

BÝLGÝ TABANLI BÝR YÖNETÝM SÝSTEMÝNDE KAVRAMSAL MODELLEMENÝN ÖNEMÝ

Zeynep ALTAN

Ýstanbul Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliði Bölümü
34850, Avcýlar, Ýstanbul
zaltan@istanbul.edu.tr

ABSTRACT

The development of object-oriented models have effectively been influenced by conceptual models. Many semantic data models have been developed that capture some of the meaning, as well as the structure of data. These models allow explicit representation of special semantic relationships, or abstractions. In fact, there is a relation between linguistic and conceptual representations.

Keywords: *Object oriented model, conceptual modelling, data abstractions, computational models, knowledge representation.*

ÖZET

Nesneye yönelik modellerin geliºimi kavramsal modellerden oldukça etkilenmiºtir. Anlam belirlemede kolaylýk saðlamak amacy ile, pek çok semantik (anlambilimsel) veri modelinden yararlanýlabilir. Bu modeller, özel anlamsal baðýntýlarý tanımlamakta ve aynı zamanda birtakým soyutlamalara izin vermektedir. Gerçekte, dilbilimsel ve kavramsal simgeleniºler arasýnda bir iliºki vardýr.

Anahtar Kelimeler: *Nesneye Yönelik Model, Kavramsal Modelleme, Veri Soyutlamalarý, Hesaplanabilir Modeller, Bilgi Simgeleniºi*

I. GÝRÝº

Bilindiði gibi bir yazýlým geliºtirme sistemi veriler ve bu veriler üzerine gerçekteºtirilen süreçler ºeklinde iki temel kavramdan oluºmaktadır. Bu nedenle bir yazýlým modelinin geliºtirilmesi sýrasýnda problemin ayrýºtırýlmasý veri ya da süreç faktörünün önemine göre gerçekteºtirilir. Süreçleri temel alan programlama modelleri C, Pascal gibi yapýsal programlama dillerinin birer kara kutu gibi iºleyen fonksiyonlarýný kullanýrlar. Oysa veriler temel alýnarak programlama tasarýmý yapan nesneye dayalı programlama yöntemlerinde, bilgi tabanlı sistemlerde olduðu gibi, sistemin hangi süreçleri ne tür veriler üzerinde uyguladýđý araºtırýlýr.

Gerçek dünya uygulamalarý esas olarak iki kýsýmdan oluºmaktadır:

Varlýklarýn simgeleniºinin incelendiði nesneye yönelik veri modellerinin desteklenmesi,

Mevcut nesneye-yönelik veri tabaný sistemleri ile tam olarak desteklenmeyen varlýklar arasý iliºkilerin simgelenmesi.

Nesneye yönelik veri tabanlarýnýn geliºimi hem nesneye yönelik programlama dillerinin hem de anlamsal veri modellerinin geliºmesi ile güçlenmiºtir. Aslýnda nesneye yönelik veri tabanýnýn yapýsal bileºenlerinin anlamsal veri modelleri ile ortak yönleri olmasına raðmen, iliºkilerin ele alýnýºtarzý daha çok nesneye yönelik programlama dillerdeki gibi gerçektir. Ýliºkileri nesnelere kullanarak modellemek her zaman kolay olmayabilir. Nesneye yönelik veri tabaný sistemlerinde varlýklar arasý iliºkiler:

Nesne yapýsýnýn bir parçasýný oluºturan nesne özelliklerinin simgeleniºi iç-varlýk

ilişkisi; örneğin kitap ile kitabın bağlılığı arasındaki ilişki,

Varlıklar arası ilişkilerin daha zayıf olduğu ara-varlık ilişkisi; örneğinkurum ile kurumda çalışanlar ilişkisi şeklinde nesne özelliklerine göre sınıflandırılabilir [2]. Bu iki tanım arasındaki farkı şöyle de açıklayabiliriz:

Çıç-varlık ilişkisi varlıklarla bir anlam taşıırken, ara-varlık ilişkileri varlıklardan ayrı simgelenirler.

Varlıkların öncelikli özelliklerinin ara-varlık ilişkilerinden ayrırt edilmesi nedeni ile, iç-varlık ilişkileri ile oluşturulan veri tabanı oldukça esnek; yani modülerdir.

Ara-varlık ilişkilerinde ise nesne özellikleri bazı karmaşık ilişkileri tam olarak gerçekleştirmez; örneğin yuvalanmış ilişkilerin sıradüzensel olarak örgütlenmesi nedeni ile, inlenmeleri oldukça güçtür. Bir düzeydeki bir varlık bir sonraki düzeydeki bir dizi varlık ile inlenilebilir.

Ara varlık ilişkilerinin tanımlanması ve inlenmesi SQL benzeri yüksek düzeyli çalışmalarla desteklenebilir.

Avrupa kökenli O2, Amerika kökenli GemStone ve ONTOS nesneye yönelik veri tabanı yönetim sistemleri:

1. Uygulama dilleri ve platform farklılıkları,
2. Temel veri tanımlama ve yönetim özellikleri,
3. Aktif/pasif nesnelerin kullanımı,
4. Çoklu miras ilişkisini destekleme,
5. Birleik nesne özellikleri destekleme,
6. Dinamik inleme özellikleri,
7. Veri depolama yönetimi,
8. Bildirim deitirebilme özelliği,
9. Uygunluk ve sorgu yönetimi,
10. Veri tabanı erişim hızı ve güvenlik,
11. Kullanıcı arayüzleri.

gibi özellikleri ile farklılık gösteren, C++ programlama dilinde (GemStone ilave olarak Smalltalk 'u da desteklemektedir) geliştirilmiş ticari nesneye yönelik veri tabanı yönetim sistemleridir [1].

II. KAVRAMSAL MODELLEMENİN ÖNEMİ

Nesneye yönelik bir dil gibi kavramsal modelleme dili de bir gerçek dünya varlığı ile ilgili tüm bilgiyi davranışı ile birlikte nesnenin kendisinde kapsar. Kavramsal modelleme dilleri gerçek dünya varlıklarının bilgisayar modelindeki nesneye doğrudan eşitirmesi nedeni ile, bu tür sistemlerin tasarımı ve yapma geçirmesi oldukça kolaydır. İlk nesneye yönelik standart dillerden olan Smalltalk, örnek deitkenlerinin özelliklerle yer deitirmesi ile

kısıtların bütünlüğünü ve ilişkilerin diğer nesnelere semantiğini kapsar.

İlişkisel veri modeli pek çok gerçek dünya veritabanı yönetiminde için standart olarak uygulanmaktadır. Geliştirilecek modelin, örneğin bir bilgi modelinin, veri tabanları ile varlıklar arasındaki ilişkilerin deitlerinin tutulduğu tablolara uyumlu olup olmadığı araştırılır. Bu yaklaşım, yüzyıllardan beri insanların bilgi dizilerini tutmakta kullandığı iki boyutlu tablolara çok benzeyen tek ifadelik yapıyla oldukça güçlüdür. Sonucun olumlu olması için de veri güvenilirliği ve başarı oranı yüksek uygulamaların geliştirilmesi gerekir. Bu nedenle ilişkisel veri tabanı teknolojisi karmaşık bilgi sistemlerinin isteklerini tam olarak karşılayamaz. Oysa nesneye yönelik veri tabanı tasarımı ile son derece karmaşık bilginin hızlı bir şekilde yönetilmesi mümkündür. Herhangi bir gerçek dünya varlığının tanımlanması için gerekli olan bilginin, Pascal 'da kayıtlarla ve C' de yapılarla tek bir depolama nesnesinde depolanabilmesine rağmen, bu varlığın çeitli mesajlara karşı göstereceği davranış ve ilişkilerinin diğer nesnelere üzerindeki etkileri ve kısıtlarının programın tümüne yayılması zorunluluğu vardır .

Bilindiği gibi veri tabanı kendi içinde verinin anlamı hakkında hiç bir şey saklayamaz. Bu da veri tabanı uygulamalarında sınırlama oluşturmuştur. Oysa anlambilimsel veri modeli verinin yapısının geliştirilmesini sağlayarak uygulamadaki kısıtları ortadan kaldırmıştır [7]. Anlambilimsel veri modelleri kavramsal modellemenin üst seviyesinde geliştirilir, tasarımı gerçekleştirme aşaması ile bir bağlantı yoktur. Bu tür modellemede ifade kolaylığı, basitlik, kesinlik ve simgesellik temel ilkelerdir. Anlambilimsel adımlar düğümler nesnelere simgelerken, bağlantılar veya yaylar bağlantıları simgeler. Anlambilimsel modellerin olumlu yönleri kolay anlaşılabilirliği, miras desteklemesi, bilgiyi incelemenin doğal yolu olması iken; olumsuz yönleri de istisna durumlarının modellemedeki zorluk, yordamsal bilgiyi simgelemedeki güçlük, düğümleri ve bağlantıları tanımlamak için bir standart geliştirilmemiş olmasıdır. Aslında ticari veri tabanı yönetim sistemleri anlambilimsel veri modelleme kavramları kapsamında deitir. Çünkü bu tür modelleme ile birbirinden farklı veya çelişkili yorumlar ortaya çıkabilir. Ayrıca anlambilimsel içeriğin ilişkisel bir yapıya geçirmede korunması da oldukça güçtür. Veritabanında bilginin birden fazla anlamının sunulmasını mümkün kılan anlambilimsel veri modelleri çeitli veri soyutlamaları ile geliştirilebilir.

III. NESNE MODELLEME TEKNİYDİ: VERİ SOYUTLAMALARI

Veri soyutlamaları ile gerçek dünya, varlıklar ve varlıklar arasındaki ilişkiler şeklinde çeşitli bakış açılarına göre tanımlanabilir [4].

Alt tip-üst tip ilişkilerini simgeleyen kapsama,

Yki veya daha fazla nesne arasındaki ilişkiye izin vererek daha yüksek seviyede bir nesne oluşturulan kümeleme,

Ödelerin koleksiyonunu daha yüksek seviyede bir küme olarak tanımlayan birliktelik incelenen sistemin yapısının simgeleyen nesneye yönelik modelleme metodolojisidir.

Kapsama soyutlaması genellikle "is-a" ifadesi ile gösterilir. İlişkilerde "A is-a B" dir. A özel bir varlık tipi, B ise genel nesne tipi ile ilgilidir. Kapsama soyutlaması bir çok çeşide ayrılabilir:

Sınıflandırma, varlık oluşumu ve onun uygun varlık tipi arasında bir kapsama soyutlamasıdır. Örneğin; "Ayşe sekreterdir". Varlık tipinden varlık oluşumuna doğru ters ilişki de tanımlanabilir.

Genel varlık tipi birbiriyle çakılmayan özel varlık tiplerinin birleşimi olarak tanımlandığında bu soyutlama *genelleştirme* olur. Özel varlık tipleri genel bir parçasıdır. Örneğin; öğrenci lisans ve yüksek lisans öğrencilerin genelleştirmesidir.

Özelleştirme ise genelleştirmenin tersidir. Genel varlık tipi olayları badımsız altsınıflar şeklinde görüntüleyen bazı ayırt edici karakteristikler içerir. Örneğin; mezun öğrenciler öğrencilerin özel bir halidir.

Tek bir genellemeye birden fazla özel varlık tipi ilişkilendirildiğinde, özel varlıklar arasındaki ilişkiyi kontrol etmek gerekir. Örneğin özel varlık tipi numarasının genel varlık tipi için tanımlanması gerekebilir. *Çalışanlar* isimli genel varlık tipinin; *sekreter*, *mühendis*, *yönetici*, *tam zamanlı* ve *yarı zamanlı* çalışanlar varlık tiplerini içerdiği kabul edilsin. Bunların bir grubu *tam zamanlı* ve *yarı zamanlı* çalışanlar varlıkları olarak tüm çalışanları içerirler ve örtüme oluşturmazlar. Bir başka grup ise *teknisyen*, *sekreter* ve *mühendis* olarak tüm çalışanları içermemesine rağmen örtüme de oluşturmaz. Fakat bu iki grup arasında örtüme olabilir. Örtüme olduğunda özel varlıklar arasında alt küme sıradüzeni oluşur.

Kümeleme, nesnelere arasındaki ilişkinin daha yüksek düzeyde bir nesne olarak ele alındığı soyutlamadır. Küme oluşturulurken kümeyi oluşturulan nesnelere özgül özellikleri bastırılır. Örnek olarak "rezervasyon" otel, insan ve tarihin kümeleme mi halidir. Nesnenin

kendine ait özelliği olabileceği gibi başka ilişkiler de içerebilir. Üç tür kümeleme tanımlanabilir:

Bir özellik diğer özelliklerin birleşimi olabilir. Mesela "adres" özelliği numara, cadde, posta kutusu, posta kodu özelliklerinin bir kümelemesidir.

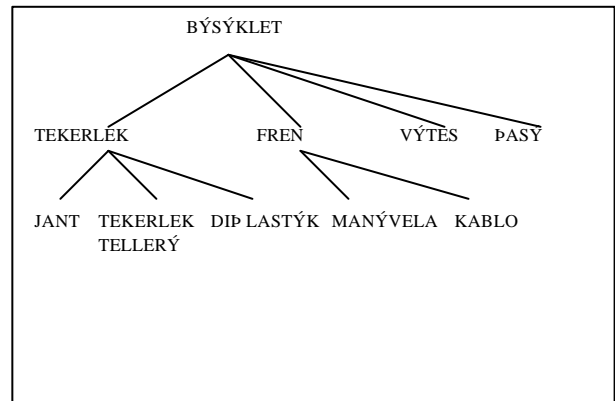
Bir varlık, varlık ve /veya özelliklerin bir topluluğu olabilir. Örneğin *komite* bir varlık olarak, bir özellik olan *isimden*, bir varlık olan *başkandan*, bir varlık olan *sekreterden*, varlık takımı olan *ödelemlerden* meydana gelebilir.

Bir ilişki, varlık ve özelliklerin bir topluluğu olabilir. Örneğin; *görev* ilişkisi, hem *çalışan* ve *proje* varlık tiplerinin hem de *başlangıç tarihi* ve *bitişi tarihi* özelliklerinin kümelemesi olabilir.

Kümelemenin bileşenleri seviyelerde tanımlanabilir. Örneğin, klima arabasının *amaca uygun* bileşeni (olması şart olmayan) iken, motor *belirleyici* (karakteristik bileşenin tek başına kümeleme oluşturması) bileşendir. Kümelemenin bütünlüğünü sağlamak için, uygun olmayan bileşenler kaldırılmalı, tüm belirleyici bileşenler tanımlanmalı, herhangi bir belirleyici bileşen sadece bir defa kullanılmalıdır.

Birliktelik, öde nesnelere ayrıntılarının bastırıldığı, küme nesnesinin özelliklerinin ön plana çıkarıldığı soyutlama tipidir. Kümeleme için tanımlanan *komite* örneğindeki *ödelik*, *kişi* varlık tipinin birlikteliğidir. Birliktelik "*komitenin ödelik bileşeni*" şeklinde tanımlandığında ilişki için de kullanılmıyabilir olur. Birlikteliğin ödelerinin tümü aynı cinsten olmalıdır. Oysa kümelemenin bileşenleri çeşitlilikle aynı cinsten değildir. Birliktelik soyutlaması ile sonlu sayıda öde kullanılabilesine rağmen, kümelemede bileşenlerin sayısı sabit olmalıdır.

Farklı gerçek dünya kavramlarının modellemede kullanıldığımız ve yukarıdaki veri soyutlamalarına göre tanımlanan nesnelere bir içerik sıradüzeni oluştururlar. Aşağıdaki şekilde en genel şekilde çoklu içerik temsil edilmektedir.



Pekil 1: Bisiklet tasarımı için içerik sıradüzeni

Yukarıdaki tasarımda her bir bileşen bir nesne olarak modellenir ve bu bileşenlerin içerikleri aynı zamanda nesne içeriklerini oluşturur. Bir simgelenişle sınıflara ait deşiklenlerin tümü tanımlanmamış olabilir. Örneğin bisiklet sınıfsı marka-ismi isimli bir deşiklen de içerebilir.

IV. KAVRAMA YÖNELİK VERİTABANLARINDA BİR SORGULAMA ÖRNEĐİ

Kavrama yönelik veri tabanlarında sorgulamanın gerçekleşmesi duruma dayalı mantık yürüten sistemlerle gerçekleştirilebilir [6]. Bu sistemler bilgiyi hafızanın veya kullanılan kütüphanenin eski durumlarından çıkartarak amaç sorgusuna uygun verinin çıkarılması için bir mekanizma oluştururlar. Belli bir veriyi arama kesin bir sorgulama olabileceđi gibi bir amaç sorgulaması da olabilir. Kesin sorgulamalar kriterlere birebir uyan veriyi hedef alır. Örneğin “Balzac’ın eserleri nelerdir?” veya “8.15 uçađı ne zaman kalkıyor?. Sorgulamayı yapan kişi başka birinin eseri veya başka bir uçađın kalkış saati ile ilgilenmemektedir. Oysa amaç sorgulamaları o soruya yakın verileri hedef olarak tanımlar. Örneğin “kroner kalp yetmezliđi riski düşük olan hastalardan yüksek kolesterolü olanları listele” . Eđer hiç yoksa orta düzeydeki kolesterolü ve düşük kroner kalp riski olan hastalar da sorgulamayı yapanı ilgilendirebilir. Tipik amaç sorgulamaları kesin olmayan deđerler içerebilir. Böyle sınıfları belirsiz sorgulama gereksinimlerine mühendislik, tıp, yönetim gibi uygulamalarda karşılaşılmaktadır. Kavramların oluşturulması ve duruma göre mantık yürütme sistemleri tıbbi tehis, yatırımların planlaması gibi bilgi yoğunluklu uygulama alanları için bilginin integrasyonunun dođru olarak sağlanması ile dođru sonuçlar üretebilir.

Duruma dayalı mantık yürütebilmek için tanımlanacak kavram modellerinin bazı özellikleri gerçekleştirilmesi gerekir. Klasik kavram modelinde kavramların gösterimi tüm sınıfsın bir özet tanımını verir. Böylece hafızada tutulması gereken bilginin az olması sağlanır. Kavramın temsil ettiđi özellikler gerekli özellikler iken, bunlar yan yana getirildiğinde yeterli özellikleri oluştururlar. Bu nedenle de bu model ile bir çok kavram için soyut özellik tanımlı yapılmaları güç olabilir. Kavramın tanımının gerekli ve yeterli özelliklerle sınıflanmadığı ve kavramın yine tüm sınıfsın özet bir tanım ile sunulduđu model, prototip kavram modelidir. Eđer kavram soyut bir özet olarak deđil de bazı örnekleri ile tanımlanıyorsa örnek kavram modeli oluşturulur[3]. Bundan sonra sunulan kavramların veritabanında modellenmesi seçilecek bir örnek tanımlama ve kavram sorgulama dilinde gerçekleştirilir. Böylece bilgi tabanlı yönetim sistemleri için duruma-yönelik mantık yürütme ile

yeni durumların yorumlanması ve yeni problemlerin çözümü elde edilir.

V. BİLGİNİN WWW ORTAMINDA SAKLANMASI

Geliştirilen herhangi bir yazılım sisteminin sonuçlarının bir veri tabanı içerisinde tutulması geleneksel bir yaklaşımdır. Oysa hem dilbilimsel hem de kavramsal desteđi olan bir programın geliştirilmesi bilgi bilgisayar dođurur. Böylece bir konu ile ilgili bilgi edinmek isteyen ve bu bilgileri düzenleyen herhangi bir kimse, çevrimiçi metinlerden pek çok bilgiyi kolaylıkla almayıabilir. Burada geliştiricinin potansiyel bir ilk kullanıcı olduđu görülür. HTML – tabanlı bir tasarım gerçekleştirildiğinde geliştirici çok çeşitli deđiyük belgeye ulaşabilir; uzman kontrolleri ve standartlara uyum ile büyük ölçekte paylaşılabilirlik sağlanabilir. Uzman kişilerle karşılaştıkı konunun sağlanması tıp, hukuk gibi tüm alana-özel terimler içeren çalışmalarda önemlidir. Böylece problemin modellenmesi için ulusal ve uluslararası düzeylerde oluşturulacak komitelerin sadece betimsel deđil tavsiye edilmiş bilgiyi sunması sağlanır. Uygun sağlanması için kavramsal analiz çok önemlidir. Yılgı alanının anlaşılabilirliđi, o alanın kavramlarının anlaşılabilirliđidir.

Tasarlanan programın yapıma geçirilmesi için kavramsal analizi destekleyen örnek metne gereksinim vardır. Böylece belirli bir özel alana ait kavramları içerecek oportünist olmayan bir yaklaşım sergileyen bir örnek metin düzenlenir. Burada yapılan işin bilgi mühendisliđi ile ortak yönleri oldukça fazladır. Fakat örnek metnin oluşumu için gerekli olan belgelerin farklı kaynaklardan sağlanabilmesi ve bunların bilgisayar tarafından okunabilir forma dönüümü gerekmesi nedeni ile, kolaylıkla tercih edilen bir yöntem olmamasına rağmen sonuçları çok deđiyüküdür.

Bilgi tabanlı bir yönetim sistemi oluşturulurken geleneksel veri tabanı yerine dilbilimsel bilgiyi de içeren kavramsal ilişkiler çeşitli sıradüzenlerde tanımlanabilir[5] . Yukarıda ayrıntılı olarak incelenmiş olan soyutlamalar burada da tanımlanır. Örneğin; genelleştirme-özelleştirme için “araba bir araç cinsidir”, konu için“cebir matematiđin alt alanıdır” ve parçası olma (kümeleme) için “bağ vücudun bir parçasıdır”. Bu aşamada geliştirici çok sayıda tanımlanabilen ayrırt edici özelliđin, çerçeve benzeri bir yapı oluşturarak kavram ile birlikteliđini kurmalıdır. Örneğin bilişim alanına yönelik bir sözlüğünün düzenlendiđini düşünelim. CD-ROM kavramı için bir kişi “depolama kapasitesi: 700 MB” şeklinde bir ayrırt edici özellik tanımlayabilir. Bir kavram genelleştirme-özelleştirme sıradüzeninin bir parçası ise, ayrırt edici özellikler otomatik olarak

genelden daha özel kavramlara miras yolu ile geçmelidir. Burada çoklu mirasın da gerçekleşmesi sağlanabilir. Çalışmanın daha sonraki aşamalarında dađıtyık olarak oluşturulmuş farklı bilgi tabanlarının birleştirilmesi ile , o konu ile ilgili araştırma yapan kişinin daha büyük bir bilgi tabanına ulaşması mümkün olabilir. Örnek olarak Avustralya'daki arabalar ile ilgili bir bilgi tabanının Kanada'daki kamyonlarla ilgili bir bilgi tabanı ile tümeltilmesi verilebilir.

Böylece; özellikle WWW ortamında belirli bir alana-yönelik belgelerin toplanması ile bu belgeler içindeki potansiyel terimlerin tanımlarının, bu terimlerin dilbilimsel özelliklerinin de mevcut olduğu kavramsal bilginin analizi ile yapılması, daha nitelikli ve daha hızlı çalışın alana-yönelik çalışmalar olarak sonuçlanacaktır.

VI. SONUÇ

Bilgi tabanlı sistemlerde yazılımın geliştirilmesi aşamalarında pek çok sorun ile karşılaşılabilir. Analiz aşamasında yazılımın hizmet edeceği unsurlar isterlere göre çok iyi belirlenmelidir. Geniş kapsamlı olarak toplanmış veriler, geliştirilecek yazılımı başarıyla sürükleyebilir. Bu nedenle de yazılımın geliştirilmesi esnasında kavramsallaştırma temel bir problem çözme aktivitesi olarak tanımlanabilir. Kavramsal modellemenin en önemli özelliđi sistemin yüksek düzeyde simgeleniğini sergilemesidir. Amaç ise, istelerden cümleler çıkararak bunları tasarım ve program aktiviteleri için uygun hale getirmektir. Veri soyutlamaları şeklinde tanımlanan özelliklerin birçok ticari veritabanı yönetim sistemi tarafından desteklenmemesi nedeni ile, pratikte uygulanabilirliđi yüksek düzeyde deđildir. Fakat dilbilimsel ve

kavramsal tanımlar arasındaki benzerlikten yararlanarak geniş kapsamlı ve bilgi tabanlı veritabanı yönetim sisteminin oluşturulması kolaylaşmıştır.

VII. KAYNAKÇA

- [1].Barillere B, Chan B., Goff J.M., Milcent M., Huuskonen P.,Stampfli R.,Willmott C, "Corparison Between Gemstone, Ontos and O2, Oriented Database Management Systems", 1993, <http://light.web.cern.ch/Light/cortex/report/>
- [2] Beraha S., Su J.; "Support for Modeling Relationships in Object-Oriented Databases", Data& Knowledge Engineering 29 (1999) 227-257.
- [3] Dubitzky W., Büchner A.G., Hughes J.G., Bell D.A,"Towards Concept-Oriented Databases", Data&Knowledge Engineering 30 (1999) 23-55.
- [4] Goldstein R.C., Storey V.C., "Data Abstractions:Why and How?", Data&Knowledge Engineering 29 (1999) 293-311.
- [5] Meyer I., Skuce D., Kavanagh J.,Davidson L, "Integrating linguistic and conceptual analysis in a WWW-based tool for terminography, <http://www.qucis.queensu.ca/achallc97/papers/p021.html>
- [6]<http://www.cia.mty.itesm.mx/~lgarrido/Repositorios/Case-Based Reasoning Repository>
- [7]<http://www.chimu.com/publications/> , ChiMu Corporation.