

Web Tasarım Mantığının Desen Tabanlı Biçimselleştirilmesi

Ahmet Sıkıcı¹, N.Yasemin Topaloğlu²

^{1,2} Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü, 35100, Bornova, İzmir

¹ ahmets@staff.ege.edu.tr, ² yasemin@bornova.ege.edu.tr

Özet. World Wide Web yaşamımızın pek çok alanını etkileyen global bir bilgi ağıdır. Web'in dağıtık, bağlantılı, görsel ve kurlsız yapısı, onu sıradışı bir yazılım geliştirme alanı haline getirmektedir. Bu bildiride, Web alanının kendine özgü koşullarındaki tasarım çalışmalarının etkinliği ve verimliliği için desen tabanlı bir yaklaşım öneriyoruz. Bu yöntem her bir tasarıma ilişkin anlamsal özün, ayrı bir desenler kümesi olarak ifade edilmesine dayanmakta ve soyut çözümlerin zorunlu bir matematiksel kesinlik içinde formüle edilmesini gerektirmektedir. Böylece hem yeniden kullanılabilir tasarım deneyimleri yanlış anlamaya neden olmayacak biçimde kodlanabilecek hem de tasarım aşaması ile gerçekleştirim aşaması arasında sorunsuz bir geçiş sağlanacaktır.

1 Giriş

Web'in bir bilgi ortamı ve bir uygulama alanı olarak benzersiz özellikleri, birbirinden bağımsız birçok tasarım sorunu ortaya çıkarmaktadır. Web tasarımları, uygulama mantığının yanı sıra, içerik, yol-bulma (navigasyon), sunum ve görselliği vurgular. Tasarım kaygılarının bu ayrıklığı, bütünleştirme problemleri olan çok aşamalı tasarım yaklaşımlarına neden olmuştur. Oysa tasarımın bütünlüğünün sağlanabilmesi için, tüm tasarım kaygılarına aynı anda yanıt verebilecek, kapsamlı bir temel başvuru modeli gerekmektedir.

Desen paradigması [1] çevresinde şekillenen fikirlerin, Web tasarımının gereksinimleri ile uyum içinde olduğunu gözlemledik. Bu çalışmada, formal bir desen modelinin yardımıyla, desenleri temel tasarım yapıtaşları olarak görevlendiren bir yaklaşım sunuyoruz. Bu yaklaşımı örneklemek üzere, tasarımların anlamsal içeriğini, yeniden kullanılabilir desenler olarak kodlayabilen bir betimsel dil olan Web Tasarımı Tanımlama Dilini (WTTD) tanıtacağız.

Bildirinin ikinci bölümünde Web için tasarım geliştirmenin zorlukları tartışılacaktır. Üçüncü bölümde WTTD tarafından desteklenen biçimsel yaklaşım tanıtılacak ve dördüncü bölümde de WTTD'nin kullanımı bir desen tanımlanarak örneklenecektir. Beşinci bölüm sonuçları içermektedir.

2 Web Tasarımına Yaklaşım Stratejisi

Büyük çaplı Web sistemlerinin geliştirilme ve bakım zorluğu, Web Modelleme Dilleri (WMD) adı verilen yeni bir geliştirme aracı sınıfının doğmasına neden olmuştur. Bazı popüler WMD'ler: OOHDM [2], WebML [3] ve Conallen [4]'dir. WMD'ler nesneye yönelik tasarım, XML ve bazı özelleşmiş modellerden değişik oranlarda yararlanırlar. Kavramsal yapı, içerik düzenlemesi, yol-bulma (navigation), kullanıcı arayüzü, ve işlemlerin tasarlanması, WMD'lerin temel görevleridir. Gözlenen genel yaklaşım bu gereksinimlerin tasarım aşamaları veya katmanları şeklinde düzenlenerek, her birinin kendi mantığı, araçları ve diyagramlarıyla ele alınmasıdır. Dolayısıyla, bir WMD'nin başarısı için belirleyici olan kritik etken, bu katmanların bütünleştirilmesi olmaktadır. Bu bütünleştirme içinde en zoru, hiper-ortam (hypermedia) yapısı ile uygulama mantığının uzlaştırılmasıdır[5]. Bu soruna pekçok kullanışlı yaklaşım öne sürülmüş olmakla birlikte, Web geliştirme süreçlerinin bugünkü durumu yazılım geliştirme süreçlerinin eriştiği düzeyde değildir. Karmaşık Web sitelerini kurmak ve kullanıma açmak için başvuru araçlar; problemin temelinde yatan mantığa tek tip ve sistematik bir biçimde ulaşılmasını gerektiren yazılım geliştirme süreçlerinin izlenmesine uygun değildir [6].

Web geliştirme alanının benzersiz özellikleri, özel ilgi gerektiren ve bazı geleneksel tasarım alışkanlıklarının yeniden gözden geçirilmesini gerektirebilecek belirli teknik sorunlar getirmektedir. Web yapıtlarının dağıtık, bağlantılı ve belge-tabanlı doğası, günümüz yazılım geliştirme kültürünün modülerlik, tip kontrolü ve koruma (encapsulation) gibi bazı temel kavramlarıyla uyumsuzluk göstermektedir.

Web sistemlerinin modülerliği, yapıtaşları arasındaki yüksek dereceli ilintinin (coupling) baskısı altındadır. Sorunun kaynağı, Web'in bağlantılı yapısı ve bir birimin kimlik ve anlamının, içinde yer aldığı bağlama, yüksek derecede bağımlı olmasıdır. Bu durumda Web yapıtlarının geliştirilme süreci, modüler olmaktan çok bütüncül bir bakış açısı gerektirmektedir. Web yapıtları, anlam, kullanım ve kurulumlarını yansıtan bir bağlamla bütünleşirler [7]. Bu tür bir bağlam bilgisi olmadan, bir Web tasarımı tamamlanmış bir yapıt olarak görülmemelidir.

Web'i oluşturan yapıtların çoğunda, üzerinde bir tip kontrolü sisteminin çalışabileceği katı bir yapı bulmak zordur. Web, en temel niteliklerini İnternet'in fiziksel şekilden almıştır ve bu onun global, dağıtık, dinamik, merkezsiz ve sıradüzensiz bir yapıda olduğu anlamına gelmektedir. Bunun bir sonucu olarak, topluluk davranışı, çoğul roller, dağıtık ve bazen örtüşen ifade şekilleri, güvenilir biçimlere fazla yer bırakmayacak ölçüde yaygındır.

Günümüzde yazılım geliştirmenin temel kavramlarından olan koruma (encapsulation), karmaşıklığın giderilmesine yönelik etkin bir araçtır. Ancak geleneksel koruma yaklaşımı Web uygulamalarının geliştirilmesinde yeterli yarar sağlamayacaktır. Sorunun kaynağı, içerme ilişkisinin kavramsal anlamda geçişli olması ve herhangi bir yapının tipine ilişkin kullanışlı bir doğrulanmanın, bu yapıtı oluşturan bileşenlerin de içeriğini incelemeyi gerektirmesidir. Örnek olarak “resimli sayfa” gibi bir tanımın doğrulanması düşünülebilir. Bir Web sayfasının resim içerip içermediğini bilmek için gereken arama, içiçe geçmiş çerçeve ve tablolar üzerinde, dıştan içe doğru ilerleyerek tüm düzeyleri denetleyebilmelidir. Dolayısıyla katı bir koruma, Web için pratik değildir.

3 Web Tasarımına Biçimsel Bir Yaklaşım

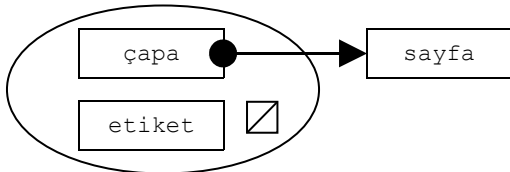
Web tasarımlarının biçimselleştirilmesi projesi, tasarım bilgisinin kesin ve belirleyici bir ortam yardımıyla doğrudan bir gösterimini gerektirmektedir. Bir Web projesinin içeriğini oluşturan unsurların birleştirilebilmesi için bu ortam, Web’e ilişkin tüm tasarım adımlarında, tüm soyutlama ve ayrıntı düzeylerinde etkili olabilmelidir. Ayrıca istenen kesinlik düzeyinde, bir yapının, verilen bir tasarıma uyup uymadığı matematiksel yöntemlerle belirlenebilmelidir.

Web uygulamaları geliştirilmesi konusuna yaklaşımımızı Web Tasarımı Tanımlama Dili adını verdiğimiz desen tabanlı bir betimsel dil olarak somutlaştırdık. Desenler, bir sistemi oluşturan parçalar yerine, bu parçalar arasındaki ilişkiler üzerine yoğunlaştıkları için[1], Web’in merkezsiz, örtüşen ve bağlantılı yapısı ile uyum içindedirler. Soyutlama yetenekleri ve anlamsal ağların[8] esnekliğini yansıtmalarıyla WTTD desenleri, tasarım bilgisini taşımaya son derece uygun veri yapılarıdır. Yaklaşımımız, desen biçimselleştirilmesi konusundaki önceki çalışmalarımıza [9][10] dayanmakla birlikte Web ortamına özgü koşulları da göz önünde bulundurmıştır. İzleyen bölümlerde, WTTD’nin temel yapısının tanıtılmasının ardından, bu tasarım dilinin ikinci bölümde belirtilen problemlere sunduğu çözümler incelenecektir.

3.1 Temel WTTD Yaklaşımı

WTTD şematik bir dildir ve basit birkaç kurala dayanmaktadır. WTTD şemalarında bakılması gereken en önemli eleman bir elipstir. Bu elips şemanın tanımlamayı amaçladığı terimi, şema içinde temsil eder. Dolayısıyla elipsin içi, terimin içeriğini, dışı ise bağlamını göstermektedir. İçerik ve bağlam tek bir tanım içinde bütünleşmiştir ve bir anlamsal ağın iki ayrı bölgesi olarak görülebilir.

yolBulma :



Şekil 1. WTTD şeması örneği

Şekil 2.’de WTTD tanımlarına sade bir örnek verilerek genel anlatım yapısı gösterilmiştir. Örnekte basit bir metinsel yol bulma bağlantısının yapısı görülmektedir. Desen *yolBulma* adı altında tanımlanmıştır ve içeriğinde bir *etiket* ve bir *çapa* bulunmaktadır. Etiket, ekranda görülebilen ve üzeri tıklanabilir bir metin parçasıdır; *çapa* (anchor) ise etiket tıklandığında gidilen Web sayfasının adresini taşır. Elipsin içindeki, üzeri diyagonal çizili dörtgen, bir *sonlandırıcı düğüm*dir. Sonlandırıcı düğümün varlığı bu bölgede daha fazla düğümün olamayacağı anlamına gelir. Eğer elipsin içine yeni bir düğüm eklenirse, tanım artık *yolBulma* tipinden olamaz ve bu durum mekanik yöntemlerle saptanabilir. Şemanın bağlam bölümünde bağlantının hedeflediği Web sayfası görülmektedir. Bir yol bulma bağlantısını temsil eden bir ok, *çapa*’nın bu sayfayı gösterdiğini belirtmektedir.

3.2 Kavramsal Modülerlik

Web ortamı modülerliği mimari açıdan baskılamış olsa da, desenler yardımıyla farklı türden bir modülerlik korunabilir. Ne var ki modülerlik terimi desenler söz konusu olduğunda yanlış yönlendirici olabilir, çünkü bir tasarım çözümünün gerçekleştirilmesi sırasında desenler sıklıkla örtüşebilmektedir. Bu durumda görev bölümü, sadece mantıksal düzeyde gerçekleşmekte ve modülleri ayırdetmek için fiziksel sınırlar yerine desenlerin kesin tanımlarına dayanılmaktadır.

WTTD’nin ilinti problemine yaklaşımı, yapıtları bağlamları ile birlikte tanımlama şeklindedir. Bir WTTD tanımı tanımladığı nesnenin iç yapısı ile birlikte, içinde bulunması gerektiği çevreye göre konumunu da tam olarak gösteren büyük bir röntgen görüntüsü gibidir. Her bir tanım, kendi konusunun gerektirdiği hem içerik hem de bağlamı kapsar ve

elemanlar arasında, tanımlanan kavramın gerektirdiği tüm bağlantıları içerir.

3.3 Sıradışı (irregular) Tip Kontrolü

WTTD tip denetimini zorlaştıran Web'e özgü yapısal sıradışılıkları karşılamaya yönelik mekanizmalar içermektedir. Web üzerinde sabit yapılar bulmak zor olsa da uygun derecede soyut bir tanımlama dili kullanarak dinamik bir sistemin değişmezlerini kodlamak olasıdır. Soyut düzeyde çalışıldığında, biribirinden ayırık görünen sistemler arasındaki benzerlikler yakalanabilir.

WTTD desenleri değişkenlerin ve sınırlamaların bir bileşimi olarak tanımlanır ve bu yolla geniş bir betimleme alanına ulaşılır. WTTD'nin iskeleti içiçe yuvalanabilen küme ve ağ desenlerine dayanır. Ağ mantığı birimler arasındaki ilişkilerle ilgilenirken, küme mantığı birimlerin içerikleri hakkında önermelerde bulunur. Küme ve ağ kavramları birbirine sıkı sıkıya bağlıdır ve tanımlama sırasında birinden diğerine kolayca geçilebilir. Örneğin "belli bir düğümün tüm bağlantıları" bir küme ile ifade edilirken herhangi bir küme de, kümenin kimliğini taşıyan bir düğüm ile bu düğümü küme elemanlarına bağlayan bir demet kapsama ilişkisi olarak ifade edilebilir.

Küme mantığının önemli bir getirisi *sonlandırıcı işaretlerin* varlığıdır. Bir sonlandırıcı işaret, belli bir türden bir elemanın, o desenin herhangi bir örneğinde (instance) bulunmadığını belirtir. Sonlandırıcı işaretler herhangi bir türden bir düğüm veya bağlantı üzerinde iş görebilirler. (Sonlandırıcı düğüm veya sonlandırıcı bağlantı adını alarak) Sonlandırıcı işaret kavramı iki nedenden dolayı önemlidir. Öncelikle bu sınırlamalar aracılığıyla, bir tasarımın alt tipi olmayan formlara gelişigüzel evrimleşmesi önlenir. İkinci olarak da, "menüsüz bir sayfa" gibi olumsuz içerikli önermelerle tanım yapmak mümkün olur.

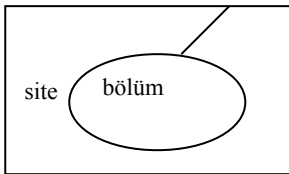
WTTD tanımlarında mantıksal işlemler de kullanılabilir. VEYA işlemcisini kullanarak birbirine benzemeyen örnekleri olan desenler tanımlanabilir. VE işlemcisi ile ise bir yapının pek çok farklı görünümü, herbiri ayrı birer tanım olarak verilebilir.

Böyle bir esneklikle, bir tasarımın geliştirilebilen yönlerine ilişkin genel fakat duyarlı tanımlar yapılabilir. Dinamik olarak değişen bir yapının temel mantığı bir desen olarak ifade edildiğinde, bu yapının, amaçlanan tasarımının dışına çıkmadan, hangi yönlerde büyüyüp nerelerde küçülebileceği bilinebilmektedir.

3.4 Koruma Denetimi

WTTD'de bütünleme (aggregation) ilişkisi parçanın, bütünü hemen içinde mi yoksa iç içe geçmiş bir dizi kap içinde mi yuvalandığı konusunda bilgi vermez. Hemen içinde yuvalanmış olmak özel bir durumdur ve açıkça belirtilmelidir. Tanım yapılırken yuvalanma katmanlarının sırası ve sayısı üzerine çeşitli kısıtlar konabilmektedir. Yuvalanma zincirleri sıralı oldukları için ağ mantığının bir parçasıdır. Yuvalanma derinliği ve ara kapların özellikleri ile oynanarak yuvalanma desenleri elde edilebilmektedir.

anaBölüm:



Şekil 2. Hemen içinde yuvalanmanın gösterilmesi

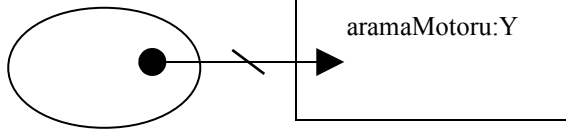
Şekil 2'de hemen içinde yuvalanma durumunu örneklemek amacıyla anaBölüm adında bir desen tanımlanmıştır. Site ve bölüm düğümlerini birleştiren çapraz çizgi, bölümün, sitenin hemen içinde yuvalanmış olduğunu göstermektedir.

İçerilen bir birim, kendisini içeren kabın dışındaki bir birimle bağlantı kurduğunda ortaya çıkan duruma koruma aşımı (encapsulation violation) adı verilir. Koruma aşımına herbir yapıya giren ve ondan çıkan bağlantılar tarafından yol açılır. WTTD'de birimler, içeriklerini bağlamlarından ayıran bir yüzey olarak düşünülebilirler. İçerik ve bağlam arasındaki tüm ilişkiler bu yüzeyin içinden geçmektedir. Yüzeyi delen bağlantılar arasında bir sıralama ilişkisi bulunmadığından bunlar küme mantığı ile ele alınabilirler. Bu şekilde koruma aşımı desenleri oluşturmak mümkündür.

Her ne kadar ismi bir hata durumunu çağırırsa da koruma aşımı, yazılım geliştirme sürecinin her aşamasında karşımıza çıkabilecek bir meta-desendir. Web uygulamaları için ise varlığı veya yokluğu bir tanımın gerekli bir ögesi olabilmektedir. Örneğin "X sitesinden herhangi bir arama motoruna geçiş yoktur." dediğimizde bu önermeyi şu şekilde de düşünmek olasıdır: "X sitesi içinden başlayıp X'in dışına çıkan ve bir arama motoru sitesinin içindeki bir sayfayı

hedefleyen herhangi bir yol bulma bağlantısı yoktur.” Sınırlamanın hedefi öncelikle X sitesinin yüzeyidir ve bu yüzeyi delip geçen belirli türden bağlantılar yasaklanmıştır.

X:



Şekil 3. Koruma aşımı örneği

Şekil 3’de bu tanımın WTTD ile ifade edilişi görülmektedir. Şema tarafından tanımlanan X sitesinin içinden başlayan ve aramaMotoru tipinden olan bir Y sitesini gösteren bir yolbulma bağlantısı, bir *sonlandırıcı bağlantı* olarak ifade edilerek yasaklanmıştır.

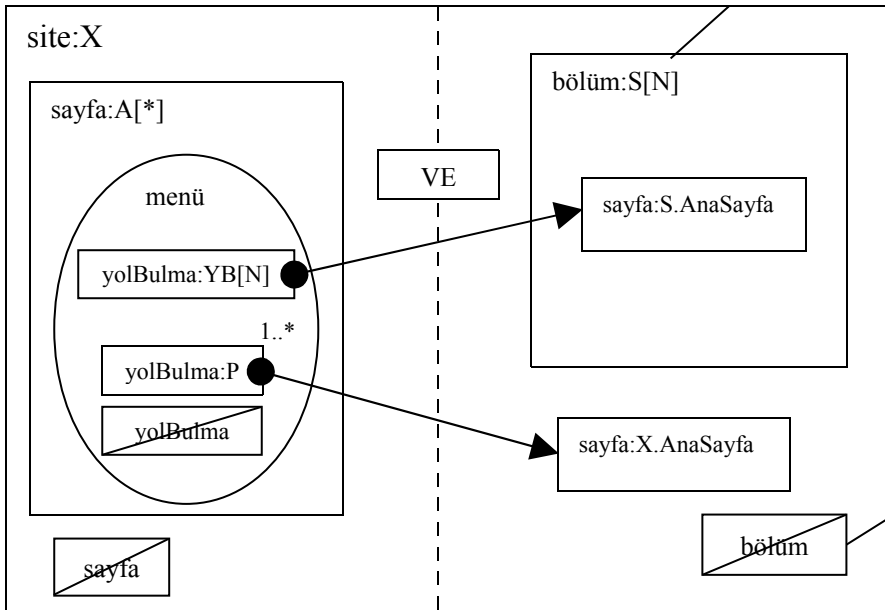
WTTD’nin anafikri, tasarımların Web projeleri geliştirme alanında anlamlı olan ölçütlere göre sınıflandırılabilmesini sağlamaktır. Web üzerinde işlev görececek olan bileşenlerin ve diğer yapıtların, ne tür bir soruna tasarımsal bir çözüm getirdiği, hem kullanım hem de yeniden kullanım açısından önemlidir. WTTD’deki ifade biçimleri temel Web tasarımı sorunsallarına değinmek amacıyla içerik, bağlam, ilişkiler ve genel topolojik yapı hakkında pekçok önermeyi kodlayabilirler.

Yazılım tasarımında yeğlenen katı yapısal tutumun aksine, bilgi tabanlı bir yaklaşımla, üst menü, site haritası, yol bulma paneli gibi bileşenler, anasayfa, hoşgeldin sayfası, resimsiz sayfa, bilgi ekranı gibi sayfa düzeni standartları, ve de katalog, sunum, resim albümü, merkezi site, Web halkası (Web ring), sanal şehir gibi topolojik kısıtlar barındıran tanımlar sadece üretime yönelik değil, aynı zamanda var olan yapıtların tanınması, standart isimler altında toplanması, ve yeniden kullanılabilir bileşenlere dönüştürülmesini destekler şekilde ele alınmaktadır.

Tasarım bilgilerinin, tasarımcı ve programcılar tarafından yorumlanmayı gerektiren bulanık ifadeler içermemesi otomatikleşmeye uygun bir ortam sunar. Bir üst katmanda yer alacak olan araçların WTTD tanımlarının tutarlılık, kapsama, kesişme gibi ilişkilerini iyi tanımlanmış yöntemlerle araştırmasının yanında tanımları iç içe yerleştirerek veya üst üste bidirerek yeni tanımlar üretmesi de olasıdır.

4 Site Yol Bulma Menüsü Deseni

WTTD’nin özelliklerini gerçek bir Web deseni üzerinde görmek faydalı olacaktır. Tanıtacağımız desen Web üzerinde oldukça yaygın bir menu kullanımını anlatmaktadır. Site Yol Bulma Menüsü adını verdiğimiz bu yapı, bir sitenin herhangi bir noktasından o sitenin ana bölümlerinden birine atlamayı sağlamaktadır.



Şekil 4. Site Yol Bulma Menüsü Deseni

Tasarımın özü aşağıdaki gibi özetlenebilir:

- 1.Desen tarafından betimlenen menü, sitenin her sayfasında yer almaktadır.
- 2.Menü sitenin her bir ana bölümüne bir bağlantı içermektedir.
- 3.Menü sitenin ana sayfasına da bir bağlantı içermektedir.
- 4.Menü başka herhangi bir bağlantı içeremez.

Şekil 4'te desenin WTTD ile ifadesi gösterilmiştir. Şemadaki en çok göze çarpan özellik tüm desenin `site:X` olarak etiketlenmiş tek bir büyük düğüm içinde yer alıyor olmasıdır. Bu düğüm, menünün, üzerinde yol bulma hizmeti vereceği siteyi temsil etmektedir. `VE` mantıksal işlemcisi ile etiketlenmiş bir ayraç bu düğümü iki parçaya ayırmaktadır. Bir şema bölgesinin `VE` ayraç ile bölünmesi, o bölge için her ikisi de geçerli olan iki desen parçası tanımlanmasına olanak sağlar. Örnekte de `X` sitesinin her ikisi de geçerli olan iki ayrı görünümü kullanılmıştır. Bu görünümlerden biri siteyi bir sayfalar kümesi olarak algılamak, diğeri bir bölümler sıradüzeni olarak yorumlamaktır.

Sayfalar kümesi üzerinde ifade edilmek istenen önerme basittir. Her sayfa üzerinde bir menü bulunmalıdır. `sayfa:A[*]` çok sayıda sayfa demektir ve bir sayfa sonlandırıcısı ile aynı bağlamı paylaştığı için *her sayfa A* şeklinde okunabilir. Bunun içinde bulunan menü olarak etiketlenmiş elips, sitedeki her sayfada bulunan bir menünün tanımlanmakta olduğunu gösterir. Menünün seçenekleri daha önce tanımlanmış olan `yolBulma` tipindendir. Bu seçeneklerin `N` tanesi sitenin bölümlerinin ana sayfalarına, bir tanesi de tüm sitenin anasayfasına `yol-bulma` bağlantısı ile bağlıdır. `yolBulma` tipinden olan düğümler sonlandırılarak, menüde başka seçeneklerin olması engellenmiştir.

Sitenin sağ taraftaki yorumunda ise menünün `yol-bulma`ya ilişkin hedefleri tanımlanmıştır. Bu tanıma göre *ana bölüm*, sitenin *hemen içinde* yer alan bölümdür ve kendine ait yerel bir *ana sayfası* vardır. Ana bölüm ve ilgili menü seçenekleri için aynı çoğulluk değişkeni (`N`) kullanılarak, bunlar bire-bir eşlenmiştir. Sitenin *hemen içindeki* bölümler sağ alttaki sonlandırıcı düğüm ile sonlandırılarak, yapılan tanımın tüm ana bölümleri kapsamı sağlanmıştır.

Tüm ana bölümlerin her bir ana sayfası, bir menü seçeneği tarafından hedeflenmektedir. Bunun yanında tüm sitenin ana sayfası da, hiçbir bölümün içinde olmayacak şekilde yerini almıştır ve bir menü seçeneği ile eşleşmesini sağlayan bir `yol bulma` bağlantısının hedefi durumundadır.

5 Sonuç

Web ve benzeri hiper ortamları hedefleyen geliştirme çalışmaları, klasik yazılım geliştirme kuramsal altyapısıyla kapsanamayacak esneklikte bilgi gösterimi mekanizmaları gerektirirler. Klasik yazılım geliştirme paradigmalarını temel alan ve çok fazla yaklaşımlarla anlatım gücüne eklemeler yapmaya çalışan Web geliştirme yöntemleri yerine, Web'e özgü bir geliştirme paradigması için çalışmak uygun görünmektedir.

Geliştirmiş olduğumuz Web Tasarımı Tanımlama Dili'nin bu yönde atılmış bir adım olduğunu düşünüyoruz. Düğümler ve bağlantılar kullanan bilgi gösterim biçiminin yanında, içerme, koruma, bağlam gibi kavramların Web ortamına uygun esneklikte ele alınmış olması, anlatım gücünü arttıran etkenlerdir. Anlatım gücünün artmasıyla Web projelerinin tüm yönleri tek bir araç tarafından ele alınabilecek ve bu alanda geniş çaplı otomasyonun önü açılacaktır.

Kaynakça

1. Coplien, J.O.: Patterns and Art., C++ Report 12(1) (2000) 41-43.
2. Schawbe, D., Rossi, G., Barbrosa S.: Abstraction, Composition and Lay-out Definition Mechanisms in OOHDM, in Cruz,I.F., Maks, J. and Wittenburg, K. (eds.) Proceedings of the ACM Workshop on effective Abstractions in Multimedia, San Fransisco, CA (1995).
3. Ceri S. Freternali, P., Bongio, A.: Web Modelling Language(WebML): A Modelling Language for Designing Web Sites. WWW9 Conference, Amsterdam (2000).
4. Conallen, J.: Modeling Web Application Architectures with UML. Communications of the ACM 42(10) (1999) 63-70.
5. Gu, A.,Henderson-Sellers, B.,Lowe, D.: Web Modelling Languages: The Gap Between Requirements And Current Exemplars. The Eighth Australian WWW Conference (2002).
6. Gomez,J., Cachero, C. and Pastor, O.: Conceptual Modeling of Device-Independent Web Applications. IEEE Multimedia April-June (2001) 26-39.
7. December, J. and Ginsburg, M.: HTML and CGI 3.2 Professional Reference Edition. <http://docs.rinet.ru:8080/CGI3.2/>
8. Rich, E. and Knight K.: Artificial Intelligence. Mc Graw Hill (1991).
9. Sıkıcı, A. and Topaloglu, N.Y.: A Knowledge Based Approach to Design with Patterns. in T.Hruska and M.Hashimoto (eds.): Knowledge Based Software Engineering, IOS Press (2000) 133-136.

10. Sıkıcı, A. Topaloglu, N.Y.: Towards Software Design Automation with Patterns. *Informatica* Vol.25, No. 3 (2001) 309-317.