EÜ Devlet Türk Musikisi Konservatuarı Dergisi
2015 (6): 47-59

BİLGİSAYAR MÜZİĞİ DİLLERİNİN TARİHÇESİ

Mehmet Can ÖZER*

ÖZET

Bilgisayar müziği, son 65 yılda bilgisayarların gelişmesiyle ortaya çıkan disiplinler arası bir alandır. Eskiden telekomünikasyonda deneysel bir araştırma sahasıyken, sonrasıda çağdaş sanatta popüler bir alan haline gelmiştir. Bu gelişmeler, bilgisayar müziği dillerinin ilerlemesiyle takip edilebilir. Bilgisayarlar çağdaş ve popüler kültürde müzik yapma ve dinleme biçimlerini değiştirmiştir. Bilgisayar müziği kavramı, bu dillerin gelişimi doğrutusunda oluşmuş, özelde elektronik müziğin, genelde çağdaş müziğin bir dalıdır.

Anahtar Kelimeler: bilgisayar müziği, bilgisayar müziği dilleri, elektronik müzik, dijital sanat, çağdaş sanat, ses sentezleme, sayısal sinyal işleme.

A BRİEF HİSTORY OF COMPUTER MUSİC LANGUAGES

ABSTRACT

Computer music is an interdisciplinary field which had arisenin last 65 years with the advances of computer technologies. While it was an experimental research field in the telecommunications, then has become a popular genre in contemporary art. Those deveopments can be traced with the advances of computer music languages. Computers has changed the rules of making and listening to music in both contemporary and popular culture. The term of computer music, which is a branch of electronic music in particular and a genre in contemporary music, has been formed towards the advances in those languages.

Keywords: Computer music, computer music languages, electronic music, digital art, contemporary art, sound synthesis, digital signal processing

Yöntem

Makalede kullanılan ana kaynaklar yabancı dilde ve teknik terimlerden oluştuğundan, bunların güncel karşılıkları için internet Bilişim Sözlüğü (bilisimsozlugu.net) tercih edilmiştir. Literatüre yakın okuyucular için de parantez içinde asıl terimler sunulmuştur. Literatür, bilgisayar müziği için birincil kaynaklar sayılan Computer Music Journal, Leonardo Music Journal ve International Computer Music Conference yayınlarından derlenmiş, gerekli yerlerde kitaplara ve kullanım

* Doç. Dr., Yaşar Üniversitesi, Sanat ve Tasarım Fakültesi, Müzik Bölümü, Öğretim Üyesi. mehmetcan.ozer@yasar.edu.tr

kılavuzlarına da başvurulmuştur. Esas olan, zaman dizinsel olarak bilgisayar müziği dillerinin tarihsel gelişimi ve geçirdiği evrimi yansıtmak, bununla birlikte elektroakustik müzik özelinde katkılarını imlemektir. Özet ve sonuç kısımları bilinçli olarak kısa tutulmuş, özellikle de tarihsel süreci teknik yönüyle ele almaya çalışan bu makalede sonuç ve geleceğe dair öngörülerden ziyade, kısıtlamalardan uzak naçizane tavsiyelere yer verilmiştir. Tarihsel süreç yıl bazında değil, olay bazında yapılandırılmıştır.

Giriş

1951 yılında Remington Rand adlı Amerikan firması tarafından ilk ticari bilgisayarın piyasaya sürülmesinin ardından, tüm dünyada önemi neredeyse tekerleğin icadıyla kıyaslanabilecek bir teknolojik devrim yaşanmıştır. Endüstrileşmiş ülkelerde üretim ve rekabet, denetim sistemlerinin yani otomasyonun hegemonyası altına girmiştir (Manning, 1981: 120). Buna koşut olarak, bilgisayar ve sanat üretimi iki kutupta değerlendirilmektedir: Disiplinler arası evliliklerden kaynaklı yaratıcılığa faydalı tutum, diğeri ise insani ifade ve duyguların hiçbir şekilde bilgisayar dolayımıyla ortaya konmaması. 60 yılı aşkın süreçte gerek müzik gerekse tüm sanatlar, bilgisayarın sunduğu imkânlar neticesinde büyük bir gelişim ve değişim göstermişlerdir. Disiplinlerin sınırları erimiş, ortaya yeni ifade araçları ve yöntemleri çıkmıştır. Bu konuda başat bir rol üstlenen bilgisayar, "dijital sanat" alanında ve müzik özelinde "Bilgisayar Müziği" adlı yeni bir disiplin oluşturmuştur.

Laske'ye göre sanat üretiminde üç temel yaklaşım vardır: Sonuç odaklı (analitik), alet odaklı (pragmatik) ve süreç odaklı (bilişsel). 20. Yüzyılın temel yaklaşımı alet odaklılıktır. Sanatçıların kullanımı için, sanatçılar ya da mühendisler (ya da ikisinin bileşkesi uygulayıcılar) tarafından üretilmiş araçlar, hem sanatı hem de sanatçıları şekillendirmiştir (Laske, 1984: 9). Sinema, yerleştirme (installation) ve video sanatı gibi yeni disiplinler bunlara örnek olarak gösterilebilir. Özellikle bu alanlardaki geçişliliği tanımlamak için video sanatının babası sayılan Nam JunePaik'in aslen elektroakustik müzik eğitimi aldığı ve ardından uygulamalarını görsellere yönelttiği gerçeği önemli bir kanıttır (Serwer, 1994: 87).

Bilgisayar Müziği ilk kez 65 yıl önce, üniversiteler ve araştırma laboratuarlarında üretilmiştir. Bilgisayar müziği terimi, seslerin üretimi, başkalaştırılması ve elde edilen bu seslerin zaman içinde örgütlenmesi için bilgisayar kullanılmasını öngören, çağdaş çoksesli müziğin geldiği noktada ve ona yer yer lokomotif görevi yapan, elektroakustik müziğin bir dalıdır. Üretim alanı yalnız sanatla kısıtlı değildir, disiplinlerarasıdır. Elektronik, fizik, matematik, psikoloji, psikoakustik, müzik kuramı ve bestecilikle iç içe gelişim gösterir. Elektroakustik müzikte ses üretim araçları için bir sınır (bilgisayar, kayıt cihazları, ses sentezleyiciler) olmamakla birlikte, günümüz bilgisayarları artık tüm eski elektroakustik müzik tekniklerini gerçekleştirebilmektedirler. Hatta günümüzde önemli elektroakustik müzik çalışmaları yalnızca bilgisayarlar aracılığıyla yürütülmektedir. Tarihsel olarak bilgisayarların kısıtlamaları bulunduğundan, bu terim farklı kullanımları içerir.

Bilgisayar yardımıyla yapılan her müzik (örneğin film müzikleri, dans müzikleri) ve bilgisayarın müzik bağlamındaki her kullanımı (örneğin nota yazım ya da

bestecilere asistanlık gibi), "bilgisayar müziği" ortaya çıkartmaz. "Bilgisayar müziği" kavramı, bilgisayar destekli müzik (computer aided music) kavramıyla karıştırılmaktadır. Tıpkı "elektronik müzik" kavramının popüler dans müziğiyle karıştırıldığı gibi. Akademik kökenli ve çağdaş çoksesli müziğin devamı olarak nitelenen, bu alanda üniversitelerde eğitim programları olan disiplin, aşağıda betimlenmeye çalışılmıştır.

Besteci Vladimir Ussachevsky'nin tanımına göre "elektronik müzik, sanatsal içerikli bir deyişle müzik yapmak isteyen bestecinin sesleri saptamak, yaratmak, başkalaştırmak ve örgütlemek için çeşitli elektro-akustik aygıtlardan yararlandığı bir yaratı alanı olarak ele alınmalıdır" (Ussachevsky, 1958: 44). Barry Schrader ise "elektronik imkânlarla üretilmiş, değiştirilmiş ve yeniden üretilmiş olan müzik türü elektroakustik müziktir" (Schrader, 1982) der. Tümünün özeti olarak elektroakustik uğraşın asıl amacı, yeni sesler ve bu sesler için yeni bestelerdir, tıpkı önceki bestecilerin "doğal" çalgılarla yaptıkları gibi. Elektroakustik müzik terimi ise ülke/ekol kaynaklı yordamsal farklılıkları bir potada eriten öneri olarak Michel Chion(1982: 14) tarafından "La musique electroacoustique)" kitabında ortaya konmuştur.

Zaman dizinsel olarak, bilgisayar müziğini ve aslen onu var eden bilgisayar müziği programlama dillerini, işlevsellik, kullanılan ana dil ve makineler üzerinden inceleyeceğiz.

Öncüller (1951-1965)

Bir bilgisayar dili, komutlardan oluşmuş yapay bir dildir (Bilişim Sözlüğü, 2014). Bilgisayar dili, soyut bir hesapsal (computational) model ortaya koyarak, çözülecek problemin dışındaki sorunlardan bağımsız olarak program yazmaya olanak tanır. Gücü ise, soyutlama yetisinin gelişkinliğinden gelir. Soyutlama yetisi ne denli gelişkinse, programcının soruna odaklanması o denli kolaylaşır (McCartney, 2002: 61).

Dünyanın ilk bilgisayar müziği, Avusturalya'da CSIRAC adlı bilgisayarın Thomas Cherry tarafından programlanması ile üretilmiştir. Ancak bu durum Avusturalya'da gelişime ve kurumsallaşmaya gitmemiş, yaklaşık 60 yıl kadar tarihin tozlu sayfalarında keşfedilmeyi beklemiştir (Doornbusch, 2004: 11). CSIRAC yardımıyla, onu ziyarete gelenlere bilgisayarın neler yapabildiği gösterilmiş, kuramsal derinliği ve gelecek bağlantısı olmadan popüler şarkılardan oluşan bir seçki, bilgisayar dolayımıyla seslendirilmiştir. 1951 yılına ait BBC kayıtları, 2008 yılında arşivlerde şans eseri bulununca uluslararası bir ilgi oluşturmuştur.

Bilgisayarı bestecilik amaçlı ilk kullanan, Amerikalı genç kimya araştırmacısı ve besteci Lejaren A. Hiller olur. Hiller, stokastik modeller üzerine araştırma yaparken bu yöntemi müzik üzerinde kullanır ve IllionisÜniversitesinde'ki süper bilgisayar ILLIAC'ı (Illinois Automatic Computer) programlayarak, bilgisayarın beste yapmasını sağlar. Notaları yaylı dördül'e çaldırır ve eserin adını "ILLIAC Suite for String Quartett" koyar (Moore, 1996: 28). Kendi ifadesiyle, ortaya çıkan eser estetikten ziyade didaktiktir. "Monte Carlo Yöntemi" denilen bu teknikle, kuralları belirledikten sonra bilgisayar olasılık hesaplarıyla notaları, süreleri ve dinamikleri kararlaştırır. Bu yaklaşım bilgisayarlı algoritmik kompozisyonun da ilk örneği sayılabilir.

Bilgisayarın telekomünikasyon alanındaki geleceğini gören AT&T (American Telephone and Telegraph Company) Bell Laboratuarları, analog teknolojideki gürültü aktarımını sonlandırmak için, dijital aktarımı destekler ve bilgisayar teknolojisinin bu konudaki potansiyelini araştırmaya başlar. Müzik çalışmalarının desteklenmesindeki temel amaç, konuşma ve müziğin yapısal seslerden oluşmasıdır. Genç bir telekomünikasyon mühendisi ve amatör kemancı olan Max Mathews, telefon konuşmaları için analiz, işlem ve sentezleme yapılabileceği gibi, müzikal sesleri için de analiz, işlem ve sentezleme yapılabileceğini ortaya koyar. Bunun ilk örneği olan "MUSIC I" dili, 1957'de doğar ve böylece de bilgisayar müziği sahası da Hiller'in ilk çabalarının somutlaşması anlamında oluşur (Moore, 1996: 31).

MUSIC I (1957) bilgisayarda üçgen (triangle) ses dalgası üretimi yapmaya olanak tanıyan bir dildir. Seslerin perde, genlik ve süre değişkenleri kontrol edilebilir. Üst seviye diller tanımlaması henüz yapılmadığı, tüm programlama bir seferde ve FORTRAN'da (FORTRAN henüz bir yaşındayken) yazılmıştır. MUSIC III (1960) Max Mathews'ün tümüyle yeni bir kavram olan "birim üreteçleri-unitgenerators" geliştirdiği dildir. Aslında program elektronik modüllerin benzeşimidir, neredeyse aynı zamanlarda ortaya çıkan Moog synthesizerlarda olduğu gibi. Ayrıca o zamanlarda bilinen programlama tekniklerinden olan, çeşitli yerlerde kullanılabilen küçük modüler yapılar da ilk kez burada görülmektedir. MUSIC IV (1963) programlanması, Bell Laboratuarları'nın bilgisayarı değiştirmesi sonucu olmuştur. Mathews, MUSIC IV'ün, MUSIC III'den hiçbir farkı olmadığını, yalnızca değişen dil ve bilgisayardan ötürü olduğunu kendisi de belirtmiştir (Mathews, 1978: 32).

SSP (Sound Synthesis Program), Utrecht Üniversitesi Sonoloji Enstitüsü kendi bilgisayarını edindikten kısa bir zaman sonra Gottfried Michael Koenig tarafından yazılır. Analog elektronik müzik stüdyolarında tükenen deneysel keşiflere yardımcı olması için tasarlanmıştır. 1964 yılında Koenig tarafından yazılan algoritmik kompozisyon yazılımı Project 1 ve Project 2'nin sistematik çalışmaları sonucu ortaya çıkan verileri, ampirik buluşlarla sese dönüştürmektir. Koenig, Köln'den Hollanda'ya taşındığında, kendi dili olan Project 1'i tümüyle FORTRAN'dan ALGOL'a geçirmeye mecbur kalmıştır. SSP kullanıcısı genlik ve süre değerleri için iki liste belirler ve seçim yöntemleri istenen miktarda ses kesiti ile karıştırılarak tamamlanır (Banks ve diğerleri, 1979: 2). Bu yıllarda kullanılan bilgisayarların hepsi anaçatı (mainframe)'dir (Lippe, 1996: 22). (Anaçatı bilgisayarlar en az büyük bir salon büyüklüğünde ve tüm duvarlara yayılmış büyük parçalardan oluşmaktadır).

Ses Sentezleme ve Yapıların Belirmesi (1965-75)

Bilgisayar müziğinin ilk sorunlarından birisi, bütünüyle müziksel karakterli ses sentezlemek olmuştur. 1960'ların ortalarında müzik araştırmacıları sentezlenen sesin sorununun rengi (tınısı-timbre) olduğuna varmışlardır. Mathews'in laboratuarında çalışan fizikçi ve besteci Jean-Claude Risset, trompet sesi üzerine araştırma yaparken, önce içgüdüsel keşfettiği ardından bilimsel olarak kanıtladığı ses spektrumunun zamana bağlı olarak doğru bir şekilde değişimi ilkesi, ses sentezlemede yeni bir çağın başlangıcı olmuştur (Moore, 1996: 34). Risset ile çalışan genç fizikçi Pierre Ruiz, bu

gelişmelerin ışığında çok önemli bir ses sentezleme tekniği olan "fiziksel modelleme"yi henüz tanımlamış ve keman sesi üzerinde çalışmaya başlamıştır. Bu olağanüstü yaklaşımı getiren Ruiz, ardından deneysel tiyatro yapmak için bilimi ve tüm çalışmaları bırakmıştır. Ancak Mathews çalışmayı sürdürmüş ve belki de en önemli özelliklerinden biri taşınabilirliği (yani başka bilgisayarda kullanılabilmesi) olan MUSIC V'i geliştirmiştir. 1960'ların sonlarında Stanford Üniversitesi'nde John Chowning tarafından (kendi deyimiyle yanlışlıkla) frekans modülasyon tekniği ses sentezlemeye aktarılmıştır. MUSIC V'in en önemli katkılarından birisi FM tekniğinin oluşmasını sağlayabilecek esneklikte programlanabilmesidir. Bu önemli keşfin ardından, Stanford Üniversitesi tüm haklarını Yamaha Grubuna satarak, dijital müzik dünyasının en çok satan ses sentezcisi YAMAHA DX-7'nin oluşmasına (yaklaşık 500.000 adet) neden olmuştur (Moore, 1996: 41).

1971 dijital devrim yılı olarak nitelendirilir. Sebebi ise MarcianHoff isimli mühendisin Intel Corporation'da mikroişlemciyi icat etmesidir. Bu sayede, bütün bir merkezi işlem ünitesi (CPU), tırnak büyüklüğünde silikon bir çipin içine sığarak, boyutu çok daha küçük ve kişisel bilgisayarların da üretilmesinin önünü açar (Bassett, 2007: 48).

1972 yılında Kanadalı besteci ve programcı Barry Truax tarafından yazılan POD (Poisson Distribution), gerçek zamanlı sentezleme ve etkileşimli besteleme teknikleri üzerinden geliştirilmiştir. PDP-15, HP 2116 ve Nova 3 bilgisayarlarda yazılan bu dil, kullanıcının biraz programlama önbilgisiyle bilgisayar müziği yapmasını hedefleyen bir dildir. Besteleme hiyerarşisi sırasıyla ses nesnesi seçimi, yazım alanı belirtimi, yayılım algoritması ve icra değişkenleri şeklindedir. Bunun sonucunda bestenin bölümü, icra değişkenleri, ses nesnesi ve olay (beste) ortaya çıkar. Truax, POD'un ileriki sürümlerinde Tanecik Sentezleme (Granular Synthesis) tekniğini ortaya koymuştur (Truax, 1977: 32).

Büyük üniversitelerdeki çalışma şartlarının olumsuzlaşmasına koşut olarak, 1970'lerin başlarında bazı besteciler taşınabilir sistemler için tasarımlara başlarlar. Ed Kobrin'in HYBRID sistemi voltaj denetimli osilatörler, yükselticiler ve filtreler için PDP 11/10 bilgisayarda tasarlanmıştır. Bir diğeri ise EMS1 (Electronic Music Studios) sistem ya da HYBRID 0 adıyla bilinen, Moog modülleri ve CA (Computer Associates) mini bilgisayarları için tasarlanan dildir. Bunlar temelde kullanıcıdan alfanumerik klavyeden veri alıp, hızlıca donanım bileşenlerine ileten arayüzlerdir. Kullanımlarındaki zorluklar nedeniyle (temelde dilin komut yazımlarının neredeyse makine dili seviyesinde olması) bestecilerin büyük bölümü bunlardan uzak durmaya başlamıştır (Fuchs, 1988: 41).

Mikroişlemci ve Sinyal İşleme (1975-1985)

Yukarıda sayılan kullanım zorlukları ve kendini açıklamaktan yoksun sistemler, besteci Iannis Xenakis'i bilgisayarda grafiksel girdiyi sese dönüştürmeye yarayan bir sistemi tasarlamaya itmiş ve bilgisayar mühendisi Patrick Saint-Jean ile çalışarak ortaya 1977 yılında UPIC (Unité Polyagogique Informatique du CEMAMu) dilinin prototiplerini çıkartmıştır. Buradaki temel amaç, neredeyse herkesin grafiği çizerek

ses sentezlemesine olanak sağlamaktır. Diğer dillerdeki fonksiyonları tanımlayıp işlemleri ardıllamak yerine, UPIC'de çizilen grafik yazılımı ses üretimine yönlendirir. Örneğin çizilen dikey çizgi, perdenin (ya da gürlüğün) de yükselmesini sağlar. Bu tür çok gelişmiş sistemlerin en büyük sorunu yalnızca tek bir bilgisayarda çalışmalarıdır. Sistemle çalışmak isteyen besteciler ancak CEMAMu (Centre d'Etudes de Mathématique et Automatique Musicales) Paris'te isteklerini gerçekleştirebileceklerdir (Fuchs, 1988: 42).

IRCAM'da (Institut de Recherche et Coordination Acoustique/Musique) 1979-1983 yılları arasında, Xavier Rodet tarafından Sail dilinde PDP-10 bilgisayarda yazılmaya başlanıp, FORTRAN'a aktarılmasıyla önce VAX-11/780 ve ardından 4X'e aktarılan CHANT (Fransızca şarkı) projesi (Favreau, 1986: 370) aslen şarkı söyleyen sesin analiz ve sentezlenmesiyle ilgili tasarlanır. Bu çalışma sonrasında, kurallar temelinde (algoritmik) insan sesi (şarkılayan) sentezlemeye yönelir. CHANT önceleri insan sesi davranışıyla başlar ancak geliştirdikçe model ve analiz olarak kullanımını geçer. Araştırmacılar insan sesinin tek yüzlü ve basit olarak ele alınamayacağını, tam tersine onun zenginliği ve karmaşık yapısının yeni bir ses sentezleme tekniği ve dili olarak ortaya konmasına çalışırlar. CHANT Gerald Bennett, Jonathan Harvey (özellikle Mortuos Plango, Vivos Voco adlı eseri çok önemlidir), Harrison Birtwistle, Gerard Grisey ve daha pek çok besteci tarafından eserlerinde kullanılmıştır (Rodet, 1984: 15).

1970'li yıllarda mikroişlemci devriminin ardından, bilgisayar müziği dilleri kendi özel donanımlarını da geliştirerek ticari boyutta atılımlar yapmışlardır. Burada bilgisayar müziği tanımı ve dilleri kapsamında olsak da, yapılan çalışmaların derinliği ve kısmen elektronik müzik, büyük oranda ise popüler müzik kültürünü etkilemesi açısından, üretilen aygıtların isimlerinden bahsetmek yerinde olacaktır. Orijinal tasarımı ve geliştirmesi John Appleton, Sydney Alonso ve CameronJones tarafından Darthmouth College'de gerçekleştirilen Synclavier, 1973 yılında geliştirilmeye başlanmış ve 1977 yılında piyasaya sürülmüştür (Byrd, 1977: 55). Avusturalya'da ise 1979 yılında Peter Vogel ve Kim Ryrie tarafından geliştirilen Fairlight CMI (computer music instrument) ise ilk dijital örnekleyici olmakla birlikte, pek çok özel ses üretim algoritmasını da barındırmaktadır. 1986 yılında ise kendi ses işlemcisiyle birlikte anılan bir dil olan Kyma, Carla Scaletti ve Kurt J. Hebel tarafından tasarlanır. Günümüzde de üretilen Kyma System, nesne yönelimli bir dil ve onun için tasarlanan Pacarana ses süper bilgisayarı (kendi tanımlarıyla) ile satılmaktadır.

Kişiselleşen Bilgisayarlar ve Ticari Başarılar (1985-1995)

1980'lerde UCSD (University of California, San Diego)'de Dick Moore ve Loy Garreth tarafından açık kaynak kodlu ve taşınabilir (anaçatı bilgisayarlarla sınırlı olmayan) bir ses sentezleme ve sinyal işleme dili/sistemi olarak CARL (computer audio research laboratory) ortaya çıkar. Nedenleri arasında C programlama dilinin ortaya çıkışı ve yaygınlaşması, kişisel bilgisayar kavramını ve UNIX işletim sisteminin kazandırdıkları sayılabilir. CARL ile ilk kez çeşitli kullanıcılar arasında yazılım, eser ve araştırmaların dolaşımı başlar. Otomatik yükleme betikleri de bu dille ortaya çıkan yenikliklerden birisidir. Bu sayede bilgisayar müziği üniversite ve araştırma

merkezlerinin fiziksel şartlarından ayrışarak, bestecilerin ve araştırmacıların kendi mekanlarında da kullanılmaya başlamıştır (Loy, 2002: 53).

1986 yılında CDP (Composer's Desktop Project), İngiltere'de Atari 1040 bilgisayarlarına kişisel erişim kolaylığı dolayısıyla ortaya çıkar. Yazılım GROUCHO (Groucho Marx'a atfen) denilen ses işlemi birimlerinin koleksiyonu olarak ve açık sistem (proje üyelerinden herhangi birinin yeni yazılımlar ekleyebildiği) algısıyla şekillendirilir. UNIX anaçatı bilgisayarlardan devşirilen CMUSIC dili, CDP için başlangıç olmuştur. 1987 yılında resmi kuruluşunu yapan CDP, aynı yıl ilk sürümünü de yayınlar. 2014 yılına dek 7 sürüm gerçekleşmiştir (Fuchs, 1988: 41).

Csound, Barry Vercoe tarafından 1986'da MIT'de (Massachusetts Institute of Technology) geliştirilen bir ses programlama dilidir. C dilinde yazıldığı için ve öncelleri olan "Music N" dillerinden ayrışmak için başında "C" kullanılmıştır. Vercoe ilk geliştirdiği dile, Max Mathews'in ardından Music 11 adını vermiş, daha sonraki geliştirmelerde ise doğrudan Csound ismini kullanmıştır. Açık kaynak kodlu bir dil olan Csound, dünyadan pek çok kullanıcı tarafından geliştirilerek, şu ana dek 1700'ün üzerinde birim üreteci yazılmıştır. İlk sürümünün üzerinden 30 yıla yakın geçmesine rağmen, ilk gün yazılan kodları halen işleyebilmekte, dolayısıyla tutarlı ve kararlı bir çizgide ilerlemektedir. Kod tabanlı bir dil olsa da, 1998 yılından itibaren GUI (grafiksel kullanıcı arabirimi) anlayışı ile kabuklar üretilmiş ve kullanımı gün geçtikçe daha da kolaylaşmıştır. Cecilia (Piché ve Burton, 1998: 55), Blue, CsoundQT gibi sürümlerinin de geliştirmeleri devam etmekte, gerçek zamanlı olarak da kullanılabilmektedir (Vercoe, 1986: 13).

CMIX, Music N ailesinden bir ses sentezleme ve sinyal işleme dilidir. Csound ve CMusic dillerine benzerlikler gösterse de, hepsinin olduğu gibi onun da atası Mathews'in geliştirdiği MUSIC V'dir. Ses dosyalarını işlemek ya da yeni sesler sentezlemek için tasarlanan dil, ham ikili (binary) veriyi de sese ya da tam tersi olarak uygun cihazlarla (A/D çeviricilerle) sesi ham ikili veriye dönüştürmeyi amaçlar. İlk başlarda gerçek zamanlı olmadan tasarlanan dil, Paul Lansky tarafından 1980'lerin başında yazılmıştır. 1995 yılında ise, Brad Garlton ve DaveTopper tarafından yeni sürümü hazırlanan CMIX, platform bağımsız geliştirilme tavrını buradan başlatır. Günümüzde RTCMIX olarak geliştirilen bu dil, gerçek zamanlı ses sentezleme, analiz ve sinyal işleme için kullanılmaktadır (DuBois, 1993: 17).

Veri akışı (dataflow) programlama paradigmasının ortaya çıkışı ile (Johnston ve diğerleri, 2004: 3), 4. Kuşak programlama dilleri de varlığını göstermeye başlamıştır. Bu dillerin temel özelliği, veri akışını takiben özelleşmiş nesnelerin birbirine bağlanması (patching) ve kodlama bilgisi olmadan gerçek zamanlı programlama yapmaya olanak tanımalarıdır. Bilgisayar müziğinde ve sonraları dijital sanatta çığır açan dillerden biri MAX olmuştur.

MAX (Puckette 1988; Opcode 1990: 13, 2), gerçek zamanlı müziksel yazılımları üretmeye yarayan bir grafiksel programlama dilidir. Nesne yönelimli bir mimaride başlayan dil ilk kez Apple Macintosh bilgisayarda yazılmış ardından NeXT bilgisayara IRCAM Music Workstation'un bir parçası olarak geçirilmiştir (Lindemann, 1991: 50). İlk kez veri ve sinyal akışını birleştirip denetlemek için kullanılan bu dil, tarihsel olarak

MIDI üzerinden geliştirilmeye başlanmıştır. MIDI üzerinden ilerlemenin temel nedeni IRCAM'ın 4X işlemcileri ancak Machintosh ile MIDI seri bağlantısı protokolünden bağlantılanabilmesidir (Puckette, 1991: 70).

MAX'in temel kavramı Patch'dir (yama). Patch, birbirlerine çizgilerle bağlanan kutulardan oluşur. Kutular birbirlerine mesajlar iletmeyi sağlayan nesneleri temsil eder. Kutuların giriş ve çıkışları olabilir (Puckette, 1991: 74).

David Zicarelli MAX'in Machintosh'a aktarımında çok büyük katkı sağlamıştır. Frederic Durieux, Michael Jarrell ve Philippe Manoury gibi besteciler MAX'i ilk kez IRCAM'da çalıştıkları süre boyunca eserlerinde kullanmışlardır. Tabii ki MAX, Max Mathews'un anısına isimlendirilmiştir. Aynı zamanda "gerçek zamanlı çizelgeleme yaklaşımı" (Mathews ve Pasquale, 1981: 286) RTSKED programı, MAX tarafından sahiplenilip kullanılmıştır (Puckette, 1991: 76). David Zicarelli tarafından ticari lisansla üretilmeye başlanan Max, Miller Puckette tarafından IRCAM'da JMax adıyla 1998 yılında açık kaynak kodlu olarak yazılsa da devamı gelmemiştir.

MAX'in açık kaynak kodlu kuzeni PD (Pure Data), Miller Puckette tarafından 1990'larda geliştirilmeye başlanmıştır. Halen geliştirilen ve yaygın bir şekilde kullanılan PD, temelde MAX ile aynı işlevselliğe sahip olmasına rağmen, açık kaynak kodlu yazılımların tümünde görülen "uzmanlık beklentisi", okul dışı kullanıcıları pek tatmin etmemekte, ücretli MAX'a adeta yönlendirmektedir. Burada sözü edilen eksiklik temelde dokümantasyon üzerinedir.

MAX'in bir programlama dili olup olmadığına dair çeşitli tartışmalar süregelmektedir. MAX, üretenlerin ifadeleriyle "medya için görsel programlama dili"dir (Cycling74.com, 2014: 1). Günümüzde 7. sürümünün yayınlandığı bu yapıyı dil olarak kabul etmemizin temel nedeni, yukarıdaki bilgisayar dili tanımına uymasıdır. "Programcının soruna odaklanması" kavramını gerek grafik arayüzü, gerekse de desteklediği Java, Ruby, Python ve C dilleriyle entegre çalışması ve 6. sürümden itibaren gelen "Gen" desteği ile rahatça sağlayan, istenirse kendi başına çalışan (standalone) programlar üretebilen yapısı dolayısıyla MAX, bir programlama dilidir.

Güncel Örnekler (1995 ve sonrası)

1997'de Richard Dannenberg tarafından yazılan Nyquist, ses sentezleme ve besteleme için ileri seviye işlevsel bir dildir. Nyquist'in amaçlarından biri "Music N" diye anılan dillere oranla, sistem kaynakları kullanımı açısından daha verimli olmaktır. Aynı zamanda tüm dillerdeki yazım kuralları (syntax) mekansal ve zamansal olarak kaynak kullanımını olumsuz etkilediğinden, Nyquist bu sorunlar göz önüne alınarak tasarlanmıştır. Özellikleri arasında artımlı hesaplama (incremental computation), devingen bellek ve geri çağırma, yeni sinyallerin devinimli simgelenmesi, sonsuz seslerin simgelenmesi, çok kanal desteği ve çoklu örnekleme hızı sayılabilir. Nyquist, önceli sayılabilecek Arctic (Dannenberg, McAvinney ve Rubine tarafından 1986 yılında), Canon (Dannenberg tarafından 1989 yılında) ve Fugue (Dannenberg, Fraley ve Velikonja tarafından 1991 yılında) dillerinin halefidir. Sayılan tüm bu diller, sinyal işleme ve ses sentezlemede besteci odaklı yani müzikal malzemeler (melodi, akorüretimi) olsa da, pratik kullanımda pek çok kısıtlama getirmekteydi. Örneğin

Canon örneklenmiş sesleri işleyemiyor, Fugue ise geliştirmesi ve hafıza kullanımı oldukça sorunlu bir dildi. Nyquist, Lisp tabanlı yapısıyla, ses sentezlemeyi anında yaparak, hafıza kullanımını büyük ölçüde çözümler. Temelde Music V'den Csound'a kadar tüm işlem algısını değiştiren ilk dil olması açısından önemlidir (Dannenberg, 1997: 81).

Music N ailesinin soyutlamaları (Csound dahil), birim üreteçler, ses örneği hesaplama döngüsü, birim üreteçleri ve enstrümanların (dile özgü soyutlama olarak kullanılmıştır) temsil edilmeleri ve serbest bırakılmalıdır. Bu soyutlamalar sinyal işleme algoritmalarını kolaylaştırır. Bununla beraber, Music N ailesi dillerde gerçek anlamda denetim yapıları ve kullanıcı fonksiyonları sağlamaz. MAX ise, insanların programlama yaptığını bilmeden programlama yapmasına olanak sağlamaktadır. Nesne yönelimli dillerin devingen veri ilişkilendirme, devingen veri tipleri özellikleriyle donatılmıştır. Bu da nesneleri aslında statik kılmaktadır. Bu eksiklerden hareketle, James McCarthy tarafından 1997'de yazılmaya başlanan Super Collider, besteleme ve sinyal işleme için gerekli soyutlamaları en basit ve doğrudan haliyle ve dinamik değişkenlerle sunmaya yönelik tasarlanmıştır. Eğer bir kullanıcı değişmez bir sonuç istiyorsa, Music N dillerinin sunduğu geleneksel nota/orkestra yapısı yeterli olacaktır. Super Collider'ın temel ilkesi, geleneksel müzikte olduğu gibi her icranın değişken olması ve icracı/besteciye gerçek zamanlı işlemler yapabilmesini sağlamaktır. Super Collider, temelde üst-seviye dil ve sentezleme sunucusu olarak ikili bir algıyla geliştirilmiştir. Sentezleme çalışırken, yeni modüller yaratılabilir, yok edilebilir, yeniden bağlantılanabilir. Sinyal işleme, sinyal akışının içine devinimsel olarak ve çizgelenerek/zaman planına dahil edilerek (scheduled) sağlanabilir. Tüm komut istemleri TCP ya da UDP gibi OSC'ün basitleştirilmiş halleriyle iletilmektedir. İlk iki sürümü ücretli olan SuperCollider, 2002 yılındaki üçüncü sürümü itibariyle CC lisansı ile ve ücretsiz olarak dağıtılmaktadır (McCartney, 2002: 68).

Veri akışı programlama paradigmasına karşıt olarak, 2002 yılında GRAME (Fransa)'da geliştirilen FAUST (Functional Audio Stream), blok diyagramı yapısında gerçek zamanlı ses sentezleme ve sinyal işleme dilidir. FAUST kodları tümüyle derlenir, yorumlanmaz. Derleyici, yazılan kodu en etkili şekilde C++ koduna çevirir. Dolayısıyla yazılan tüm programlar, başka kütüphanelerden bağımsız ve çok hızlı çalışır. Yazılan kodlar örnek (sample) seviyesinde çalıştığından, düşük seviye DSP fonksiyonları için uygundur. Yazımı yalnızca akademik çevreler için olmadığından, basit ve iyi tanımlıdır. Temel bakış açısı, sinyal işlemeyi matematiksel bakış açısıyla tatminkar bir biçimde ifadelendirmek ve uygulama detaylarından kaçınmaktır (Orlarey ve diğerleri, 2002: 3).

Bilgisayar destekli bestelemede, Common Music gibi notasyonla birlikte, sessel çıktıdan ziyade çalgısal bestecilik üzerine yardımcı araçlar olarak geliştirilen diller de mevcuttur. Common Music, CCRMA'da LISP içerisinde geliştirilen bir besteleme programıdır (Taube, 1991: 30). Bu örneği, ses sentezleme ve sinyal işlemeyle bir araya getiren ise PWGLSynth olur. Finlandiya Akademisinin desteğiyle yürütülen proje, ilk kez 2003 yılında PW adıyla ve ardından PWGL (open GL desteği ile) ile sürülmüştür. Düşük maliyetli donanımların gerçek zamanlı ses sentezlemeyi ve sinyal işlemeyi çok daha güçlendirmesiyle, PatchWork (Laurson ve diğerleri, 2005: 34), Open Music

(Assayag, 1999: 3) gibi gerçek zamanlı olmayan dillerin çoğunlukla notasal bestecilik destekleri, PWGLSynth ile hem sessel çıktıya hem de denetim birimlerine dönüştürülebilir. C ve Lisp temelli iki bileşenden oluşan dilde, ses sentezleme, gerçek zamanlı sıralama, ardıllama (sequencer), Midi desteği ve ses işlem kütüphaneleri bulunmaktadır. Kullanıcılara görsel ses sentezleme ve sanal çalgı tasarımı deneyimini büyük bir başarıyla aktaran dil, açık kaynak kodludur (Laurson ve diğerleri, 2005).

"Canlı Kodlama" ilk kez 1985 yılında Ron Kuivila tarafından STEIM (Amsterdam)'da gerçekleştirilir. Kuivila, Formula programlama dili ile 30 dakikalık bir icrada bulunur. Buradaki amaç, dinleyicinin hem ekrandan yaratımı takip edebilmesi, hem de şahit olduğu kodlamayı dinleyebilmesidir. 1980'lerin sonunda bir avuç insanın yaptığı, 1990'larda duraksayan bu akım, Super Collider'in gerçek zamanlı yapısı nedeniyle yeniden canlanır (Gresham, 1998: 43). Canlı kodlamaya yönelik geliştirilen diller ise ChucK (2002) ve Impromptu (2005) olmuştur.

ChucK, 2002 yılında Ge Wang ve Perry R. Cook tarafından Princeton Üniversitesi'nde geliştirilmeye başlanmıştır. ChucK'ın amacı eşzamanlı olarak karmaşık ses sinyallerinin işleyebilme ve sentezleyebilme, esnek ve incelikli bir zaman kullanımı, çoklu, değişken ve devingen kontrol, canlı kodlama yapmaya olanak sağlayan bir yapı olarak dört maddede özetlenebilir. ChucK, Linux, Windows, Solaris ve MacOS işletim sistemlerinde çalışabilir. Birçok girdi (Midi, Serial, USB, grafik, vb.) doğal desteklenmektedir. Canlı kodlama doğası sayesinde, çalışırken yapılan değişiklikler icrayı hiçbir şekilde etkilememektedir. Ücretsiz ve açık kaynak kodludur (Wang ve diğerleri, 2003: 2).

Canlı kodlama için 2005 yılında Andrew Sorensen tarafından yazılan bir diğer dil ise Impromptu'dur. Laptop performansından hareketle ve algoritmaların canlı olarak tasarlanmasını öngören bu dil devingen, gerçek zamanlı, çok kullanıcı destekli olarak tasarlanmıştır. Sinyal işleme ve ses sentezleme motoru, gerçek zamanlı çizelgeleme (scheduling) motoru, şema yorumlayıcı ve bütünleşik geliştirme ortamından oluşur. Impromptu üç ayrı yolun kesişmesinden ortaya çıkmıştır: Devingen diller ve etkileşimli geliştirme ortamlarına olan ilgi, programlamayı canlı icra aracı olarak görme arzusu ve AIME (Sorensen'in geliştirdiği yeni C++ sentezleme/çizelgeleme motoru). Geliştiriciler, sonraki sürümlerde yapay zeka kütüphaneleri üzerinden yeni bir paradigma çabasındadır. Ücretsiz ve açık kaynak kodludur (Sorensen, 2005: 7).

2012'de NickCollins tarafından geliştirilmeye başlanan NSound ve 2013'de Ronald Schlenker tarafından geliştirilmeye başlanan BYOND (Build Your Own Device), açık kaynak kodlu ve dünyadan kullanıcı ve geliştiricilerin beğenisine sunulan en yeni proto diller olarak karşımıza çıkmaktadır. Veri akışı programlama tekniği ve MAX'in yöntemleri doğrultusunda, Reaktor, Audio Mulch, Bidule ve Usine gibi çok amaçlı, gelişkin ses üretme yazılımları üretilerek, ticari lisanslar olarak son 10 yıldır satılmakta ve sürekli geliştirilmektedir. Ayrıca popüler elektronik dans müziği türleri için özelleşen, Ableton Live ve Bitwig Studio yazılımları da, sağladığı kullanım kolaylıkları ve programlama dilleri uyumları nedeniyle hem akademik hem de popüler müzik camiaları tarafından kullanılmaktadır.

Sonuç

Bilgisayarlar, sanat üretiminde doğrudan ya da dolaylı olarak yer almaktadır. Günümüzde mikroişlemci kullanılmayan bir sanat etkinliği bulmak neredeyse olanaksızdır (Burada etkinlik sanat eserinin yalnızca üretim yöntemi olarak değil, ona erişim süreci olarak da ele alınmıştır). Dikkat edilmesi gereken noktalardan biri, aracın amaç olmaya başlamamasıdır. Üretilen teknoloji ne kadar gelişkin ve ilgi çekici de olsa, araçtır. Sanat eseri üretmek için tek bir araca bağımlı olmak, gitgide o aracın fetişine dönüşebilir. Matematikçilerin bir formülün öncelikle "estetik" olması beklentisi (King, 1997: 28) ya da programcıların kodlarının "şık" olması için çabalamaları, yalnızca görsel zevkin tatmini değil, algının yani bilişsel sürecin de üretimden hoşnutluğunun gerekliliğidir. Müzik gibi teknik ve estetiğin bileşkesi olan bir dalda, bilgisayar gibi son derece teknik bir aygıtı kullanmak sorumluluk gerektirir. Fordizmin teknolojiyi herkes için, tüketilebilir ve ucuz hale getirmesiyle, insanlarda yeteneğin ön koşul olmadan yalnızca teknolojik aygıtlarla sanat üretilebileceği sanrısı gelişmiştir. Sanat her zaman olduğu gibi şekil değiştirmiş, yeni aygıtları kullanıp bu aygıtların doğasındaki estetiği ve ifadeyi aramış, bunu da sanatçılar eliyle gerçekleştirmiştir. Bilgisayarla müzik yapabilmek için öncelikle müzik yapabiliyor olmak, ardından da bunu bilgisayar edimiyle gerçekleştirmek gereklidir. Bilgisayar müziği tarihindeki örneklerin sanat eseri olarak kabul edilenlerinin, müzik hakkında eğitimi, becerisi ve ciddi çabası olanlar tarafından üretildiğinin unutulmaması gerekir.

Bilgisayar programcılarının bildiği "kağıt üzerinde çalışan program bilgisayarda da çalışır" düsturu, müzik için de geçerlidir. Sözü edilen, algoritmaların yani fikir ağlarının sağlamlığıdır. Problem çözümü kağıt üzerinde (ya da herhangi bir ortamda) işliyorsa, uygulaması da çalışacaktır. Buradaki metafor, fikrin uygulamanın üst çatısı olarak kullanılması yönündedir. Örnek olarak, somut müzik (musique concréte) ile başlayan doğrudan ses ile inşa edilen müzik, günümüzde akusmatik (acousmatic) müzik ile devam etmekte ve gerekmedikçe kağıt üzerinde notlara başvurmamaktadır. Üstelik günümüz sayısal teknolojisini hem geliştiren hem de kullanan unsurları bir potada eritir. Ancak bestecileri teknolojinin yönlendirmesiyle değil, teknolojiyi yönlendirerek müzik bestelemektedirler. Bu tavrı, elektroakustik ya da çalgısal çağdaş çoksesli müzik bestecilerinin eserlerinde görmek mümkündür. Sanat eseri, tüm deneylerin ardılı olarak tutarlı bir yaklaşım ve yapı barındırmalıdır.

Kaynakça

ABBOTT, C. (1978)."A Software Approach to Interactive Processing of Musical Sound", *Computer Music Journal*, Vol. 2, No. 1, ss. 19-23, USA: The MIT Press.

ASSAYAG, G. (1999)."Computer Assisted Composition at IRCAM: From PatchWork to Open Music", *Computer Music Journal*, Vol. 23/3, USA: The MIT Press.

BASSETT, Ross Knox (2007)."To the Digital Age", ss. 48, JHU Press.

BYRD, Donald (1977). "An Integrated Computer Music Software System", *Computer Music Journal,* Vol. 1, No. 2, ss. 55-60, USA: The MIT Press.

BANKS, J.D. Berg, P., Rowe, R., Theriault, D. (1979). "SSP. A Bi-Parametric Approach to Sound Synthesis", *Institute of Sonology,* Utrecht.

CHION, Michel (1982). *La musique electroacoustique*, Fransa : Presses Universitaires de France.

DANNENBERG, Roger B. (1997). "The Implementation of Nyquist, A Sound Synthesis Language", *Computer Music Journal,* Vol. 21, No. 3, ss. 71-82, USA: The MIT Press.

DOOMBUSCH, Paul (2004). "Computer Sound Synthesis in 1951: The Music of CSIRAC", *Computer Music Journal,* Vol. 28, No. 1, ss. 10-25, USA: The MIT Press.

FAVREAU, E. (1986). "Software Developments for the 4X Real-Time System", *Proceedings of the International Computer Music Conference,* ss. 369-373, San Francisco: Computer Music Association.

FUCHS, Mathias (1988). "Computer Music Languages... And the Real World**",** *Leonardo Supplemental Issue,* Vol. 1, Electronic Art , ss. 39-42, USA: The MIT Press.

GRESHAM, L. (1998). "The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Computer Music", *Leonardo Music Journal,* Vol. 8, ss. 39-44.

JOHNSTON, W. M., Hanna, J. R. P., Millar, R. J. (2004). "Dataflow Programming Paradigm, Advances in Dataflow Programming Languages", *ACM Computing Surveys,* 36, ss. 1-34.

KING, Jerry (1997). *Matematik Sanatı,* Ankara: Tübitak Yayınları.

LAURSON, M., Norilo V., Kuuskankare, M. (2005). "PWGLSynth: A Visual Synthesis Language for Virtual Instrument Design and Control", *Computer Music Journal,* Vol. 29, No. 3, ss. 29-41, USA: The MIT Press.

LASKE, Otto (1984). "Definitions of Computer Art", *Computer Music Journal,* Vol. 8, No. 4, ss. 9-11, USA: The MIT Press.

LENT, Keith, Pinkston, Russell, Silsbee, Peter, (1989). "Accelerando: A Real-Time, General Purpose Computer Music System", *Computer Music Journal,* Vol. 13, No. 4, ss. 54-64, USA: The MIT Press.

LINDERMANN, E. (1991). "The Architecture of the IRCAM Musical Workstation", *Computer Music Journal,* Vol. 15, ss. 41-50, USA: The MIT Press.

LIPPE, Cort (1996). "Real-Time Interactive Digital Signal Processing: A View of Computer Music Source", *Computer Music Journal,* Vol. 20, No. 4, ss. 21-24, USA: The MIT Press.

LOY, Gareth. (2002). "The CARL System: Premises, History, and Fate", *Computer Music Journal,* Vol. 26, No. 4, Languages and Environments for Computer Music, ss. 52-60, USA: The MIT Press.

MANNING, Peter (1981). "Computers and Music Composition", *Proceedings of the Royal Musical Association,* Vol. 107, ss. 119-131, Taylor & Francis Ltd.

MATHEWS, M., Pasquale J. (1981). "RTSKED a Scheduled Performance Language for the Crumar General Development System", *Proceedings of the International Computer Music Conference,* s. 286, San Francisco: Computer Music Association.

MATHEWS, Max (1978). *Computer Music Journal,* Issue 1978/4, s. 32, USA

MCCARTNEY, James, (2002), "Rethinking the Computer Music Language: Super Collider", *Computer Music Journal,* Vol. 26, No. 4, ss. 61-68, Languages and Environments for Computer Music, USA.

MOORE, F. Richard (1996). "Dreams of Computer Music: Then and Now", *Computer Music Journal,* Vol. 20, No. 1, ss. 25-41, USA: The MIT Press.

OPCODE, Inc. (1990). *MAX Documentation*, Palo Alto, California: Opcode, Inc.

ORLAREY Y., Fober, D., Grame, S. L. (2002). "An Algebra for Block Diagram Languages".

PICHEJ., Burton A. (1998). "Cecilia: A Production Interface to Csound", *Computer Music Journal,* Vol. 22, No. 2, ss. 52-55, USA: The MIT Press.

PUCKETTE, Miller, (1988). "The Patcher", *Proceedings of the International Computer Music Conference,* ss. 420-425, San Francisco: Computer Music Association.

PUCKETTE, Miller (1991). "Combining Event and Signal Processing in the MAX Graphical Programming Environment", *Computer Music Journal,* Vol. 15, No. 3, ss. 68-77, USA: The MIT Press.

RODET, X., Potard Y., Barrière, Jean-Baptiste, (1984). "The CHANT Project: From the Synthesis of the Singing Voice to Synthesis", *Computer Music Journal,* Vol. 8, No. 3, ss. 15-31, USA: The MIT Press.

SCHRADER, Barry (1982). *Introduction to Electroacoustic Music*, Prentice Hall.

SERWER, J. D. (1994). "Nam June Paik: Technology", *American Art,* Vol. 8, No. 2, ss. 87-91, The University of Chicago Press.

SORENSEN, A. (2005). "Impromptu: An interactive programming environment for composition and performance".

TAUBE, Heinrich. (1991), "Common Music: A Music Composition Language in Common Lisp and CLOS", *Computer Music Journal,* Vol. 15, No. 2, ss. 21-32, USA: The MIT Press.

TRUAX, Barry (1977). "The POD System of Interactive Composition Programs", *Computer Music Journal,* Vol. 1, No. 3, ss. 30-39, USA: The MIT Press.

USSACHEVSKY, Vladimir (1958). "Music in the Tape Medium", *The Julliard Review*.

VERCOE, B. (1986). "CSOUND: A Manual for the Audio Processing System and Supporting Programs", *Program Documentation*. MIT Media Lab., Cambridge: Massachusetts.

WANG, G., Cook, P. R. (2003). "ChucK: A Concurrent, On-the-fly, Audio Programming Language", *Proceedings of the 2003 International Computer Music Conference*.

İnternet Kaynakları

Bilişim Sözlüğü, (2014). www.bilisimsozlugu.net, (erişim tarihi 16 Nisan 2014).

Cycling74.com, (2014). Max, a visual programming language for media (erişim tarihi 22 Ağustos 2014).

DuBois, Luke (1993). "A Brief and Largely Vague History of Cmix", http://music.columbia.edu/cmix/history.html, (Erişim Tarihi 22.02.2014).