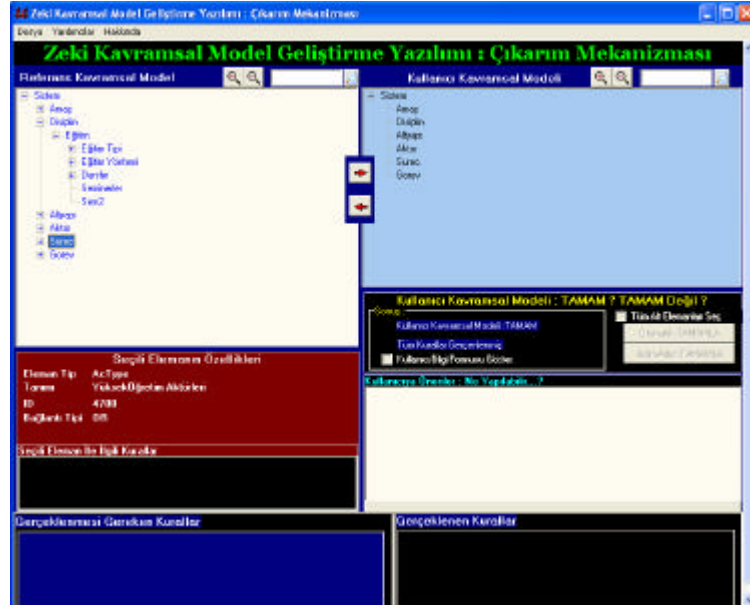
Ana ekran 10 bölümden oluşmaktadır. Sekil A-13 üzerinde numaralandırılmış olan bu bölümler:

1. Referans Kavramsal Modelin (RKM) gösterildiği alandır. Bu alan üzerinden, Kullanıcı Kavramsal Modeli üzerine eleman eklenir.
2. Kullanıcı Kavramsal Modelinin gösterildiği alandır. Kullanıcı RKM üzerinden eleman eklemek ile ya da kuralları işletmek ile ya da otomatik tamamlama ile bu alan üzerinde kavramsal model oluşturur.
3. RKM üzerinde yada Kullanıcı Kavramsal Modeli üzerinde seçili olan elemanla ilgili Tipi, Tanımı, Bağlantı Tipi gibi bilgilerin gösterildiği alandır.
4. RKM üzerinde yada Kullanıcı Kavramsal Modeli üzerinde seçili olan elemanla ilgili mevcut kuralların gösterildiği alandır.
5. Kullanıcı Kavramsal Model için gerçekleşmesi gereken alanla ilgili kuralların gösterildiği bölümdür. Bu kuralların gerçekleşmesi kullanıcıya aittir. Kurallar gerçekleşmese dahi Kullanıcı Kavramsal Modeli tam olabilir.
6. Kullanıcı Kavramsal Modelin Tamam mı? ya da Kuralların Tamam mı? olduğunda dair bilginin gösterildiği alandır.
7. Kullanıcı Kavramsal Modelinin “Otomatik” ya da “Adım Adım Tamamlama” yapıldığı bölümdür.

8. Kullaniciya Öneriler. Ne Yapilabilir ? ile ilgili Çikarim Mekanizmasini kullananlara verilen açıklamalardir.
9. Kullanici Kavramsal Modeli olusturulur iken gerçekteştirilen kuralların listesinin gösterildiği bölümdür.
10. Menü çubugu : Burasi Dosya, Yardimcilar ve Hakkinda seçeneklerinden oluşur. Her bir öge kendisi ile ilgili alt menü öğelerine sahiptir.

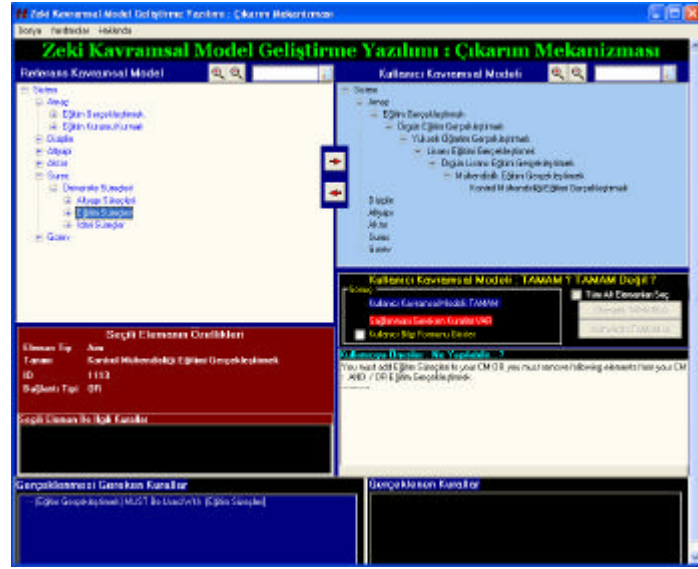
A.3.1 Çikarim Mekanizmasi'nda Gerçekteştirilen İşlemler

Çikarim Mekanizmasi üzerinde Kullanici Kavramsal Modeli olusturulur. Bunun için öncelikle Dosya Menüsünden “YENI Kullanici KM Aç” seçilir. Bu durumda Kullanici KM Sistem ve altı alt elemanı ile gösterilir. Bu asamada Kullanici Kavramsal Modeli : TAMAM'dir ve Tüm Kurallar Gerçeklenmistir. Bu durum Sekil A-14'de görülmektedir.



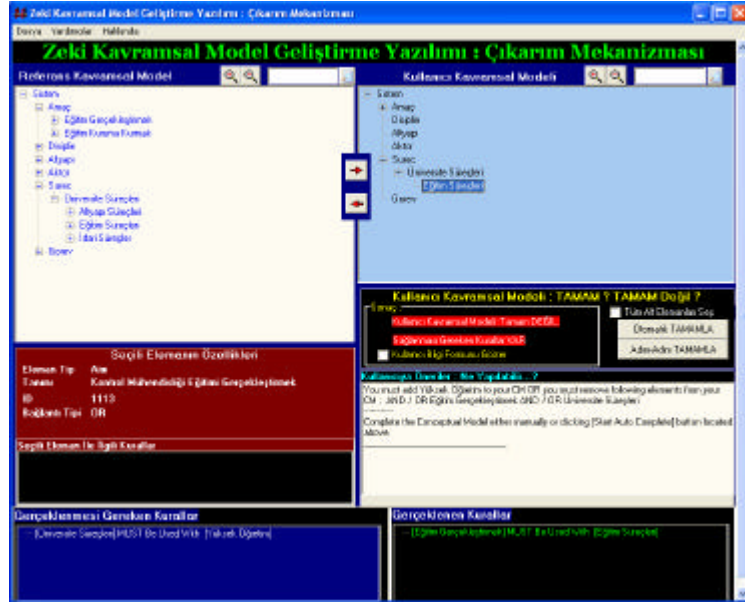
Sekil A- 14. Yeni Kullanici KM olusturulmasi.

Daha sonra kullanıcı isterleri RKM üzerinden Kullanici KM üzerine eklenir. Bu durumda Kavramsal Modelin tamliği bozulur ve gerçekleşmesi gereken kurallar oluşur. Örneğin, Kontrol Mühendisliği Eğitimi gerçeklemeyi Amaç olarak seçip, Kullanici KM üzerine eklendiğinde Sekil A-15'deki gibi Kavramsal Model tamliğini korurken gerçekleşmesi gereken kurallar ortaya çıkmaktadır.



Sekil A- 15. Kullanıcı KM’ne eleman ekleme.

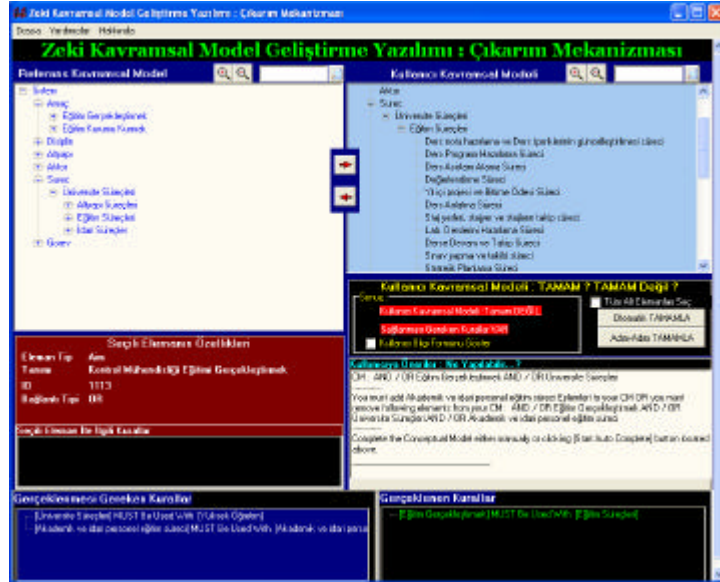
Gerçeklenmesi istenen kural yerine getirildiğinde ise Kullanıcı KM’nin tamlığı bozulmakta ve kullanıcıya bu gösterilmektedir. Sekil A-16’da bu durum görülmektedir. Bu aşamadan sonra kullanıcı KM Adım-Adım tamamlanabilir gibi Otomatik olarak da tamamlanabilir.



Sekil A- 16. Kullanıcı KM’nin tamliğinin bozulması.

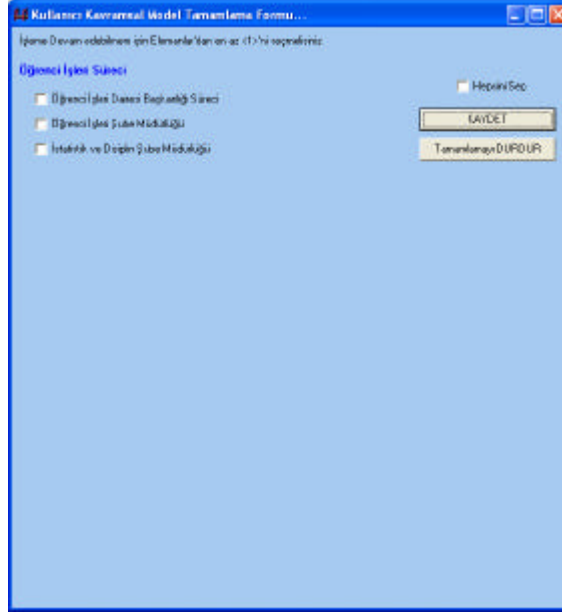
Adım-Adım Tamamlama seçildiğinde program o an aktif eleman ile ilgili olan tamamlama işlemini gerçekleştirir. Örneğin bu ekran görünümünde “Eğitim Süreçleri” aktif olan ve tamamlama işlemi bu elemana ait alt tanımlar

gerçekleştirilerek yapılıyor. Ancak, Kullanıcı KM'nin tamlığı gerçekleşmeyebilir. Bunun nedeni Tamlik için eklenen her bir elemanın yeni elemanları gerektirmesidir. Adım- Adım ekleme yeni gereken bu elemanları kavramsal model içerisine eklemeyecektir. Bu durum Sekil A-17'de görülmektedir.



Sekil A- 17. Adım Adım Tamamlamada Kullanıcı KM'si.

Otomatik Tamamlama seçildiğinde Çıkarım Mekanizması Kullanıcı KM'nin en üstünden başlayarak kavramsal modeli tamamlamaya çalışacaktır. Yazılım kendisinin karar veremediği durumlarda kavrar vermesi için Sekil A-18'de görülen ekran yardımı ile kullanıcıdan ek bilgi isteyecek ve kullanıcının tercihinə göre tamamlamaya devam edecektir. Bu tamamlamanın sonunda kullanıcı Kavramsal Modeli tam olsada gerçekleştirilmeyen kurallar bulunabilir. Ancak bu oluşturulan kavramsal modelini etkilemez. Kullanıcı kavramsal model tamamlamayı istediği anda kesebilir. Çıkarım mekanizmasını kapatabilir. Daha sonra açtığında yazılım en son üzerinde çalışılan model üzerinden kendini açacaktır.



Sekil A- 18. Kullanici Kavramsal Model Tamamlama Formu.

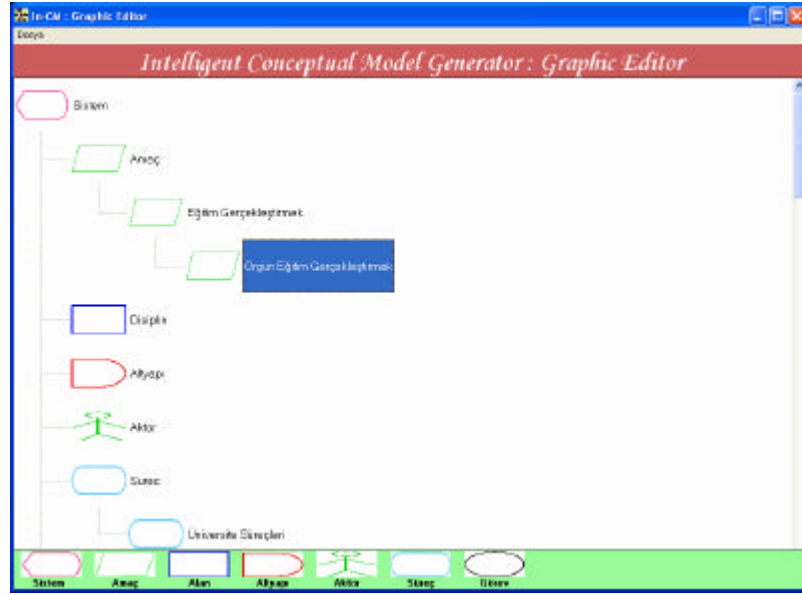
Kullanici üzerinde çalıştigi bir kavramsal modeli Dosya Menü'sü üzerinden Farkli Bir isim ile kaydedip saklayabilir. Kavramsal model tamamlamanin her asamasinda kullaniciyi yönlendiren ya da bilgilendiren diyalog kutulari program tarafından kullaniciya yol göstermektedir.

Bununla birlikte, kullanici, eger uzmanlik bilgisine sahip ise In-CM BTE'nü Yardimcilar-> Bilgi Tabani Editörü: Baslat ile baslatabilir. Gerekli kural ve eleman tanımlarini yapabilir. Daha sonra Çıkarım Mekanizmasi'ni yeni bastan baslatarak ya da Dosya-Referans KM:Yenile ile bunlari aktif hale getirebilir.

Kullanici ayni zamanda olusturdugu kavramsal modelin grafik görünümünü Yardimcilar-> In-CM Grafik Modülü: Baslat ile grafik modülünü baslatarak görebilir.

A.4 In-CM Grafik Editör Yazilimi Tanitimi

In-CM Zeki Kavramsal Model Gelistirme Aracinin bir diger modülü de In-CM Grafik Editörü'dür. Grafik Editörü, çıkarım mekanizmasi tarafından olusturulan Kullanici Kavramsal Modelini tanımlanmis olan gösterim sekillerine uygun olarak çizer. Bu modül kavramsal modelin tamligi ile ilgilenmez. Sadece grafiksel gösterim saglar. Sekil A-19'da grafik editörünün ana ekran görünümü görülmektedir.



Sekil A- 19. In-CM’de Grafik Editörü ana ekran görünümü.

ÖZGEÇMİS

1974 yılında İzmir’de doğdu. İlk, orta ve lise öğrenimini İzmir’de tamamladı. 1992 yılında İzmir Atatürk Teknik Lisesinden mezun oldu. Aynı yıl, Marmara Üniversitesi TEF’de Elektronik- Bilgisayar Eğitimi Bölümünde lisans eğitimine başladı, bir yıl yabancı dil İngilizce hazırlık eğitimi aldı, 1997 yılında mezun oldu. Aynı yıl MÜ, Fen Bilimleri Enstitüsü Elektronik-Bilgisayar Eğitimi Bölümü, Bilgisayar-Kontrol Eğitimi Programında Yüksek Lisansa başlayarak 2000 yılında bu eğitimini tamamladı. 2000 yılında aynı bölümde başladığı doktora eğitimine devam etmektedir.

1997 yılında TÜBİTAK MAM’da araştırmacı olarak işe başladı. Bu süre içerisinde iki uluslararası, biri ulusal olarak 3 projeyi başarılı bir şekilde tamamladı. Bu projelerden 1 Başarı Ödülü, 1 Tesvik Ödülü aldı. Üçüncü proje için Başarı Ödülü’ne aday gösterildi. Halen bu kurumda çalışmalarını sürdürmektedir. Evli ve 1 çocuk sahibidir.