

Saraiva, Antonio Mauro, André Riyuti Hirakawa, and Carlos Eduardo Cugnasca. 2006. Section 3.3 Topics on Software Evolution, pp. 139-153 of Chapter 3 Methods, Algorithms, and Software, in CIGR Handbook of Agricultural Engineering Volume VI Information Technology. Edited by CIGR-The International Commission of Agricultural Engineering; Volume Editor, Axel Munack. St. Joseph, Michigan, USA: ASABE. Copyright American Society of Agricultural Engineers.

Çevirmen: Tekin ÖZTEKİN

Çeviri Editörleri: Sefa TARHAN ve Mehmet Metin ÖZGÜVEN

3.3 Yazılım Evrimi Hakkında Konular

Yazarlar: A. M. Saraiva, A. R. Hirakawa ve C. E. Cugnasca

Çevirmen: Tekin ÖZTEKİN

Özet: Bilgisayar sistemleri on yıllar öncesi beklentilerinin daha da ötesinde bir gelişim gösterdi. Bu gelişim sürecinin temelini donanım evrimi oluşturmuş olsa da, kullanıcı için en görünür kısım yazılım evrimi olmuştur. Birçok durumda kullanıcının görmediği gelişmeler arasında yazılım geliştirme süreçlerindeki, yazılım paradigmalarındaki ve endüstride kullanılan şimdilerde ise günlük yaşamımızda mevcut olan teknolojideki değişiklikler sayılabilir. Sadece en görünen kısmı ifade etmek gerekirse temel seviyedeki işletim sistemlerinin komut yönetiminden grafiksel yönetimli sistemlere dönüşmesidir. Bilgisayarlar veya mikroşlemcili sistemler, mikrodalga fırınlardan traktörlere kadar farklı alet ve ekipmana girmiştir. Onlar ve cep telefonlarının yanı sıra, avuç içi bilgisayarları ve oyunlar gibi diğer yeni el cihazları için özel işletim sistemleri geliştirilmek zorunda kalındı. Yazılım maliyeti, donanım maliyeti ile karşılaştırıldığında daha kritik bir duruma geldi. Şimdilerde geliştirilmekte olan sistemler daha da karmaşıktır. Bunun neticesinde, yazılım geliştirme için yeni yaklaşımlara, metotlara, paradigmalara, dillere, teknolojilere ve araçlara ihtiyaç duyulmaktadır. Yazılım endüstrisi, şimdilerde büyük şirketler ve bağımsız yazılım toplulukları tarafından paylaşılan milyar dolarlık bir markettir. Bu konuları tamamen kapsayan kitaplar olmasına rağmen burada sadece seçilen bazı konuların bazı yönlerinin kısa bir tanıtımı sunulacaktır.

Anahtar Kelimeler: Nesne yönelimli, Açık yazılım, İşletim sistemi, Yazılım modelleme, Görsel program vasıtaları, UML, XML.

3.3.1 Giriş

Son on yıldaki bilgisayar sistemlerinin muazzam evriminin eşit önemde iki unsuru: donanım ve yazılımdır. Donanımla karşılaştırıldığında yazılım maliyetleri önemli derecede arttığı için yazılım geliştirme süreci dikkate değer bir ilgi odağı olmuştur. Maliyetlerinin yanında birkaç yönüne de değinirsek; yazılımın kalitesi, yazılımın kullanım kolaylığı, daha bir profesyonel yaklaşımı gerektirmektedir. Bu konulara ilişkin birkaç noktaya bu bölümde değinilmiş olup bunlar; işletim sistemleri, yazılım dilleri, nesne yönelimli yazılım modelleme, veri değişimi için nesneye dayalı yazılımın Tümleşik Modelleme Dili (UML, Unified Modeling

Language), Genişletilebilir İşaretleme Dili (XML, Extensible Markup Language) ve açık yazılım kavramlarıdır.

3.3.2 İşletim Sistemleri

Her bilgisayarın daha rahat kullanımı için bir tür sistem yazılımına ihtiyacı vardır. Bu sisteme *işletim sistemi* (İS) denilmekte olup, uygulama programlarının düzgün çalışmalarını ve kendi özel görevlerini yerine getirmelerine imkan sağlayan bilgisayarın yazılım ve donanımlarının kontrolünden sorumludur. Kullanıcı programının doğrudan sistem kaynaklarını (işlemci, bellek ve çevre birimleri) kontrol etmesine izin verilmez. Bunun yerine birçok program, işletim sistemine dahil olan standart yazılım sürücülerini paylaşmalıdır. Buna bağlı olarak işletim sistemleri, uygulama gereksinimlerine göre farklı özelliklere sahip olacak şekilde geliştirilebilir. Örneğin, kullanım gereksinimlerine uygun olarak bazıları bilgisayar kullanımını basitleştirmek için planlanırken diğer işletim sistemleri ise çoklu görevliliği, çoklu kullanıcıyı destekleme veya gerçek zamanlı uygulamalarda kullanım gibi bilgisayarı daha verimli hale getirmek için karmaşık olabilir.

İşletim sistemleri evrimi, mikrobilgisayar başarısı için oldukça önemli olmuştur. Aşağıdaki öğeler, günümüzdeki mikrobilgisayarlarda kullanılan en önemli işletim sistemleridir [1].

Masaüstü Bilgisayarı İşletim Sistemleri

Disk İşletim Sistemi (DOS, Disc Operating System), kişisel bilgisayarların piyasaya sürüldüğünden bu yana uzun yıllardır kullanılmaktadır. Kendi *komuta dayalı ara yüzünden* dolayı çoğu kullanıcı için sezgisel olmayıp oldukça da zordur. İlk *grafiksel kullanıcı ara yüzü (GUI, Graphical User Interface)*, Xerox şirketine ait Palo Alto Araştırma Merkezinde 1970'lerde icat edilmiştir. *WIMP* veya *Windows, simgeler, menüler ve işaretçi* o zamanın mevcut grafiksel kullanıcı ara yüzlerindendi ve *WIMP GUI* olarak adlandırıldılar. Günümüzde ise bunların hepsi basitçe GUI kısaltmasıyla bilinmektedir. Bir GUI'de; resimler ve grafik sembolleri, komutları, seçenekleri veya eylemleri temsil eder. Bu arayüzlerin diğer bir ortak yönü ise masaüstü metaforu olmalarıdır: görüntü ekranında simgelerin gruplandırıldığı alan, masaüstü olarak adlandırılır. Bunun sebebi simgelerin, gerçek bir masa üstünde gerçek nesnelere temsil için tasarlanmasıdır. Masaüstü pencerelere ayrılır ve her birinde, farklı programlar çalıştırılabilirler veya farklı dosyalar görüntülenebilirler.

Bazı uzman kullanıcıların, bir komut odaklı arayüzü ile daha etkin çalışma hissetmelerine rağmen iyi tasarlanmış grafik kullanıcı arayüzleri, kullanıcıyı karmaşık komut dillerini öğrenmekten kurtarır. Bu arayüzler 1983 yılından bu yana ticari olarak önemli hale gelmiştir. Bu arayüzlerin ilki Apple firmasından Mac İS olup daha sonra bunu IBM uyumlu mikro bilgisayarlar için popüler hale gelmiş olan Microsoft Windows takip etmiştir.

Takip eden ögeler ise masaüstü bilgisayarlar için en popüler işletim sistemlerinin temel özelliklerini ve bazı evrim aşamalarını göstermektedir.

Mac İS

Apple Mac İS, basit ve sezgisel olması için geliştirilmiş olup daha az uyumluluk ve yapılandırma problemlerine sahiptir. Birçok kullanıcı, özellikle multimedya düzenleme söz konusu olduğunda kullanımını daha kolay, daha ilgi çekici ve daha verimli bulduğu için Mac İS'nin Windows'tan daha iyi olduğu kanaatindedirler. Mac İS, aynı zamanda virüs ve diğer kötü amaçlı kod saldırılarına daha az etkilenmektedir. Buna rağmen Mac İS, bilgisayarların % 10'undan daha azında kullanılmaktadır. Mac İS, sürekli olarak geliştirilmiş olması gerçeğine rağmen onun çekirdeği ilk sürümünden bu yana önemli ölçüde değişmemiştir. Ek olarak, aynı zamanda çoklu büyük uygulamaları bir arada çalıştırırken problemler çıkartmakta ve çoklu görev yetenekleri kısıtlıdır.

Birçok yeni özellik ve iyileştirme içeren Mac İS X adı verilen önemli bir güncelleştirme gerçekleştirildi. Darwin olarak adlandırılan çekirdeği, Berkeley, Kaliforniya Üniversitesinde geliştirilen UNIX sürümü olan BSD UNIX'i esas almaktadır. Microsoft'un tersine, Apple açık kaynak kodu fikrini benimsemiştir. Bu, topluma değişiklik ve gelişmeleri öne sürme konusunda cesaret kazandırabilir ve aynı zamanda özel ihtiyaçları karşılamak için Mac İS X'in uyarlanmasını da kolaylaştıracaktır. Mac İS X'in temel özellikleri; sanal bir bellek yöneticisine sahip olması, uygulama programları arasında bir müdahalenin olmaması, dayanıklı bir çoklu görev, kullanıcıların ihtiyacı olacak çoğu servisin mevcut oluşu ve önemli internet kaynaklarıdır. Masaüstüne daha bir foto-gerçekçi görünüm sağlaması nedeniyle görünümü de oldukça farklıdır. Mac tarafından diğer birçok ilginç özellikte sunulmaktadır. Ancak, bir ikilem söz konusudur. Bu ikilem, yüksek performansa sahip olmasına rağmen sınırlı birkaç Mac üzerinde kullanılacak yazılımın bulunması ve bunların yüksek maliyetli olmasıdır. Bu sebeple, diğer işletim sistemlerine göre çok küçük bir topluluk Mac kullanmaktadır. Pazar payını arttırmak için yapılan girişimlerde başarılı olmamıştır; belki düşük fiyatlı olmasına rağmen oldukça güçlü bir bilgisayar olan Mac Mini'nin piyasaya sürülmesi bu resmi değiştirmeye yardımcı olabilir.

Windows

Windows'un ilk sürümleri, ilk olarak MS-DOS'un yüklenmesini gerektiren gerçek bir işletim ortamıydı. Daha sonraları, MS-DOS fonksiyonları, Windows'a dahil edilmiştir.

PC'lerin % 90 dan fazlası bir Windows işletim sistemi ile çalışmasına rağmen, çoğunluğu performans ve hataları üzerine olmak üzere sık sık eleştiri almaktadır. Problemlerin asıl nedenleri; MS-DOS ile uyumluluk ve hızlı bir şekilde piyasada veya pazarda yer alma arzusudur. Windows 95 sürümü bu problemleri çözmeye

başlamıştır; onun Tak ve Çalıştır (bilgisayara bir cihaz takıldığında bilgisayarın o cihazı otomatik olarak tanınmasını sağlayan standart) özelliği, her zaman mükemmel olmasa da, Windows'lu PC'lerin kullanılabilirliği açısından önemli bir evrimdir. Ancak, Windows'un ana avantajlarından biri de, Windows için çok sayıda programın geliştirilmiş olmasıdır.

Şu anda mevcut olan Windows'un diğer sürümleri aşağıdaki özelliklere sahiptir:

- **Windows ME (Millennium Edition):** Windows'un bu versiyonu tüketici odaklı bir İS olup, insanların PC'lerden beklentileri olan çok çeşitli fonksiyonları yürütmeleri için tasarlanmıştır. Bu fonksiyonlar; ev ve işyeri uygulamaları, basit ağ ve internete bağlantı, oyun oynama ve multimedya uygulamaları (film düzenleme, MP3 dinleme vb. gibi) dir. Etkin olması gereken ve işletim sisteminin önemli bir parçası olan 9x çekirdeğini temel alan ME, muhtemelen MS-DOS programlarını destekleyen son sürüm olacaktır. ME'nin daha da ilginç yeni özellikleri; önemli bir sorun sonrası daha önceki bir duruma geri dönmek için *Sistem Geri Yükleme* aracının olması, PC internete bağlı olduğunda kritik İS güncellemeleri ve düzeltmeleri (gerekirse onları internetten indirerek) için *Otomatik Güncelleme* fonksiyonuna sahip olması ve ev ağı ve multimedya için sahip olduğu diğer olanaklardır.
- **Windows NT ve Windows 2000:** Bu ürünler, güvenilir ve güvenli bir ortam gerektiren iş istasyonu ve kurumsal internet ağı sağlayıcı (server) pazarında rekabet etmektedirler. Yeni teknoloji (NT, New Technology) Windows 9x'den farklı olarak 32 bit İS olarak tasarlanmıştır. Windows 2000, aynı güvenilirlik ile fakat bir tüketici İS esnekliği ile NT'nin bir evrimidir. Bir yazılım uygulamasının bir diğer uygulamaya müdahale etme şansını azalttığından Windows 2000, Windows 9x den daha hızlı ve daha güvenilirdir. Ayrıca, ağ kaynaklarına bağlantı kolaydır. NT'den farklı olarak Windows 2000, çok sayıda ürünü (ME den daha az sayıda olmasına rağmen) destekleyen Tak ve Çalıştır özelliği için tam destek sunmaktadır.
- **Windows XP:** Windows XP'nin geliştirilmesindeki amaç; Windows 2000'nin istikrarını ME'nin çok yönlülüğü ile birleştirerek ve aynı zamanda yazılım ve donanım geliştirmeyi basitleştirerek daha istikrarlı ve daha basitleştirilmiş bir hesaplama ortamı sağlamak olmuştur. Windows 98/ME/2000 sürümlerinin yükseltilmesi ile de Windows XP sürümü oluşturulabilir. İki sürümü vardır: şirket ve yetkili kullanıcılar için Windows XP Professional ve tüketiciler için Windows XP Home Edition. ME'nin özelliklerine ilaveten birçok yeni özellik eklenmiştir. Bu eklenen yeni özellikler; bekleme modunda düşük güç tüketimi ile PC'ye hızlı geri dönüş (devam) operasyonu, birçok eğlence özellikleri ve multimedya desteği, bireysel kullanıcı hesapları, her uygulama için korumalı bellek alanı, güvenlik duvarı koruması ve uzaktan erişimdir.

Linux

Linux, küçük çekirdeği assembly dilinde geri kalan kısmı ise C dilinde yazılmış eski UNIX işletim sistemi tabanlı ücretsiz bir İS dir. Böylece, bu işletim sistemi göreceli basitlikle özelleştirilebilir. Linux, 1990'lı yıllarda ortaya çıkışından bu yana, daha çok Linux tabanlı Apache'nin çok popüler bir internet platformu olduğu web sunucu pazarında daha fazla takipçi kazanmıştır. Uygun maliyetli bir işletim sistemi olup, açık kaynak kodludur ve basit kullanımlar için iş istasyonları ya da küçük grup sunucular için ilginç ve ucuz bir çözüm olup, eski bilgisayarlarda oldukça iyi sonuçlarla işletimi için daha az sistem kaynağına ihtiyaç duyar.

Linux yine de bir komut satırı ara yüzü olduğu için çoğu kullanıcı için uygun değildir. Bundan dolayı, Linux ortamını basitleştirmek için Windows ortamına benzer Gnome ve K Desktop Environment gibi birçok GUI ara yüzü tasarlanmıştır. Başka bir nokta ise Windows için geliştirilen çoğu yazılımın Linux sürümlerinde bulunamamasıdır. Bununla birlikte, bazı firmalar sunucu pazarında ve diğer popüler masaüstü uygulamaları için Linux ürünlerine yatırım yapmaktadırlar. Birçok kullanıcı, Linux'un kurulum ve kullanım için Windows'tan daha zor olduğu düşüncesindedir. Red Hat ve VA Linux gibi bazı firmalar, kurulumu daha kolay olan Linux sürümleri sunmaktadır. Microsoft ürünleri hala piyasada hakim olsa da Linux, masaüstü İS ve aynı zamanda sunucu pazarında sürekli olarak büyümüştür. IBM gibi birçok önemli şirket, Linux'e önemli yatırım yapmaktadır. Dünya da bazı ülkeler, eyaletler ve belediyeler hükümet düzeyinde Linux'u İS'leri olarak benimsemeleri konusunda güçlü bir eğilimle, ücretsiz yazılım kullanımını teşvik etmektedirler. Bu eylem, önümüzdeki yıllarda yazılım senaryosunu önemli derecede değiştirebilir.

Taşınabilir Bilgisayar İS

Taşınabilir bilgisayar pazarı yeni olmasına rağmen, donanım ve İS de dahil olmak üzere tüketici için birçok seçenek mevcuttur. Genel olarak bunlar kişisel dijital yardımcılar (KDY) olarak bilinmekte olup İS'ne bağlı olarak aşağıdaki gibi kategorilere ayrılabilirler:

- Palm şirketinin [3] Palm İS, tüm Palm modellerinde, Handspring Visor ve Sony Clie tarafından kullanılan temel bir İS dir. Bu işletim sisteminin başarısına, esas olarak kullanım sadeliği atfedilmektedir. Klavyeye ihtiyaç olmayıp, oldukça hızlı, basit ve kolayca isteğe uyarlanabilir. Yeni uygulamaların geliştirilmesini basitleştiren Palm API (Uygulama Programlama Arayüzü) C dili tabanlıdır.
- Microsoft'un [4] Windows CE'si, Windows tabanlı olup, avuç içi bilgi işlem cihazları, cep telefonları ve diğer uygulamalar için tasarlanmıştır. Bu, bazı değişiklikler ve yeniden derleme sonrası Windows uygulamalarının CE'ye adapte olmasına izin verir. Bu, klavye veya fare olmadan, isteğe bağlı ses ve CompactFlash yuvası ile *cep PC*'sinde kullanılabildiği gibi, dokunmatik bir ekran ve küçük bir klavye kullanılarak isteğe bağlı ses ve CompactFlash

ve/veya bir PCMCIA yuvası ile bir el PC'sinde de kullanılabilir. Palm İS'den farklı olarak, kullanıcının Windows ile aşinalığına bağlıdır ve 32-bitlik, çoklu kullanımlı, çok görevli bir İS'dir. Bu benzerlik, yeni yazılım geliştirmek için oldukça önemlidir. Bu güçlü çözüm, MP3 ve video oynatma gibi işlemci yoğunluklu uygulamalara izin verir. Word, Excel Outlook gibi popüler uygulamaların olması da, kullanıcılar için oldukça caziptir. Bununla birlikte, Windows CE daha fazla bellek gerektirir ve aynı zamanda pil tüketimi de yüksektir. Yeni sürümler Windows XP'ye benzer ve İnternet Explorer, Ağ Erişim (Network Access), MSN Messenger, Windows XP masa üstü tabanlı yeni bir görünüme sahiptir.

Psion KDY'lerin Eloc'u gibi diğer İS'leri mevcut olup ve son zamanlarda Agenda Computing, Samsung ve Sharp gibi Linux tabanlı KDY'lerin gelişimi teşvik edilmiştir.

Gerçek Zamanlı İşletim Sistemleri ve Gömülü Yazılım

Birçok uygulama, zamanla değişen özellikleri olan ve öngörülebilir zaman bağımlı tepkileri gerektiren ortamlarla etkileşim içerisindedir. Bundan dolayı, bu uygulamalar için planlanmış sistemler ve bu sistemlere ait yazılımlar, diğer koşullar ve yükü ne olursa olsun sadece doğru bir biçimde hesaplanmış tepkiler vermek yerine, aynı zamanda öngörülebilir zaman bağımlı davranışlarda vermelidirler. Bunlara *Gerçek Zamanlı Sistemler (GZS)* denir [5, 6].

GZS'lerde bazı önemli özelliklerin olması arzu edilir; örneğin zaman yönetimi, zamana bağlı görevlere uygunluk, pik yükü karşılama, zaman tepki tahmin edilebilirliği gibi. GZS'ler, endüstriyel süreçler, hava taşıtları ve otomotiv gibi kritik uygulamalarda sıkça kullanıldıkları için hata toleransı da önemli bir özelliktir. Onların standart bir İS'den daha hızlı olmaları gereksizdir. Fakat onların dışsal uyarıcıların tahmin edilebilir olup olmamasından bağımsız olarak tahmin edilebilir bir zaman tepkisine sahip olmaları gerekir. İşlemcinin kesme mekanizmaları, birçok durumda tahmin edilebilirlik sağlamak için kullanılabilir. Görev zamanlaması; her seferinde işlemcilere daha uygun görevler atayarak, Gerçek Zamanlı İşletim Sistemleri (GZİS) için dikkatli bir biçimde tasarlanmış olmalıdır. Herhangi bir noktada daha önemli bir görevin işlemciye hemen ulaşması için halihazırda yapılan bir görevin kesilmesine izin veren öne alım öncelikli zamanlama desteklenmelidir. Tablo 1'de bazı GZİS örnekleri verilmiştir.

Tablo 1. Bazı ticari gerçek zamanlı işletim sistemleri

GZİS	Tedarikçi Firma
pSOS Sistemleri	Integrated Systems
VxWorks	Wind River Systems
VRTX	Microtec Research
OS-9	Microware Systems
SPOX	Spectron Microsystems
QNX Software	QNX
HP-RT	Hewlett-Packard
LynxOS	Lynx Real-Time Systems
Inferno	Lucent
Virtouso RTOS	Eonic Systems

Günümüzde, cep telefonları, MP3 çalarlar, otomobil cihazları, bilgisayar çevre birimleri ve birçok diğerleri gibi mikroişlemci (gömülü sistem veya GS) tabanlı birçok elektronik ürün vardır. Bunların kontrolü için *gömülü yazılım* kullanılır ve bu gömülü yazılım kısa, etkili, güvenilir, girdi ve çıktılarının işlenmesinde hassas olmalıdır. Silikonlu sistemlerin ortaya çıkmasıyla GZİS'ler, uygulamalarını daha da genişleterek GS'lerin bir unsuru olacaklardır.

3.3.3 Yazılım Geliştirme Araçları, Diller ve Platformlar

Mikroişlemci ve bilgisayar tabanlı sistemlerin geliştirilmesi genellikle bir programlama dilinin seçildiği yazılım geliştirme ve sistem davranışının yapılandırılması ve uygulanması anlamına gelmektedir. Programlama, ilk bilgisayarların icadından bu yana çok değişmiştir. Orijinal programlar, bugünün ayrıntılı veri tabanları, kelime işlemcileri, zaman planlayıcıları ve aksiyon oyunları ile karşılaştırıldığında oldukça basit ve anlaşılır idiler [7].

Gittikçe karmaşıklaşan bilgisayar programlarını yazmak için farklı bilgisayar dilleri geliştirilmiştir. Bu bilgisayar dilleri, konuşma dillerine ne kadar yakın olduklarına ve bilgisayarın kendi iç dilinden ne kadar uzak olduklarına göre sınıflandırılabilirler.

- *Makine dili* veya *1ND* (ilk nesil dili), bilgisayarın merkezi işlemci birimi veya CPU dilidir. En düşük seviyeli dil olup 0 ve 1'lerden oluşmaktadır.
- *Assembly dilleri* veya *2ND* (ikinci nesil dilleri), programın anlaşılmasını basitleştiren ve makine dilini temsil etmek için kısa komut veya harfleri (mnemonik) kullanır.
- *Yüksek seviyeli diller* veya *3ND* (üçüncü nesil dilleri), program deyimleri-kelimeler ve cebir tipi ifadeler kullanır. Bir program üst düzey dillerinden birinde yazıldıktan sonra, ya derlenmeli (makine dilinde tekrar yazılmalı) veya yorumlanmalı (tercüme edilmeli) dir. COBOL, FORTRAN, BASIC, PASCAL, PL/I, C ve JAVA, 3ND'lere örnek programlama dilleridir.

- *Oldukça yüksek seviyeli diller* veya *4ND* (dördüncü nesil dilleri), nesne yönelimli, veri tabanı sorgu dilleridir. Programcılar için daha az seçenek olmasına rağmen, program yazmak daha alt düzey dillere göre daha kolaydır. Bunlarında, derlenmesi veya yorumlanması gereklidir. Bu dile temel örnek SQL (Structured Query Language-Yapılandırılmış Sorgu Dili) dir.

- *Doğal diller* veya *5ND* (beşinci nesil diller), deyimlerin normal cümleler olarak yazılabildiği dillerdir. Şu anda, doğal dil kullanan bir programlama dili yoktur.

Programlama dillerini sınıflandırmanın diğer bir yolu daha vardır. *Görsel programlama dili* (GPD), kullanıcının iki (ya da daha fazla) boyutlu bir şekilde bir programı belirlemesini sağlar. Derleme veya yorumlama işlemleri geleneksel metin dillerini tek boyutlu karakterler dizisi olarak işlediği için bu diller iki boyutlu olarak dikkate alınmamaktadır. Visual Basic, Visual C++, Delphi, Kylix ve diğer Visual ailesi isimlerine rağmen, görsel programlama dilleri değildirler. Bir GPD, görsel deyimlerle programlama yapmamızı sağlar: metin ve grafiksel sembollerin mekânsal düzenlemeleridir. GPD'ler daha da ileri olarak, kullanılan görsel ifade tipi ve kapsamına göre; simge tabanlı, form tabanlı ve diyagram diller olarak sınıflandırılabilirler. Program yapımı için bazı özel yersel dilbilgisine göre interaktif bir şekilde kullanıcı tarafından manipüle edilebilir ortamlar, grafiksel veya simgesel elemanlar sağlar. Hyperpascal, Cube, VIPR, ARK, CODE 2.0, Fabrik, Tinkertoy, Pict ve Prograph görsel programlama dillerine ait bazı örneklerdir.

Bu farklı yaklaşımların tümünün amacı aynıdır: programcılar için programlamayı daha kolay hale getirmek ve uzman olmayan programcılara erişebilirliktir. Bazıları, hızlı prototipleme (ön örnekleme) ve hızlı uygulamalar geliştirmek için kullanılırlar; diğer bazıları sistemler veya uygulamalar tasarımı için kullanılırlar ve geri kalan diğerleri ise dağıtım için tek başına uygulamalar üretebilirler.

Visual Basic, Visual C, Delphi, Kylix ve diğerleri gibi bazı *Hızlı Uygulama Geliştirme (HUG)* araçları, daha yüksek bir soyutlama seviyesinde yazılım geliştirmeye yardımcı olan güçlü dillerdir. Bunlar, programlama ara yüzlerini daha kolay yapmak için grafiksel bir GUI oluşturucu kullanan, metin tabanlı dillerdir. Bunlar, kullanıcının ekrandaki grafiksel nesnelere üzerinde belirli işlemler gerçekleştirdiği zaman program kodunun kısımlarını çalıştırmasıyla, özellikle olay güdümlü programlarda Windows ara yüzleri geliştirmek için ilginçtirler. Bunlar, prototipleme ve ev içi uygulama programı geliştirmek için yaygın bir biçimde kullanılmaktadır. Prototipler, özellikle insan-makine ara yüzler ve temel işlevler ile ilgili olarak, kullanıcı ile gereksinimleri ortaya çıkarmak için güçlü araçlar olabilirler. Bunlar, geliştirme sürecinin erken aşamalarında kullanıldıklarında, yazılım kalitesini iyileştirebilir ve geliştirme maliyetlerini azaltabilirler; bununla birlikte, kod devamlılığının kötü olduğu iddia edilmektedir. Her durumda bunlar,

kullanıcıların çabalarını bir programlama dili hakkında öğrenmekten ziyade, problemlerini çözüme kullanmalarını sağlamaktadır. Başarıları ve uygulamaların artan karmaşıklığı, web ve dağınık sistemlerde dahil olmak üzere, .NET ve J2 gibi yeni ve daha tam platformların gelişmesini teşvik etmiştir.

.NET

.NET, internet olanakları ve Windows ortamı ile ilgili hemen hemen de her şeyi bünyesinde barındıran, Microsoft'un yeni bir programlama modelidir [8]. İçinde çoğu programlama görevlerinin kolay bir biçimde yapılabilir olduğu, tamamen yeni bir altyapı tabanlı olup en başından geliştirilmiş olduğu iddia edilmektedir. Veri erişimi, pencereleme, internete bağlanma ve Win32 API'nın çoğu işlevselliğini de kapsayacak şekilde Windows üzerinden yapılacak şeyler şimdi oldukça basit nesne modeli aracılığı ile çoğunlukla erişilebilir. VB (Visual Basic) dili büyük çapta yükseltilmiş, böylece şimdi önceden C++'da erişilebilir çoğu özellikler ve sınıfları içermiştir. C++'ın yeterliliği ile VB'nin bazı gelişme kolaylıklarını birleştiren yeni bir dil, C Sharp veya C# geliştirilmiştir. .Net uygulamaları için bellek yönetimi çok daha karmaşık olup, bunun anlamı, kötü biçimde davranan bir .NET bileşeni aynı süreçte çalışan diğer unsurların çökmesine nadiren neden olur. ASP'nin yerini ASP.NET (Active Server Pages-Aktif Sunucu Sayfaları) almıştır. ASS, genellikle Activex komutlarını (genellikle de VB Script veya Jscript kodu) kullanan, dinamik olarak yaratılan .ASP uzantılı bir web sayfası için bir özelliktir. Bir web tarayıcı bir ASP sayfası talep ettiğinde, web sunucusu HTML kodu ile bir sayfa oluşturur ve bunu web tarayıcısına geri gönderir. Bundan dolayı ASP'ler CGI komutlarına benzerler, fakat onlar Visual Basic programcılarının aşına oldukları araçlarla çalışmalarına olanak sağlarlar.

Yeni ASP.NET, derlenmiş web sayfaları (web isteklerinin işlenmesini çok daha verimli yapan) sunar ve yaygın olarak kullanılan HTML biçimi ve kullanıcı ara yüzü unsurları oluşturabilir büyük sayıda önceden yazılmış unsurları içerir. Ana programlama dilleri, oldukça birbirine yakın hareket ettirilmiştir. Bundan dolayı VB, C++ ve C#'de yazılmış kodlar karışmış olabilir. Unsurlar, unsurların kullanımını ve yüklenmesini oldukça kolaylaştıran, son derece kendi kendini tanımlayan, bileşen adı verilen yeni bir birimde toplanmıştır.

.NET mimarisinin en önemli özelliği, C# ve VB'deki kod, doğal çalıştırılabilir normal işletilebilir bir kod yerine bir Ara Dil'e (IL, Intermediate Language) derlenir, normal olarak çalıştırma sırasında olan en son adımda ise doğal çalıştırılabilir forma dönüştürülür. Böyle bir koda Yönetilen Kod (Managed Code) C++ denir. Bu da sizin C++ kodunuzun VB ve C# ile birlikte çalışmasını sağlar ve size tüm .NET özelliklerinden yararlanma avantajı sağlar, fakat .NET de desteklenmeyen bazı C++ özelliklerinin (örneğin çoklu kalıtım gibi) kullanımını kısıtlar.

Java

Bir başka popüler programlama dili Java ve onun yeni Platform Sürümleridir. Java, modern programlama süreçlerindeki birçok problemi çözmek için tasarlanmıştır. Java; basit, nesne yönelikli, ağa hazır, yorumlanmış, sağlam, güvenli, mimari bağımsız, taşınabilir, yüksek performanslı, çok kullanımlı, dinamik bir dildir. Java, C++ gibi dillerin kafa karıştırıcı, nadiren kullanılan, anlaşılması zor birçok yapı özelliğini ihmal eder. Basit olmasının diğer bir yönü, küçük makinalarda çalıştırılabilir yazılımın tertibini sağlayacak kadar küçük olmasıdır. Ayrıca, Java yorumlayıcısı ve standart kütüphaneler küçük bir alan kaplar, öyle ki bunlar küçük bir disk veya bellek alanı gerektirirler. Java, nesne yönelimli olup, bu da ara yüzlerin temiz (şifresiz) tanımını kolaylaştırdığı ve C++ araçlarına dinamik olarak bağlanmak için yeniden kullanılabilir yazılım sağlamayı mümkün kıldığı için, Java oldukça güçlüdür. Java'nın http ve FTP gibi TCP/IP protokolleri ile kolayca başa çıkmak için kapsamlı bir rutinler kütüphanesi vardır. Bu, ağ bağlantılarını kolaylaştırır ve yerel bir dosya sistemine ulaşırken ki gibi, aynı yolla URL'ler vasıtasıyla web üzerindeki nesnelere ulaşan ve açan uygulamalar oluşturur. Java ile bir uygulamanın aynı versiyonu tüm platformlarda çalışır. Belirli bir bilgisayar mimarisi ile ilgili olmayan genel bytecode komutlarının kullanımı, uygulamayı taşınabilir yapar. Şaşırtıcı olarak, Java yüksek bir performansa sahiptir. Bu, uygulamanın çalışmaya devam ettiği özel CPU için bytecode'un çalışma sırasında makine koduna dönüştürülmesiyle başarılı.

Son zamanlarda, Sun Microsystems, Java platformu mimarisini Java 2 adını vererek yeniden tanımlamıştır. Standart Edition (J2SE), Enterprise Edition (J2EE), ve Micro Edition (J2ME), Java 2 platform kısmının üç ürünüdür. Bu sürümlerin her biri, bir Java sanal makinesi (JVM), Java programlama dili, teknolojiler ve her ürün için temel olan özelliklerden ibarettir.

J2SE, JavaOne tabanlı olup, bireysel masaüstü ve işyerlerinde çalıştırılmak üzere optimize edilmiştir. J2SE; Java Foundation Classes (JFC) API'yı, Java tak çalıştır yazılımını, uluslararası desteği, heterojen ortamlarda çalıştırılabilme desteğini, bir 2D API'yi, yeni bir güvenlik modelini ve Java HotSpot performans motorunu içerir.

J2SE temeli üzerine inşa edilen J2EE; Enterprise JavaBeans unsurları, Java Servlets API'yı, JavaServer Pages'i ve XML teknolojisini tam destek sağlamaktadır. J2EE standart, uygulamaların J2EE'yi destekleyebilen mevcut işletme sistemlerinin geniş bir dizisinde taşınabilirliğini sağlamak için tam özelliklere sahiptir.

J2ME; cep telefonları, çağrı cihazları, kişisel dijital sekreterler, ekran telefonları, dijital kiosklar ve otomobil sistemleri gibi oldukça küçük ve kısıtlı hafızalı aygıtlar için optimize edilmiş çalıştırma ortamıdır. J2ME'nin en önemli bileşeni çok küçük alanlı K sanal makinesidir (KVM). Bu konuda en önemli şey, küçük aygıtların masaüstü bilgisayarlar ve büyük kurumsal sistemlerle bağlanmasıdır.

3.3.4 Nesne Tabanlılık ve UML (Unified Modeling Language): Bileşik Modelleme Dili

Nesne Tabanlılık

Yazılım geliştirme için geleneksel yaklaşım, temel ilkenin fonksiyon veya işlem grubunun olduğu algoritmik bir bakış açısına dayanır. Yazılımın özelliği, çalıştırılması beklenen fonksiyonlara dayanır ve geliştirme süreçleri her fonksiyonu kolayca uygulanabilir seviyede daha küçük fonksiyonlara bölmeye yöneliktir. Bu yaklaşımın en büyük problemi, sistem gereksinimleri değiştikçe ve sistem büyüdükçe fonksiyonlar değişikliklerden oldukça etkilenebilir. Bir sonuç olarak, sistemi korumak ve geliştirmek çok zordur.

Nesne tabanlı yaklaşım, yazılım geliştirme için blokların meydana getirilmesinin nesnelere veya nesne sınıfları olmasını önermektedir. Bir *nesne*, problem etki alanına (örneğin bir traktör) veya çözüm etki alanına (örneğin bir iletişim kutusu) ait bir şeydir. Bu yaklaşımın avantajı, problemi modellerken, nesnelere fonksiyonlara göre daha stabil veya bizlerin onlar için sahip olduğu kullanımları olmasıdır. Diğer bir deyişle, problem için önemli olan şeyler genellikle fonksiyonlardan daha az değişir, yani, manipüle edilecek şeyleri istediğimiz biçimde gerçekleştirir. Nesne tabanlılığı, nasıldan ziyade daha çok ne üzerine yoğunlaşır ve bu daha karardır. Bu, aynı zamanda, problemin ve onun modellerinin yazılım uzmanı olmayanların bile daha kolay anlamasını sağlar, çünkü o problem alanını insanın düşündüğü biçimde, yani nesnelere açısından modeller. Sonuç olarak, bu, yazılım müşterileri ve geliştiricileri arasında daha az yanlış anlamaya yol açar.

Bir *sınıf* ise, bazı özellikleri paylaşan nesnelere ortak kümesinin bir tanımıdır. Örneğin, bir traktör sınıfı, onlar için farklı değerlere (motor gücü, tekerlek sayısı, vb.) sahip olsa da, özelliklere ait tipleri paylaşan traktör nesnelere (örnekler olarak adlandırılan) kapsar. Nesnelere bir *kimliğe* (örneğin isimlerinden dolayı, diğerlerinden ayırt edilebilirler), bir *duruma* (veri ile ilişkili) ve davranışa (nesneye ne yapılabilir, veya o diğer nesnelere ne yapılabilir) sahiptir. Hepsi, nesne ile ilişkilidir ve onların nesne içinde kapsanması açısından veri ve davranış arasında hiçbir ayırım yoktur. Herhangi bir nesne, diğer bir nesnenin fonksiyonlarına (davranış) ve/veya verisine ancak onun arabirimi üzerinden erişebilir ve bundan dolayı o, diğer nesnelere içsel uygulamalarını bilmez. Buna *kuşatma* denir ve nesne tabanlı yazılım geliştirme için en önemli yönlerinden biridir. Diğer iki özellik ise kalıtım ve çok biçimlilik. Kalıtım, bir sınıfa üst sınıfının özelliklerini devralmasına izin verir; örneğin bir tekerlekli traktörler sınıfı, bir üst sınıf traktörün özelliklerini devralabilir (ve diğer özelliklerde daha da özel olabilir). Çok biçimlilik ise, farklı nesne sınıflarında farklı eylemler gerçekleştirmek için aynı işleme olanak sağlayan özelliktir; örneğin, “baskı” gibi bir eylem, nesne sınıfına bağlı olarak bir harita veya bir raporun yazdırılmasıyla sonuçlanabilir. Nesne tabanlılık, aynı zamanda, hem nesnelere iddia edilen istikrarı hem de değişikliklerin daha sınırlı olmaları nedenlerinden dolayı, sistemlerin

korunabilirliğini arttırır. Onlar sadece nesnelere içerisinde meydana gelebilmelerine rağmen, kuşatma veya kapsama özelliğinden dolayı sistemin geri kalanını etkilemeyeceklerdir. Son olarak, nesne tabanlılık, unsur tabanlı yazılım geliştirmek için kavramsal temel temin eder, bu yazılımın yeniden kullanımı için önemli bir durumdur.

UML: Kısa Bir Tarihçe

Nesne tabanlı modelleme dilleri, yazılım uygulamalarının karmaşıklığındaki artış ve yeni nesne tabanlı programlama dilleri ile rekabet etmek için farklı analiz ve tasarım yöntemlerine bir ihtiyaç gereği olarak 1970'li yıllara dayanmaktadır. 1990'lı yılların ortalarına gelindiğinde, nesne tabanlı yöntemlerin sayısı 50'den fazlaya ulaşmıştır. Bunların hiçbirinin insanların ihtiyaçlarını tam olarak karşılayamaması kullanıcıların kafasını karıştırdı. Çok öne çıkan yöntemlerden üç tanesi; Booch (Grady Booch tarafından), Ivar Jacobson'un OOSE (Nesne Tabanlı Yazılım Mühendisliği) ve James Rumbaugh'un OMT (Nesne Modelleme Tekniği) dir. Onlar geliştikçe birbirlerinden fikir alışverişine de başlamışlardır. Bu yazarlar, daha sonra birleşik bir yöntem ve modelleme dili fikrini geliştirmeye başladılar. Bu fikir ortak çaba ve birikmiş deneyim sayesinde daha hızlı bir biçimde olgunlaşabilmiştir. Bu, daha güçlü ve kullanışlı yazılım araçlarının gelişmesini sağladı. Büyük firmalardan bireysel işbirlikçilere kadar diğer ortaklar da, süreci hemen desteklediler ve o zamandan bu yana birkaç sürüm piyasaya çıktı. Bu arada UML, Object Management Group (OMG) tarafından 1997 yılında bir standart olarak benimsendi ve onun geliştirilmesinden bir Revision Task Force (Revizyon Görev Gücü) sorumludur [10]. UML'yi kullanan birçok kitap ve yazılım aracı mevcut olup, bu nesne tabanlı modelleme için bir fiili standart haline gelmektedir.

UML'nin Temel Özellikleri

UML, yazılım yapısını tanımlamak için standart bir dildir. Bu dil, yazılım tabanlı sistemlerin dokümantasyonu, yapımı, tanımlanması ve görselleştirilmesi için kullanılabilir. Bu dil, belirli bir gelişim sürecine bağlı olmayıp ve sadece bir geliştirme yönteminin bir parçasıdır [10].

Bir modelleme dili olarak, yapısının farklı görünümüne odaklanarak, kelimeleri ve kuralları, bir sistemin temsili için tasarlanır. Bu, gruplar arasındaki iletişimi ve anlayışı kolaylaştırmada yardımcı olan grafiksel bir dildir. Gösterimi anlamsal olarak iyi tanımlandığı için modellerinin tam, hassas ve kesin olduğu bakımından özellikler sağlar. Bu modeller, yazılım geliştirme sürecinin farklı aşamaları için uygulanırlar. Bir programlama dili, hatta görsel bir programlama dili dahi olmasa da o, UML modellerinden programlama dili yapılarına eşleştirme olanağı sağlar. Bunun tam tersi de mümkündür. Bir dokümantasyon dili olarak, sistem yapısını, detaylarını, gereksinimlerini ve testlerini açıklamak veya ifade etmek

için bir araç sağlar. UML, aynı zamanda proje planlama ve sürüm yönetimi desteği de sağlar.

Dilin kavramsal bir modelini meydana getiren üç ana unsur vardır: UML yapı blokları, blokları birleştirme kuralları ve UML içinde geçerli mekanizmalar.

Üç tip yapı bloğu vardır: nesnelere, ilişkiler ve diyagramlar. *Nesneler*, bir modeldeki ana unsur veya bileşenler olup, UML'nin temel nesne tabanlı unsur veya bileşenleridir. Onlar, bir modelin statik kısımlarını (yapısal nesnelere), dinamik kısımlarını (davranışsal nesnelere), örgütsel kısımlarını (nesnelere gruplama) ve açıklama kısımlarını (açıklama notu nesnelere) temsil edebilirler. İlişkiler, nesnelere arasındaki temel ilişkileri tanımlar. UML'de dört tip ilişki mevcuttur: bağımlılık, birleşme, genelleme ve gerçekleştirme. *Diyagramlar*, bir sistemin farklı görünümünü gösteren genellikle nesnelere ve ilişkilerin, eleman kümelerinin grafik temsilleridir. Diyagram çeşitliliği (dokuz farklı tip mevcuttur), aksi taktirde açık olmayacak bir sistemin farklı yönlerini vurgulamak açısından faydalıdır. Dokuz tip diyagram: sınıf, nesne, kullanım durumu, sıra, işbirliği, durum grafiği, aktivite, bileşen ve dağıtım şemasıdır. UML kuralları, kendi içinde tutarlı, ilgili olduğu tüm modeller ile uyumlu, iyi biçimlendirilmiş modelleri elde etmek için modellerin kurulumu için semantik (anlambilim) tanımlarlar. Örnek olarak, isimler (nesnelere ve ilişkiler için) için semantik kuralları.

UML, yeni bir dil olup fakat yazılım modellemesi için güçlü, tutarlı ve kapsamlı kavramlar ve gösterimler sağladığı için artan derecede kullanılmaktadır. Buna ek olarak, pek çok ticari vasıtalar onu desteklemekte ve onun kullanımını ve diğer yazılım geliştirme vasıtalarıyla entegrasyonunu kolaylaştırmaktadır. Bununla birlikte, iyi tanımlanmış bir metotla kullanılması gerekli olup yazılım mühendisliği sürecinin sadece (önemli bir parçası olmasına rağmen) bir kısmıdır. Bu, özellikle durum odaklı kullanım, mimari tabanlı, yinelemeli ve artan bir süreçte yeterlidir. UML ve bunun resmi teknik özelliklerine www.omg.org web adresinden ulaşılabilir.

3.3.5 XML: Veri için Evrensel Dil

XML (Extensible Markup Language) veya *Genişletilebilir İşaretleme Dili*, veri tanımlamak için yeni bir etiket tabanlı dil olarak özetlenebilir. Bu, belge veya doküman yapısını veya içeriğini açıklayan karmaşık bir standart olan Standart Genelleştirilmiş İşaretleme Dili (SGML, Standard Generalized Markup Language) nin bir alt kümesidir. Bu, 1998 yılında bir öneri olarak çıkmış ve hala yeni işlevler ve özellikler ile gelişmekte olan World Wide Web Consortium (W3C) [11] un XML Working Group'u tarafından denetlenen kişisel olmayan bir şartnamedir.

Bu bir metadildir (diğer dilleri tanımlamak için bir dil). Bunun anlamı bu dilin belirli uygulama alanları (örneğin hassas tarım gibi) ve onların belge sınıfları için özelleştirilmiş biçimlendirme dillerini tanımlamada kullanılabileceğidir. Bu, veri

sunumu için (örneğin HTML, Hypertext Markup Language gibi) bir dil olmayıp, veriyi organize etmek için bir dildir.

SGML, HTML ve XML

SGML, 1986 yılında yayınlanan ve belge yapısının tanıtımı için bir metot ve belgelerdeki açıklayıcı biçimlendirme için bir format başlatan bir ISO standardıdır (ISO 8879). Bu, yazılım ve donanım bağımsız belgelerin oluşturulmasını sağlar ve bir dizi belge yapısını destekler. Bununla birlikte, oldukça genel olduğu için aynı zamanda karmaşık ve onunla işlem programlarını uygulamak zordur.

HTML, veri sunumuna yönelik tek bir SGML özelliğine uygun etiketlerin oldukça kısıtlı alt kümesini kullanan SGML'nin bir uygulamasıdır. Bu dilin sadeliği, web-yayın dili olmasını sağlamasına rağmen, sabit biçimleri faydalılığını kısıtlamaktadır.

XML, SGML'nin sadece en önemli ve basit özelliklerini içermesinden dolayı, daha anlaşılır ve uygulama geliştirmek için kullanımı daha kolaydır ve web üzerinden birlikte çalışabilirlik ve servis için daha yeterlidir. HTML ile karşılaştırıldığında XML, veriye içerik ve anlam ekleyerek, belgenin içeriğine odaklanır. O, kullanıcıya verinin mantıklı ve planlı bir şekilde temsil edilebilmesi için kullanılacak etiketleri kullanıcının belirlemesini sağlar.

Basitliği, veriyi kapsayacak bir biçimlendirme dili oluşturmak için onun basit kurallarının olmasından kaynaklanmaktadır. Genellikle çift olan etiketler, belgedeki bilgiyi açıklarlar, böylece o neredeyse kendini kendi tanımlayan olur. XML veri, etiketler arasında düz metin olarak saklanır, böylece o, herhangi bir standart metin editörü ile düzenlenebilir. O, tüm önemli dilleri destekleyen bir karakter kodlama standardı olan Unicode Standard'ı destekler ve XML'nin hemen hemen tüm dünya karakterlerini kabul etmesini sağlar. Bu, farklı kültürler ve ülkeler tarafından erişilebilir uygulamaları geliştirmek için özellikle ilginçtir. Onun belgeleri köklü bir ağaç yapısına sahip olup bu birçok uygulama için karmaşık verileri temsil etmede güçlü, aynı zamanda işlem programlarını uygulamak ve işletmek kolaydır.

Bir belge tipi tanımı (DTD, Document Type Definition), mevcut etiketleri, onların olabilecekleri yerleri ve onları bir arada nasıl kullanılacağını belirterek, bir dilbilgisi veya bir XML belge yasal yapısı tanımlar. Bir DTD kullanmak zorunlu olmasa da o, belge adlandırma ve yapımında etiketleri geçerlilik ve standardizasyon açısından kontrol imkanı sağlar. DTD'lerin bir gelişmiş hali XML Schema dilidir. Bu dil, bir XML belgesinin çeşitli unsurları ve özellikleri için geçerli yapıyı, kısıtları ve veri tiplerini belirtir. Veri yazımı, önemli yeni bir özellik olup tüm veriyi düz metin olarak değerlendirme yerine o, XML belgelerinde uygun söz ve anlam dizimi yaptırımını sağlar.

XML belgelerini kullanmak için bir uygulama programı bir ayrıştırıcıya (parser), aralarında bir ara yüz olarak bir yazılım modülüne, belgeleri okumaya ve

uygulama programının belge içeriği ve yapısına erişimi sağlamasına ihtiyacı vardır. Genellikle, farklı dillerde ve ücretsiz ayrıştırıcılar mevcuttur.

Uygulamalar ve Kısıtlamalar

XML, içerik tanımını ekran yönergelerinden ayırır. Böylece aynı içerik, kişisel bilgisayarlar, kişisel sayısal yardımcılar (PDAs, Personal Digital Assistants) ve cep telefonları gibi farklı aygıtlar veya platformlarda aynı belgeye farklı stil çalışma sayfaları (Extensible Stylesheet Language veya XSL) uygulayarak kolayca kullanılabilir.

XML belgeleri aynı zamanda meta bilgiyi de (bilgi hakkında bilgi) içerdiği için online bilgi arama ve alma kolaydır.

XML'nin temel özelliklerinden biri de, onun farklı platform ve organizasyonlar arasında bilgi alışverişi potansiyelinin olmasıdır. Onun metin tabanlı olması, tüm platformların onu kolayca anlamasını sağlayabilir. Onun, veri tanımlaması için evrensel bir dil olduğu iddia edilmektedir. O, aynı zamanda internet üzerinden bilgi aktarımını otomatikleştirerek ortaklar (örneğin tarımsal kuruluşlar veya şirketler) arasında belirli XML sözdizimi tanımları sağlar.

Bununla birlikte XML, tek başına olan sistemler için muhtemelen iyi bir seçenek değildir. O, tek başına herhangi bir güvenlik özellikleri sağlamaz. Bu, bazı ekstra mekanizmalar (kriptografi, dijital imzalar, vb.) kullanılmadığı sürece, internet gibi ortak bir ağda bir problem olabilir.

Son olarak, standart kelime bilgileri veya etiket kümeleri eksik olup verilerin hatalı olarak yorumlanmalarını önlemek için en azından belirli sektörlerde (bu zaten halihazırda meydana geliyor) bunlar geliştirilmelidir. Örneğin, Geography Markup Language, veya GML'in geliştirilmesi gibi. Bu, coğrafik özelliklerin hem geometrisini hem de özelliklerini içeren (<http://www.opengeospatial.org/specs/> web sayfasını ziyaret ediniz) coğrafik bilgilerin depolanması ve taşınması için bir XML kodlamasıdır.

XML, birçok uygulama için önemli bir dil haline gelmekte olup muhtemelen özellikle internette bilginin kullanılma ve teslim edilme biçimini de değiştirecektir. Bu hala gelişme aşamasında ve yeni özellikler dahil edilmektedir. XML hakkında daha fazla bilgiye <http://www.w3.org/XML> web adresinden ulaşılabilir.

3.3.6 Açık Kaynak Yazılımı

Birkaç şirket, yazılım ve internete hakim olma yönünde dünyayı tehdit etmektedir. Bunların hakimiyetlerine karşı en güçlü rakip, kitlesel olarak *açık kaynak yazılım* (AKY) adı verilen ücretsiz yazılım araçları ve işletim sistemleridir. En ticari yazılımların aksine, bu tür yazılımların çekirdek kodları kolayca özelleştirilebilir, değiştirilebilir ve geliştirilebilir.

Açık kaynak, sadece kaynak koduna erişim anlamına gelmez. Açık kaynak yazılımın dağıtım koşulları, aşağıdaki kriterlere uygun olmalıdır [12].

1. Ücretsiz yeniden dağıtım: Lisans; birçok farklı kaynaktan olan programları içeren toplu bir yazılım dağıtımının bir unsuru olan yazılımın, dağıtımı veya satışından herhangi bir kesimi kısıtlamamalıdır.
2. Kaynak kodu: Programın kaynak kodu olmalı, ve kaynak kodlu ve aynı zamanda derlenmiş hali ile dağıtımına izin verilmelidir.
3. Türetilmiş eserler: Lisans, eğer orijinal yazılım lisansı ile aynı şartlarda dağıtılırsa, değişikliklere ve türetilmiş eserlere izin vermesi gerekir.
4. Yazarın kaynak kodunun bütünlüğü: Lisans, kaynak kodun sadece orijinal şekli ile dağıtılması konusunda kısıt koyabilir. Yol dosyaları, türetilmiş veya değiştirilmiş kodlar kendi lisansları ile dağıtılmalıdır.
5. Kişi ve gruplara karşı ayırım yapmama: Lisans, hiçbir kişi veya gruba ayrımcılık yapmamalıdır.
6. Mesleklere göre ayırım yapmama: Lisans, programın belirli bir uğraşı alanındaki herhangi biri tarafından kullanımını kısıtlamamalıdır.
7. Lisans Dağıtımı: Program ile ilgili haklar, ek bir lisansa gerek olmadan programın dağıtıldığı tüm kesimler için geçerlidir.
8. Lisans bir ürüne özgü olmamalıdır: Program için geçerli olan haklar, belirli bir yazılım dağıtımının bir parçası olan programa bağlı olmamalıdır.
9. Lisans diğer yazılımı kısıtlamamalıdır: Lisans, lisanslı yazılımla dağıtımı yapılan diğer yazılıma kısıt koymamalıdır.

Günümüzde AKY olarak kabul edilebilir pek çok ürün ve kendilerini bu ürünlerin geliştirilmesine adanmış grup vardır [13]. En popülerlerinden biri Linux İşletim Sistemi ve onun ilgili uygulama yazılımıdır. Çoğu UNIX tabanlı işletim sistemleri, AKY özelliklerine uymaktadır ki buda bazen UNIX'i AKY'nin babası olarak dikkate alınmasını sağlar. İnternet, yoğun miktarda ticari kullanımlı açık kaynaklı yazılımla dolu: Apache, dünyanın yarıdan fazla web sunucusunda çalışır; Perl, World Wide Web'in motorudur ve FreeBSD, MySQL, Kylix, Java ve son zamanlarda MS ofise benzer Open Office gibi diğerleri.

Açık kaynak yazılımı, dünya yazılım tekeline karşı olan yiğit bir savaşı gibi görünüyor olabilir. 2002 yılında, Windows, Sun Solaris ve diğer UNIX türevleri gibi diğer tüm büyük işletim sistemlerinin pazar payı azalırken, Linux'un web sunucu işletim sistemleri arasındaki pazar payı %29 dan %34'e yükseldi [14].

Bazı piyasa araştırmacıları önümüzdeki birkaç yıl içerisinde açık kaynak ve ücretsiz yazılımın işletim sistemlerinde aynı zamanda yaygın kullanılan emtia yazılımının çoğunda standart olacaklarını tahmin etmektedirler. Bu, aynı zamanda çoğu bilgisayar kullanıcısının ya yeni işletim sistemi ve emtia yazılımına geçeceği, ya da Microsoft'un bir Open Source (açık kaynak) lisansı adı altında Windows yayınlacağı gibi bu iki şeyden birinin olabileceği anlamındadır. Eğer insanlar işletim sistemlerini değiştirirlerse, en olası hedefin Linux olacağı görünüyor ve diğer olasılıklar ise FreeBSD veya BeOS'un bir açık kaynak sürümü olabilir. İkinci olasılık ile ilgili olarak ta, Microsoft yeni trendi benimseyebilir ve ona atlar veya

kullanıcıların AKY'lere geçişinin güçlü olmasından dolayı o böyle yapmak zorunda kalabilir. Çoğu insan her iki senaryoyu da olasılık dışı görüyor, ancak AKY'lerin popülerlik ve pazar payı kazanıyor olduğuna şüphe yoktur.

3.3.7 Sonuç

Yazılımın, çok daha geniş bir halk kesimine daha kapsamlı ve kullanımı kolay programları getirmesi ve birtakım karmaşıklık ve zorlukla daha geniş bir uygulama alanında uygulanması ile son yıllarda çok geliştiğinde şüphe yoktur. Bununla birlikte, diğerleri arasında yöntem, teknoloji, uyumluluk, entelektüel mülkiyet ve maliyet ile ilgili de daha fazla ele alınması gereken birçok konu vardır.

Yazılım çok önemli bir endüstri haline geldi ve bu, programlamanın ilk günlerinden beri meydana gelen ana değişikliklerden biridir. Büyük kaynaklar içeren dünya çapında bir iş olarak o, bu pazarla ilgili birçok insanı ve şirketi cezbetmektedir. Öte yandan, bilgi ve tecrübe paylaşımı ile ilgilenen büyüyen bir ücretsiz yazılım topluluğu, yazılım geleceğinin ne olabileceği ile ilgili olarak bir alternatif senaryo sunmaktadır.

Kaynaklar

1. Davis, S. J., J. MacCrisken, and K. M. Murphy. 1999. The evolution of the PC operating system: An economic analysis of software design. Accessed on March 03, 2005, from <http://gsbwww.uchicago.edu/fac/steven.davis/research/>.
2. Hansmann, U., L. Merk, M. S. Niklous, and T. Stober. 2003. Pervasive Computing, 2nd ed. New York, NY: Springer Professional Computing.
3. The Palm computing platform: An overview. Accessed on March 03, 2005, from <http://www.wirelessdevnet.com/channels/pda/training/palmonoverview.html>.
4. Microsoft Windows CE: An overview. Accessed on March 03, 2005, from <http://www.wirelessdevnet.com/channels/pda/training/winceoverview.html>.
5. Li, Y., M. Potkonjak, and W. Wolf. 1997. Real-time operating systems for embedded computing. International Conference on Computer Design (ICCD'97).3.4 Artificial Intelligence Methodologies 153
6. Rammig, F. J. 2001. Real time operating systems: Tutorial. XIV Symposium on Integrated Circuits and System Design (SBCCI 2001).
7. Fenton, N., and S. L. Pfleeger. 1994. Science and substance: A challenge to software engineers. IEEE Transactions on Software Engineering (July): 86-95.
8. Robbins, T. 2000. Introduction to Microsoft.NET. Microsoft Partner Technical Workshop.
9. J2SETM Platform, J2EETM Platform, J2METM Platform documentation. 2002. Sun Microsystems, Inc. Accessed on March 03, 2005, from <http://www.sun.com/java/about/>.
10. Booch, G., J. Rumbaugh, and I. Jacobson. 1999. The Unified Modeling Language User Guide. Reading, MA: Addison Wesley.
11. W3C—World Wide Web Consortium. Extensible Markup Language (XML). Accessed on March 03, 2005, from <http://www.w3.org/XML>.
12. Perens, B. 1997. The Debian free software guidelines. Debian developers manual.
13. The Free Software Foundation. Accessed on March 03, 2005. <http://www.fsf.org/>.
14. Wheeler, D. A. 2002. Why open source software/free software (OSS/FS)? Look at the numbers! Accessed on Marc