

161828

YILDIZ TEKNİK ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ

MODERN DANSTA SES VE GÖRÜNTÜ ETKİLEŞİMİ İÇİN
KURGU PROGRAMI TASARIMI

Bilgisayar Müh. Hüseyin KUŞCU

SBE Sanat Ana Sanat Dalı Sanat ve Tasarım Yüksek Lisans Programında
Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Prof. Dr. B. Tevfik AKGÜN

İSTANBUL, 2005

İÇİNDEKİLER

Sayfa

ŞEKİL LİSTESİ.....	iv
ÇİZELGE LİSTESİ	vi
ÖNSÖZ	vii
ÖZET	viii
ABSTRACT.....	ix
1. GİRİŞ	1
2. TEKNOLOJİ VE MULTİMEDYANIN DANSTA KULLANIMI	2
2.1 Genel Bakış.....	2
2.2 Dans Verisinin Sahneden Alınması İçin Kullanılan Donanımlar	4
2.2.1 Sınıflandırma.....	4
2.2.2 Kamera	5
2.2.3 Sahne Sensörleri.....	7
2.2.4 Mikrofon	8
2.2.5 Kas / İskelet Sensörleri	9
2.2.6 Beyin Sensörleri.....	10
2.3 Verinin Sahneye Geri Verilmesi (çıkış).....	11
2.3.1 Tanım	11
2.3.2 Projektörle Yansıtma	11
2.3.3 Ses Sistemi	12
2.3.4 Aydınlatma Sistemi.....	12
2.3.5 Dekor Sistemi.....	12
2.4 Dans Teknolojisi’nde Kullanılan Yazılımlar	12
2.4.1 Tanım	12
2.4.2 EyesWeb	13
2.4.3 EyeCon.....	14
2.4.4 Interactor LPT	15
2.4.5 Isadora.....	16
2.4.6 E-Merge	17
2.4.7 BigEye.....	17
2.4.8 Cyclops	17
3. ÖNERİLEN SİSTEM	19
3.1 Özellikler.....	19
3.2 Yapı.....	21
3.3 Sistemin Çalışması.....	22
3.3.1 Genel Yapı	22
3.3.2 Kontrol Modülü	23
3.3.3 Giriş Modülü.....	29

3.3.4	Çıkış Modülü	30
3.4	Dağıtık Yapı	31
3.4.1	Genel Mantık	31
3.4.2	Mesaj Yapısı	33
3.4.3	Ajan Programı	34
3.5	Sahne Etkileşimi Tasarım Programı	35
3.5.1	Arayüzün Özellikleri	35
3.5.2	Arayüzün Kullanımı	36
3.5.2.1	Yeni Çıkış Kanalı Yaratma	36
3.5.2.2	Yeni Süreç Yaratma	39
3.5.2.3	Yeni Etkileşim Yaratma	42
3.5.2.4	Yeni Geçiş Koşulu Yaratma	47
3.5.2.5	Derleme ve Dağıtma	51
3.5.2.6	Çalıştırma	53
3.5.3	Teknik Ayrıntılar	54
4.	SONUÇLAR	65
KAYNAKLAR		66
ÖZGEÇMİŞ		68

ŞEKİL LİSTESİ

Şekil 2.1 Yer sensörü	7
Şekil 2.2 Dance Dance Revolution kontrolörleri	8
Şekil 2.3 Standart bir mikrofon.....	9
Şekil 2.4 Hey You Pikachu oyunundan bir ekran görüntüsü.....	9
Şekil 2.5 Kalp atışı sensörü ve sensörlü eldiven.....	10
Şekil 2.6 EEG sensörü	11
Şekil 2.7 EyesWeb programının arayüzü	14
Şekil 2.8 EyeCon programının arayüzü	15
Şekil 2.9 Interactor LPT programının arayüzü	16
Şekil 2.10 Isadora programının arayüzü	17
Şekil 3.1 Etkileşimin genel yapısı.....	19
Şekil 3.2 Sistemin genel yapısı	21
Şekil 3.3 Sistemin modüler yapısı	22
Şekil 3.4 Örnek senaryonun zamanlama yapısı	24
Şekil 3.5 Sonlu durum makinası örneği	25
Şekil 3.6 Sistemde durum makinasının kullanımı	26
Şekil 3.7 Durum makinasının sabit matris şeklinde gösterimi	27
Şekil 3.8 Durum makinasının süreçlerle birlikte kullanımı	28
Şekil 3.9 Birden fazla çıkış kanalının kullanımı	29
Şekil 3.10 Giriş modülü	30
Şekil 3.11 Çıkış modülü.....	31
Şekil 3.12 Çıkış modüllerinin veri yığınları	31
Şekil 3.13 Dağıtık yapı	33
Şekil 3.14 Ağ mesajı yapısı	34
Şekil 3.15 Ajan programının arayüzü	35
Şekil 3.16 Boş tasarım programı.....	37
Şekil 3.17 Yeni kanal yaratma arayüzü	38
Şekil 3.18 Yaratılan kanallar	39
Şekil 3.19 Yeni süreç yaratma arayüzü	41
Şekil 3.20 Yaratılan yeni süreçler.....	42
Şekil 3.21 Boş süreç penceresi	44
Şekil 3.22 Yeni etkileşim yaratma arayüzü - 1	45
Şekil 3.23 Yeni etkileşim yaratma arayüzü - 2	46
Şekil 3.24 Yaratılan yeni etkileşimler.....	47
Şekil 3.25 Yeni geçiş koşulu yaratma arayüzü - 1	49
Şekil 3.26 Yeni geçiş koşulu yaratma arayüzü - 2	50
Şekil 3.27 Yaratılan yeni geçiş koşulları	51
Şekil 3.28 Derle ve Dağıt arayüzü	53
Şekil 3.29 Çalıştırma arayüzü.....	54
Şekil 3.30 Borland C++ Builder'da arayüz elemanlarının temel özellikleri	55
Şekil 3.31 Süreç elemanının yapısı.....	56
Şekil 3.32 Hesaplama kullanılabilecek parametreler	57
Şekil 3.33 Sol tutacaktan çekilirken fare sol tutucaktaysa.....	58
Şekil 3.34 Sol tutacaktan çekilirken fare çıkış kanalındaysa.....	58
Şekil 3.35 Sol tutacaktan çekilirken fare süreçteyse.....	59
Şekil 3.36 Sağ tutacaktan çekilirken fare sağ tutucaktaysa	59
Şekil 3.37 Sağ tutacaktan çekilirken fare süreçteyse.....	60
Şekil 3.38 Sağ tutacaktan çekilirken fare çıkış kanalındaysa	60
Şekil 3.39 Süreci taşıırken	61

Şekil 3.40 Zaman sürgüsünün kullanımı62

Şekil 3.41 Durum makinasının iki boyutlu dinamik dizi şeklinde gösterimi63

Şekil 3.42 Durum makinası dinamik veri yapısı.....63

Şekil 3.43 Durum makinasının sabit matris şeklinde gösterimi64



ÇİZELGE LİSTESİ

Tablo 2.1 Kamera karşılaştırması6

Tablo 3.1 Sonlu durum makinasının tablo gösterimi.....25



ÖNSÖZ

Bu tez çalışması Yıldız Teknik Üniversitesi Sosyal Bilimler Enstitüsü Sanat Anasanat Dalı Sanat ve Tasarım Yüksek Lisans Programı'nda hazırlanmıştır. Programın kuruluş ana fikirlerinden biri olan disiplinlerarası çalışma düşüncesi altında programa dahil olan her öğrenci kendi alanının diğer uzmanlık alanlarıyla kesişimini keşfetmek için çeşitli çalışmalar yapmaktadır. Bu tez çalışması da böyle bir çalışmanın geliştirilmiş ürünüdür. Bu tezin hazırlanmasında büyük emeği geçen tez danışmanım Sn. Prof. Dr. B. Tevfik AKGÜN'e teşekkürü bir borç bilirim.

Haziran 2005

Bilg. Müh. Hüseyin KUŞCU



ÖZET

Modern dansın bilgisayarla etkileşimi için dünya çapında pek çok çalışma yapılmaktadır. Genelde bu çalışmalar dansçının hareketinin algılanması ve algılanan harekete göre etkileşimin gerçekleşmesi şeklinde olur. Bu şekildeki tek boyutlu ve lineer etkileşimler uygulamada esneklik getirmemekte ve kareografin hayal gücünü kısıtlamaktadır.

Tez çerçevesinde sanal performanslarda kullanılabilecek bir sistem ve kareografin eldeki sesli ve görüntülü lineer etkileşim seçeneklerini kurgulayıp bu sistem için çıktı alabileceği bir bilgisayar programı hazırlanmıştır. Ayrıca, üzerinde çeşitli denemeler yapabilmek için basit bir sesli/görüntülü etkileşim sistemi tasarlanmıştır.

Anahtar Kelimeler: Etkileşim kompozisyonu, etkileşimli performans, sanal performans, dans teknolojisi, dans ve multimedya, sanal ortam, kareografı.



ABSTRACT

There are numerous studies around the world about the interaction between computer and modern dance. Generally, these studies consist of perception of dancers' movements and generating the interaction using these perceptions. This kind of one dimensional and linear interaction doesn't bring any elasticity in application and limits choreographer's imagination.

By the frame of this research work, a system that can be used for virtual performances and a computer application is prepared for choreographers to edit the audiovisual linear interaction choices in hand. Additionally, a simple audiovisual interaction system is designed for experimental purposes.

Keywords: Interaction composition, interactive performance, virtual performance, dance technology, danceand multimedia, virtual environment, dance, choreography



1. GİRİŞ

İnsan bedeninin bilgisayarla etkileşimi çok yeni bir konu olmamakla birlikte dünya çapında pek çok grup bu konu üzerinde çalışmalar yapmaktadır. Özellikle performans sanatlarında dansçının bilgisayarla etkileşimi pek çok grup tarafından incelenmekte, hatta yüksek bütçeli ticari sistemler hazırlanıp satılmaktadır. Hareket serbesiyeti sağlaması açısından genelde dansçı-bilgisayar etkileşimi görüntü yoluyla yapılmakta, dansçının sahneden alınan görüntüsü sayısal ortamda çeşitli aşamalardan geçirildikten sonra ortaya çıkan bilgilerden faydalanılarak çeşitli efektler meydana getirilmektedir. Yalnız bu etkileşim çeşidi tek boyutlu ve lineer olmaktadır. Dansçı bir efektten diğerine geçememekte, efektlerin içinde değişiklik yapılabilmesi için yeniden programcıya dönülmesi gerekmektedir.

Burada eksik olan şey hazır etkileşim imkanlarının bir çatı altında toplanmasıdır. Tez süresince bu eksiğin nasıl doldurulabileceği uygulamalı olarak incelenmiştir. Hazırlanan bilgisayar programı kareografin önceden hazırlanmış sesli ve görüntülü etkileşimlerin parametrelerini dilediği gibi değiştirebilmesini, ve bu etkileşimleri belli bir mantık çerçevesinde bir araya getirebilmesini sağlamaktadır.

Tez bünyesinde önce teknoloji ve multimedyanın performans sanatlarında kullanımına bir göz atılmıştır. İkinci bölümde dünyada dans teknolojisiyle ilgilenen kişi, grup ve kuruluşlara değinilmiştir. Bunun ardından dans teknolojisinde kullanılan giriş ve çıkış donanımları incelenmiştir. Bölüm sonunda ise halihazırda kullanılmakta olan elektronik performans tasarım programlarının bazıları incelenmiştir.

Üçüncü bölümde, yapılan incelemelere göre performans sanatlarında kullanılmak üzere bir sistem önerisinde bulunulmuştur. Hazırlanan giriş ve çıkış modüllerinden ve bir kareografi programından oluşan sistemin teorik ve pratik alt yapısı bu bölümde anlatılmıştır. Tez bünyesinde önerilen sistemin büyük bir kısmı gerçekleştirilmeye çalışılmış ve tamamlanan kısımlarla ilgili ekran görüntülerine ve programın kullanımıyla ilgili bilgilere dokümanda yer verilmiştir.

2. TEKNOLOJİ VE MULTİMEDYANIN DANSTA KULLANIMI

2.1 Genel Bakış

İlerleyen teknoloji hayatımızın pek çok yerinde olduğu gibi sanat alanlarında da kendini hissettirmektedir. Plastik sanatlarda yerini çok daha önceden bulmuş olan teknoloji, performans sanatlarında da artık sık kullanılır hale gelmiştir. Önceleri sadece önceden kaydedilmiş ses ve görüntülerin kullanıldığı bu alanda sahne sanatçısının bu elemanlar üzerinde hiçbir etkisi olamamaktaydı. Sanatçının kaydedilmiş görüntülerle senkronize hareket etmesi beklenmekteydi. Fakat gelişen donanım ve yazılım teknolojisi yardımıyla artık etkileşimli ortamlar hazırlanabilmektedir. Böylece performans sanatçısı sahnedeki görsel ve işitsel elemanların karşısında edilgen pozisyonunu bırakıp, bu elemanları kontrol edebileceği karşılıklı etkileşebileceği yeni bir konuma kavuşmuştur. Bir nevi, sanatçının kendi bedeni dahil tüm sahne bir enstrümana dönüşmüştür. (Zaks-Zilberman, 1999)

Dansçının sahne elemanları üzerindeki kontrolünün artması çeşitli sebeplerden ötürü pek çok fayda ve estetik etki getirmektedir. Her şeyden önce bu durum dansçı ve ortam arasında çok daha kesin senkronizasyon sağlamaktadır aynı zamanda sanatçının performansa doğaçlamayla kendi yorumunu katması için esnek bir ortam sunmaktadır. Bunların dışında sanatçının sahnedeki farkındalığını arttırarak vücut hareketleri üzerinde kontrolünü kuvvetlendirmesine yardım etmekte ve sanatçıya performans esnasında ilham verebilmektedir. (Siegel ve Jacobsen, 1998)

Sanatçıyla sahne arasındaki etkileşimin gerçekleşmesi bu tezin bünyesinde dört ana problem başlığı altında incelenmektedir.

1. Sahne verisinin sayısal ortama aktarılması (giriş)
2. Sayısal ortamda bu bilgilerin işlenerek sahneye verilecek görsel/işitsel/fiziksel çıkışın üretilmesi (işlem)
3. Sahne üzerinde görsel/işitsel/fiziksel çıktının uygulanması (çıkış)
4. Sahneden alınan verinin nasıl işleneceğinin, çıktıların neler olacağının belirlenmesi, zamanlama ve tetikleme düzenlenmesi (kompozisyon)

Sanatçının sahne elemanlarıyla etkileşime girebilmesi için bir şekilde sisteme dansçıyla ilgili girdi bilgisi alınması gerekir. Girdi kavramını biraz daha açmak gerekirse, sahnede sanatçının oluşturabileceği her türlü parametre girdi olarak kabul edilebilir. Şu an kullanılabilen teknolojiyle sanatçının;

- konumuna

- sayısına
- silüetine
- sahnede kapladığı bölgenin boyutuna
- varlığı/yokluğuna
- kas ve iskelet hareketlerine
- beyin dalgalarına
- çıkardığı seslere

ait ham veriler girdi olarak alınabilmektedir. Bu verileri sahneden alıp işleyebilmek için çeşitli giyilebilir ve giyilemez donanımlar geliştirilmektedir. Eskiden çok ağır ve rahatsız olan sensör sistemleri kablosuz teknolojinin gelişmesiyle artık sahnede giyilebilecek derecede hafifliğe ve rahat kullanıma kavuşmuştur. Sensör sistemlerinin dışında video işleme tekniklerindeki gelişme sayesinde video kameralar da artık etkileşimli sistemlerin vazgeçilemez parçaları haline gelmiştir.

Bu veriler ham halde kullanılabildiği gibi daha önceki verilerle karşılaştırılarak çeşitli yorumlar çıkarılması da mümkündür. Örnek vermek gerekirse; sanatçının o anki konumunu önceki konumlarıyla karşılaştırarak sahnede ne yöne hareket ettiği yada sahnedeki hareketinin hızı saptanabilmektedir.

Dünya çapında dans teknolojisi alanında gerek sanatsal gerekse akademik pek çok çalışma yapılmaktadır. Aşağıdaki listede bu alanda çalışan önemli kişi ve kuruluşların listesi verilmektedir:

Kişiler:

- Johannes Birringer - Leslie Bishko
- Merce Cunningham
- Scott deLahunta - Kent De Spain
- William Forsythe
- Bill T. Jones
- Paul Kaiser - Michael Klien - Susan Kozel
- Richard Lord
- Wayne McGregor
- Lisa Naugle
- Sita Popat - Richard Povall

- David Rokeby - Nick Rothwell - Sarah Rubridge
- Thecla Schiphorst - Yacov Sharir - Stelarc - Scott Sutherland
- Jane Turner
- Elizabeth Streb
- Mark Coniglio
- Dawn Stoppiello

Grup ve Kuruluşlar:

- AlienNation Co
- Barriedale Opera House
- CASSIEL
- Halfangel
- Palindrome Intermedia Performance Group
- Satorimedia
- Troika Ranch
- SenseLabor

Enstitüler:

- ADaPT Association for Dance and Performance Telematics.
- Ohio State University OSU Dance.
- Arizona State University ASU Dance

2.2 Dans Verisinin Sahneden Alınması İçin Kullanılan Donanımlar

2.2.1 Sınıflandırma

Sahnenin durumunu sayısal ortama aktarip işleyebilmek için çok çeşitli donanımlar kullanılmaktadır. Kullanılan donanımlar sanatçıyla fiziksel etkileşimine göre üç ana kategoride gruplanabilir:

- İçten – içe sistemler: Bu sistemlerde sanatçının üzerine takılı sensörlerle sanatçının kendisine ait veriler toplanmaktadır. Özellikle bedenın küçük parçalarına ait veriyi almada kolaylık sağlamaları açısından kullanılmaktadırlar. Bu sistemler giyilebilir olduğu için kimi zaman hareketi kısıtladıklarından kimi zaman da estetik görünümü bozduklarından tercih edilmeyebilirler. Sanatçının kalp atışları, bazı kaslarının kasılması veya beyin dalgaları bu tip sistemler kullanılarak alınabilmektedir.

- İçten-dışa sistemler: Bu tip sistemlerde sanatçının üstüne sahneye ait verileri toplamak için bazı sensörler yerleştirilir. Bu yöntem genelde sanatçının sahneye göre konumunu öğrenmek için kullanılır, fakat çalışma menzilleri ve güvenilirlikleri sınırlı olduğu için genelde tercih edilmez.
- Dıştan-içe sistemler: Bu tip sistemler sanatçının üstünde bulunmayan ve sanatçıya ait verileri algılamaya çalışan sensörlerden oluşurlar. Kameralar, mikrofonlar ve yer sensörleri dıştan-içe sistemlerde kullanılan sensörlerdir. Sanatçının üstüne giymesi gerekmediği için sanatçıya sahnede serbest hareket imkanı sağlar, ama özellikle kamera kullanımında görüntüden anlamlı veri çıkarılması yüksek işlem gücü gerektirmektedir. (Mulder, 1994b)

2.2.2 Kamera

Video işleme tekniklerindeki ve veri aktarım hızındaki gelişmeler kameraları etkileşimli sistemlerin vazgeçilmez parçaları haline getirmiştir. Temelde kameranın yaptığı iş sahnenin belli bir açıdan görüntüsünü video işleme sistemine iletmesidir. Böylece sistem kameradan aldığı veriyi işleyerek sanatçıyı sahnede tespit eder ve bu veriyi kullanır. Kamera kullanılarak yukarıda sayılan konum, sayı, silüet, boyut, varlık gibi bilgileri elde etmek mümkündür. Bunun dışında kullanım kolaylığı ve maliyetinin düşüklüğü de kamerayı veri çeşitliliğiyle beraber etkileşimli performans sistemlerinin baş girdi donanımı haline getirmektedir.

Kamera kullanımının sağladığı avantajlar dışında getirdiği bazı zorluklardan da söz etmek mümkündür. Bunların başında kameradan gelen verinin işlenmesi gelmektedir. Kamera, yapısı itibarıyla sahnenin hedefteki bölgesinin tümünü sisteme göndermektedir. Bu da alınan her kare başına büyük miktarda verinin işlenmesi ve anlamlı bilgiler çıkarılması anlamına gelmektedir. Görüntü verisinin içinden sanatçıyı tespit etmek başlı başına bir problemdir .

Bunun dışında, kameradan bilgisayara aktarılan veri bir miktar gecikmeye uğramaktadır. Bu gecikme milisaniyeler seviyesinde olduğu için çoğu sistemde önemsenebilir ama yine de sistem tasarlanırken hesaba katılması gereken bir konudur. Kurulacak sistemin ihtiyaçlarına ve bütçeye göre uygun kameranın seçimi önem taşımaktadır. Aşağıda kullanılabilecek bazı kamera cinslerinin karşılaştırması yapılmıştır. (Tablo 2.1)

Cihaz	Avantajları	Dezavantajları	Yaklaşık Fiyat	Resim
Analog kamera	+Kablo uzunluk sınırı yok +Değişik lensler mevcut (Infrared dahil) +Monte etmesi kolay +Hızlı ve iyi kalitede görüntü	-Bilgisayara bağlamak için ekstra donanım gerekiyor	100-800\$	
Webcam	+Düşük maliyet +Bilgisayara bağlamak kolay	-Kısa kablo -Sabit lens -Düşük görüntü kalitesi	40-80\$	
DV kamera	+İyi görüntü kalitesi	-Yüksek veri yoğunluğu -Gecikme -Limitli kablo uzunluğu	1000-5000\$	

Tablo 2.1 Kamera karşılaştırması

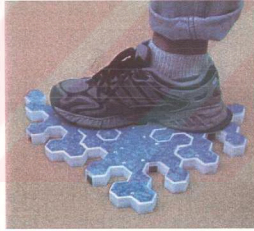
Kamera bir etkileşim aracı olarak sadece sanatta değil aynı zamanda eğlence teknolojisinde de yavaş yavaş yer edinmeye çalışan bir girdi aracıdır. Sony® firmasının Playstation© için piyasaya sürdüğü EyeToy© isimli kamera Playstation'a bağlanarak bir kontrol arabirimi olarak oyunlarda kullanılabilir. Eğlence sektörü dışında akademik alanda da kameranın etkileşim aracı olması üzerine çeşitli araştırmalar yapılmaktadır. Parsons School of Design'da yapılan bir çalışmada tüm vücut etkileşimi için kameranın kullanımında tatmin edici sonuçlar ortaya konulmuştur. (Warren, 2003)

Eğlence sektöründe kamera görüntüsünün bir başka yaygın kullanımı motion capture tekniklerinde görülmektedir. Motion capture, hareket verisinin gerçek hayattaki canlıların eklem konumlarının örneklendirilerek sayısal ortama aktarılması işlemine denmektedir. Bu işlem

için geliştirilen pek çok yöntemden biri de oyuncunun üzerine takılan ışık yayan vericilerin bir kamera sistemi tarafından sayısal ortama aktararak hareketin sayısallaştırılmasıdır. (Theobalt vd, 2004)

2.2.3 Sahne Sensörleri

Sanatçının sahnedeki konumunu saptamak için kullanılabilecek yöntemlerden bir diğeri tüm sahneye veya sahnenin gerekli görülen bir bölgesine sensörler yerleştirmektir. Bu yöntem sayesinde sahne belirli büyüklükte parçalara bölünmektedir ve sahne üzerinde bazı işaretlemeler yapılmaktadır. Sanatçı bu işaretlenmiş bölgelere girdiğinde ilgili sensör tetiklenerek sisteme sinyal göndermektedir. Yerdeki basınca göre tepki veren yer sensörleri üretilip kullanılmaktadır ancak sensörün karmaşıklığına göre maliyeti oldukça yüksek olabilmektedir. MIT Media Lab'ın "Magic Carpet" projesi (Paradiso vd, 1997), "LiteFoot" projesi (Griffith ve Fernstörn, 1998) ve "Z-Tiles" projesi (Richardson vd, 2003) performans ve enstelasyonlarda kullanılmak üzere üretilen yer sensörlerine örnek gösterilebilir. (Şekil 2.1)



Şekil 2.1 Yer sensörü

Basınçlı yer sensörleri dışında ultrason ve optik sensörler yardımıyla da dansçının sahnedeki yerini tespit etmek mümkündür. Örnek olarak İtalya'da Infomus Lab'ın geliştirdiği DanceWeb isimli ultrason sensörlerden oluşmuş düzenek enstelasyonlarda kullanılmaktadır. (InfoMus Lab, 2005)

Yine kamerada olduğu gibi yer sensörleri de sayısal eğlence alanında kullanılmaktadır, ancak maliyetin yüksekliği yüzünden görece olarak daha basit sistemler son kullanıcıya sunulmaktadır. Yer sensörlerinin yakın zamandaki kullanımına örnek olarak Playstation için üretilen Dance Dance Revolution© isimli dans simülasyonu için tasarlanan ve her biri

konsolun kontrolöründeki bir tuşa denk düşen sekiz aktif bölgesi olan yer sensörünü sayabiliriz. (Şekil 2.2) Bu aksesuarın kullanımına gelince, sekiz aktif bölgeden her biri oyundaki karakterin bir figürünü temsil etmektedir. Oyuncunun hedefi doğru zamanlarda doğru bölgeyi ayağıyla uyararak karakterin düzgün biçimde dans etmesini sağlayıp puan toplamaktadır. Böylece oyuncuyla karakter karşılıklı dans ederek aralarında bir etkileşim yaratılmaktadır. (Mohnsen, 2005)



Şekil 2.2 Dance Dance Revolution kontrolörleri

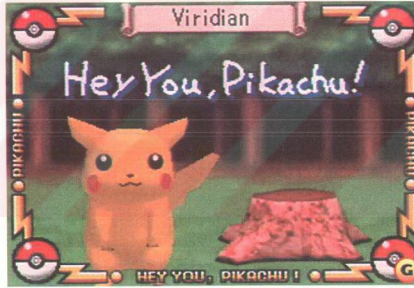
2.2.4 Mikrofon

Sahnedeki sanatçının ürettiği sesleri alabilmek için mikrofona ihtiyaç vardır. Mikrofon sayesinde sanatçının kendi bedeni veya çeşitli aletlerle çıkardığı darbe ve sesler algılanıp, bu veri üzerine kurgular üretilebilmektedir. Ancak sahnedeki sesin sadece sanatçıya ait olmayabileceği gerçeği mikrofon kullanımının bir dezavantajıdır. Sanatçının çıkardığı sesler izleyicilerin ve sahnenin kendi ses sisteminin ürettiği seslerle girişime uğrayarak yanlış algılamalara sebep olabilir. Bunun önüne geçmek için çeşitli frekans spektrumlarını süzecek filtrelerden yardım alınabilir. (Şekil 2.3)



Şekil 2.3 Standart bir mikrofon

Mikrofon da diğer örneklerde olduğu gibi yeni yeni kullanılmaya başlanan bir etkileşim aracıdır. Genel olarak görülen örnekler kullanıcının verdiği sözel emirlerin algılanması şeklinde olduğu için sesin bazı işlemlerden geçirilerek içinden kelimelerin ayıklanması gerekmektedir. Örnek olarak Nintendo® firmasının Nintendo64 konsolu için piyasaya sürdüğü Hey You Pikachu! isimli oyun verilebilir. (Şekil 2.4) Bir mikrofonla birlikte gelen oyunda oyundaki sanal hayvanla oyuncunun sözcüklere dayanan etkileşimi söz konusudur. (IGN.COM, 2005)

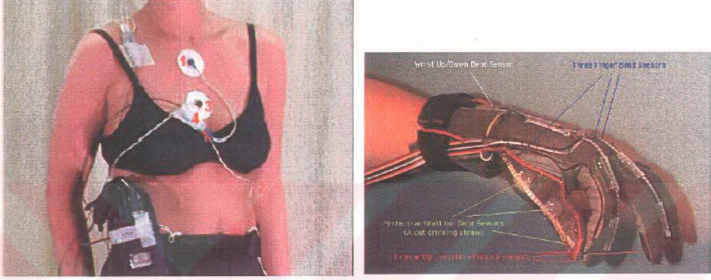


Şekil 2.4 Hey You Pikachu oyunundan bir ekran görüntüsü

2.2.5 Kas / İskelet Sensörleri

İçten-içe sistemlerde kullanılan kas ve iskelet sensörleri yardımıyla dansçının fiziksel durumuna ilişkin bilgi alınabilmektedir. Dansçının eklemelerine takılan sistemler yardımıyla her hangi bir eklemesinin açılış ve kapanışı açı olarak alınabildiği gibi göğse takılan elektrotlar yardımıyla kalp atışı ve çeşitli kaslar üzerine takılan elektrotlarla o kasların kasılma ve gevşeme bilgileri alınıp istenilen sistem içerisinde kullanılabilir. Giyilebilir sistem olduklarından önceleri kablolar yüzünden tercih sebebi olmasalar da artık kablosuz ve boyut

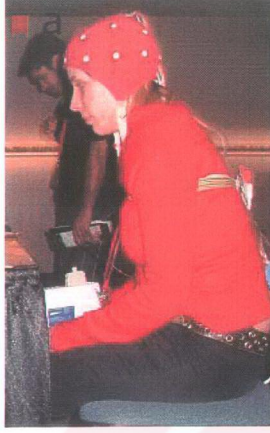
olarak oldukça küçük sensörlerin çıkması ve maliyetlerin de düşmesiyle kas/iskelet sensörlerini kullanılabilir giriş donanımları arasına sokmuştur. Troikaranch firmasının geliştirdiği MidiDancer isimli sensör sistemi kablosuz iskelet sensörlerinin ürettiği MIDI mesajlarıyla çalışmaktadır. (Troikaranch, 2005) Giyilebilir hareket sensörlerine örnek olarak içine basınç ve hareket sensörleri yerleştirilmiş eldiven ve ayakkabılar için de çeşitli çalışmalar yapılmaktadır. (Mulder, 1994a) (Paradiso, 2000) (Şekil 2.5)



Şekil 2.5 Kalp atışı sensörü ve sensörlü eldiven

2.2.6 Beyin Sensörleri

Kafatasına yerleştirilen elektrotlarla beyinden gelen EEG sinyallerini almak mümkündür. Alınan analog sinyaller çeviricilerle sayısal ortama aktarılıp orda işlenebilir. (Şekil 2.6) İçten-içe sistemlerden biri olan beyin sensörleri giyilebilir sistemlerdendir. Beyinden alınan sinyallerin kararsız olması ve çoğu zaman sistem için anlam çıkartmanın, dansçı içinse kontrol etmenin güç olması bu alanda kullanılmasını deneysel işlerle sınırlı bırakır. Beyin dalgalarının yorumlanmasıyla ilgili sanat dışında başka alanlarda da çalışma yapılmaktadır. Bu çalışmaların hedefi beyin dalgalarını kullanarak bir pilotun uçağı uçurması yada felçli bir hastanın tekerlekli sandalyesini yönlendirebilmesi gibi konulardır. Fakat bu çalışmalar henüz emekleme aşamasındadır.



Şekil 2.6 EEG sensörü

2.3 Verinin Sahneye Geri Verilmesi (çıkış)

2.3.1 Tanım

Sanatçı ile sahnenin etkileşim içinde olabilmesi için sahneden alınan verinin işlenip bu verilere dayanarak sahnenin bazı parametrelerinin değiştirilmesi gerekmektedir. Sahnenin değiştirilebilir öğelerine bakarsak üç ayrı kategoride gruplayabiliriz:

- Görsel öğeler: Işıklandırma ve sahneye yansıtılan görüntüleri içerir
- İşitsel öğeler: Müzik ve ses efektlerini içerir
- Fiziksel öğeler: Hareketli dekorları içerir

Sahneden alınan verilerin tetiklediği yada bu verilere dayanılarak oluşturulan yeni sahne öğesi durumların aktarılması için bazı çıkış aygıtları gerekmektedir. Bu aygıtlar sahnenin görsel ve işitsel yapısını etkileyebileceği gibi sahnedeki dekorları düzenleyerek sahnenin fiziki yapısının değişmesini de sağlayabilir. Aşağıda bu hedeflere ulaşılabilmesi için kullanılan bazı aygıtlar incelenmiştir.

2.3.2 Projektörle Yansıtma

Sahnede görsel olarak kullanılabilecek en etkili aygıt projektörlerdir. Bilgisayar tarafından üretilmiş her türlü görsel (slayt, video, animasyon, tipografi, vb) sahneye kolayca yansıtılabilir ve sahnede istenilen ortam oluşturulabilir. Klasik yöntemle sahne arkasına

yansıtma tekniği kullanılabildiği gibi bazı yenilikçi grupların işlerinde görüldüğü gibi dekorla birlikte farklı kullanımlar denenmektedir.

2.3.3 Ses Sistemi

Sahnede gerekli ortamı sağlayan en önemli etkenlerden bir diğeri de ses sistemidir. Sanatçı kendi ürettiği parametrelerle sahnedeki müziği etkileyebilir, ses efektlerini tetikleyebilir. Sanatçıya bu etkiler üzerinde tam kontrol vermek olasıdır, bu etkiler müziğin durdurulup çalınması, çalım hızı, çeşitli ekolayzır etkileri ve pan gibi bir çok özelliği kapsamaktadır. Salona dönecek gelişmiş sinema ses sistemlerinde her bir hoparlörden gelen sesi ayrı ayrı kontrol edilmesi de mümkündür. (Rowan vd, 2001)

2.3.4 Aydınlatma Sistemi

Klasik sistemde sahnedeki aydınlatma sisteminin kontrolü özel donanımlar yardımıyla fakat elle yapılmaktadır. Kontrolde insan faktörünün işin içine girmesi sahne tasarımıısının aydınlatma sistemi üzerindeki kontrolünü zayıflatmakta ve gösteri anında yapılacak hataları da birlikte getirmektedir. Sahnedeki ışık sisteminin kontrolünün bilgisayarlara bırakılması hem kareografin sahne üzerindeki kontrolünü arttıracak, hem de sahnedeki dansçı için ışıklandırmayı da etkileşimin bir parçası haline getirir. Aydınlatmayla ilgili sadece hangi spotun yanıp yanmayacağı değil hazırlanabilecek motorlu düzeneklerle aydınlatmanın sahne üzerinde yönelimi gibi çeşitli görsel efektlerin de kontrol edilmesi mümkündür.

2.3.5 Dekor Sistemi

Motorlar yardımıyla sahnedeki çeşitli dekor parçalarının hareket ettirilmesiyle ortaya çıkan dinamik dekorlar günümüz modern performanslarında kullanılmaktadır. Sahnedeki hemen her türlü dekor parçasını motorlar kullanarak hareket ettirmek mümkün olmakla beraber bu sistemi sayısal bir sisteme entegre etmek olasıdır.

2.4 Dans Teknolojisi'nde Kullanılan Yazılımlar

2.4.1 Tanım

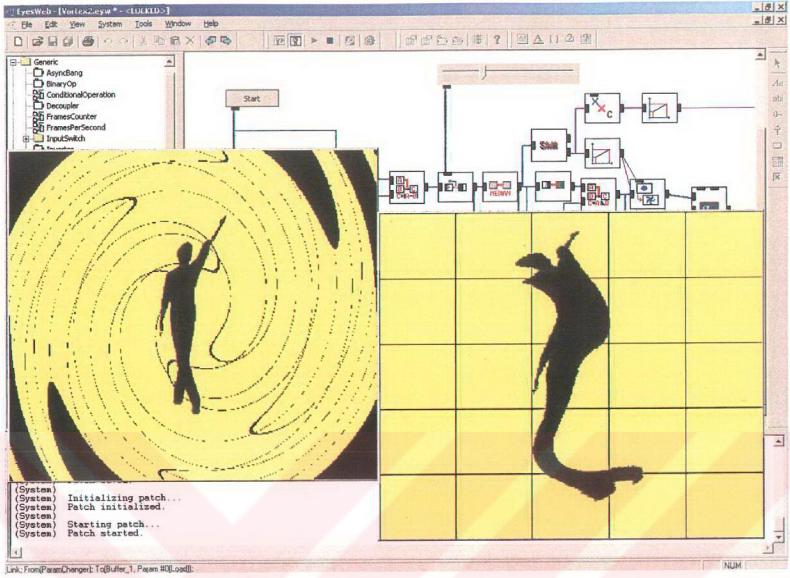
Sahnedan alınan dansçıya ait verilerin sayısal ortama alındıktan sonra işlenip bazı sonuçlar çıkartılması ve bu sonuçlara göre bazı çıktıların üretilmesi gerekmektedir. Bu ara işlemin kotarılması için çeşitli yazılımlar kullanılmaktadır. Bu yazılımların ortak mantığı sahneden alınan verinin işlenip çıkan sonuçlara göre sahneye verilecek medyanın üzerinde oynanmasına dayanmaktadır. Kullanılan yazılımlar giriş ve çıkış olarak kullandıkları medya ve donanıma

ve kurgulama için kullanıcıya sundukları arayüz mantığına göre birbirlerinden bariz şekilde ayrılmaktadırlar.

Dans teknolojiyle ilgilenen bir kısım sanatçı grubu kendi yazılımlarını üretirken büyük bir kısmı hazır paket programları üzerlerine kendi eklenti yazılımlarını(plug-in) yazarak kullanmaktadırlar. Dans teknoloji performanslarında kullanılan paket yazılımların hepsi dans için özel tasarlanmamıştır, özellikle sayısal müzik için tasarlanmış MIDI yazılımları da geniş çapta kullanılmaktadır. Yine kullanılan yazılımların pek çoğu etkileşimin sadece sahnedeki ses ve müzikle olmasını sağlamakta ve herhangi bir görsel çıktı üretmemektedir. Koreografa geniş anlamda bir etkileşim tasarımı imkanı veren yazılımlar ise çok az sayıdadır.

2.4.2 EyesWeb

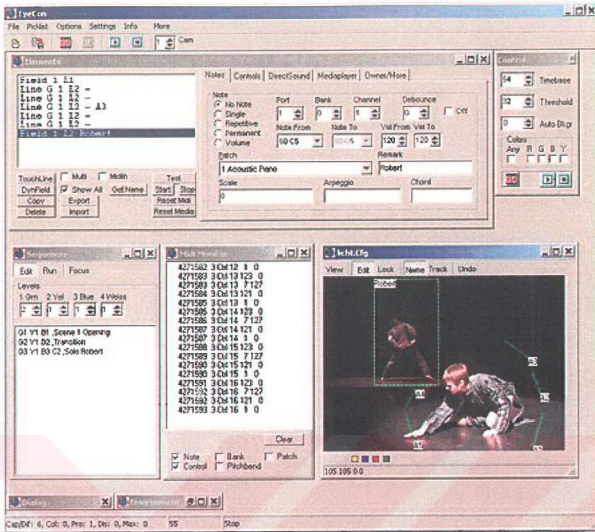
İtalya'da EU IST Mega(Multisensory Expressive Gesture Applications) Project dahilinde geliştirilen EyesWeb yazılımı temel olarak kameradan gelen görüntüyü alıp işledikten sonra çeşitli ortamlarda çıktı veren ve etkileşimin kurgulanabildiği bir yazılımdır. (Şekil 2.7) Kamera başta olmak üzere pek çok ortamdan veri alabilmekte ve pek çok ortama veri gönderebilmektedir. Kendi içinde özelleştiği konu kamera yardımıyla aldığı görüntüden sahnedeki dansçıya ait ifadeleri algılayabilmesidir. Program bünyesinde kullanılmak üzere görüntü ve jest işleme teknikleri üzerine pek çok kütüphane içermektedir. Bunun dışında multiplexer yardımıyla birleştirilen birden fazla kamera görüntüsünü ayırıp kendi için ayrı ayrı işleyebilmektedir. (Camurri vd, 2000)



Şekil 2.7 EyesWeb programının arayüzü

2.4.3 EyeCon

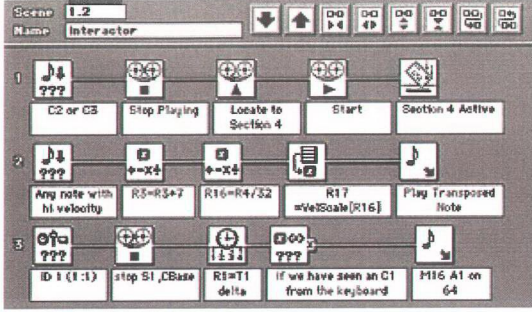
EyeCon, Frieder Weiss tarafından Alman Palindrome firmasının çatısı altında geliştirilen bir yazılımdır. (Şekil 2.8) Temel girdi olarak canlı kamera görüntüsünü kullanır ve işlediği görüntülerden elde edilen veriler sahneye aktarılabilecek medyanın kontrolü ve tetiklenmesinde kullanılır. EyeCon sahnedeki müzik, ses, fotoğraf, film ve ışıklandırmayı kontrol edebilecek yetenektir. Kameranın görüş açısında yaratılan sanal nesnelerin sahnedeki nesnelerle etkileşimini tasarlamak ve uygulamak için geliştirilmiş bir yazılımdır. (Dowling vd, 2004)



Şekil 2.8 EyeCon programının arayüzü

2.4.4 Interactor LPT

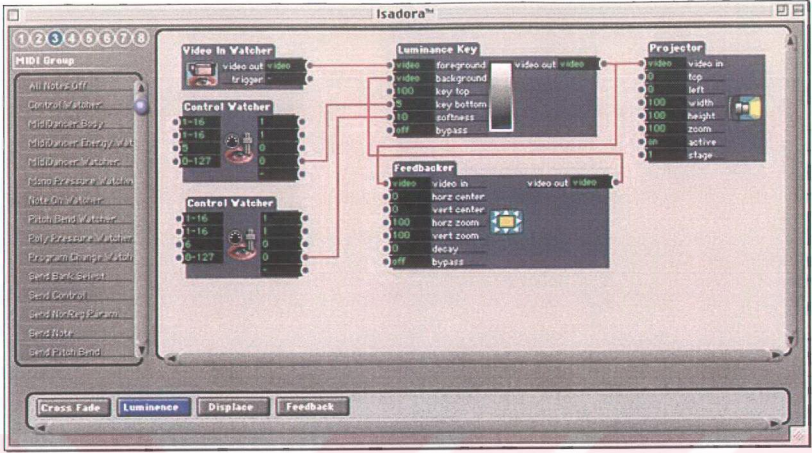
Interactor LPT, Troikaranch grubunun hem geliştirdiği hem de performanslarında kullandığı temel etkileşim tasarım yazılımıdır. (Şekil 2.9) Temel etkileşim arayüzü olarak yine Troikaranch'in geliştirdiği MidiDancer donanımı, klavye ve diğer sensörlerden gelen MIDI mesajlarını kullanır ve koreografa bu mesajların nasıl yorumlanabileceğini tasarlayabileceği bir arayüz sunar. (Troikaranch, 2005)



Şekil 2.9 Interactor LPT programının arayüzü

2.4.5 Isadora

Isadora, Troikaranch grubunun eğitimlerde kullandığı Interactor LPT motorunun üzerine kurulu fakat kullanımı daha kolay olan bir programdır. (Şekil 2.10) Genel amacı tam olarak özelleştirilebilir bir performans yaratmaya izin verecek bir geliştirme ortamını birleştirilebilir görsel bloklar kullanarak kullanıcıya sunmaktır. Actor adı verilen bu görsel bloklar basit veya karmaşık işlemleri gerçekleştiren fonksiyonlardır ve kullanıcı tarafından parametreleri değiştirilip diğer actor'larla birleştirilebilir. Pek çok giriş ve çıkış ortamının desteklendiği Isadora'nın önemli bir artısı kullanıcıların C/C++ ortamında kendi fonksiyonlarını geliştirebilecekleri bir SDK sunmasıdır, bu özellik sistemin geliştirilebilir ve sınırsız özelleştirilebilir olmasını sağlamaktadır. (Troikatronix, 2005)



Şekil 2.10 Isadora programının arayüzü

2.4.6 E-Merge

E-Merge projesi 1999'da Avrupa Komisyonu'nun Culture 2000 kültür programı sponsorluğunda koreograf Jane Turner ve besteci Daniel Biro liderliğinde başlamıştır. Projede sahne performansında teknolojiyle etkileşimin bir dil haline dönüştürülmesi hedeflenmiştir. Proje için geliştirilen yazılımda sahneden alınan ses ve MIDI sinyallerinden yola çıkılarak hazırlanan bu dilin yorumlanması gerçekleştirilmektedir. (Unsafe & Sound, 2005)

2.4.7 BigEye

Steim isimli elektronik performans araştırma ve geliştirme grubunun laboratuvarı tarafından geliştirilen BigEye programının temel çalışma ilkesi kamera yardımıyla aldığı video görüntülerini işleyip MIDI sinyalleri üretmesidir. Böylece ortamdaki müziği yönetmeye imkan tanımaktadır. Videodan alınan görüntüde sayısı 16'ya varan nesneleri rengi, büyüklüğü ve parlaklığına göre aynı anda takip edebilmektedir. Programda çeşitli sıcak bölgeler tanımlanarak bu bölgelerde nesnelerin durumuna göre müzikte çeşitli aksiyonlar tanımlanabilmektedir. MacOS9 işletim sisteminde çalışan program Quicktime destekli herhangi bir video kamera ile çalışabilmektedir. (Steim, 2005)

2.4.8 Cyclops

Mühendis ve sanatçı Eric Singer tarafından geliştirilen Cyclops, Max için yazılmış bir canlı

video takip ve işleme nesnesidir. Sahnedeki nesneleri video kamera yardımı ile alınmış görüntüsünü siyah-beyaz ve renkli analiz edebilmekte ve sonuç olarak elde edilen veriyi Max'te kontrol MIDI'si, ses ve görüntü üretmek için kullanmaktadır. (Eric Singer, 2005)



3. ÖNERİLEN SİSTEM

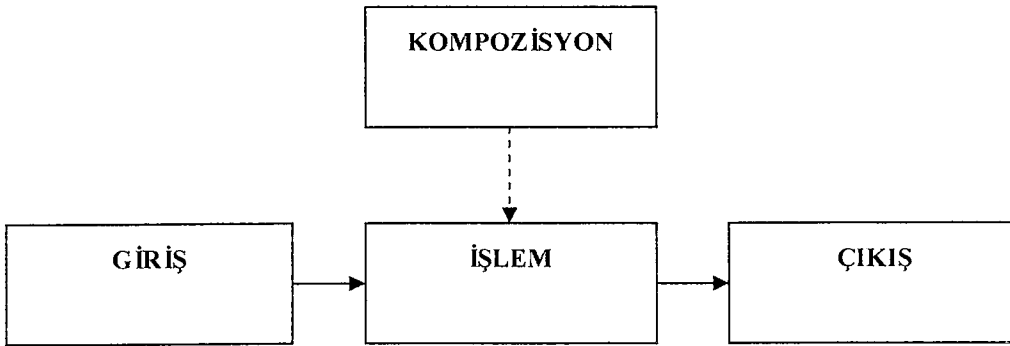
3.1 Özellikler

Önceki bölümde incelenen donanım, program ve kullanılan teknolojilerden elde edilen görüşlere dayanılarak ideal bir etkileşim sistemi tasarlanabileceği düşünülmüştür. Önerilen bu sistem sahnedeki etkileşimin kurgulanması ve gerçekleşmesinden sorumludur. İncelenen yazılımlar genelde kendi içlerindeki modüllerin çerçevesinde kullanılmakta, kullanılabilecek giriş ve çıkış donanımı ve üretilcek medya ve etkileşim bakımından kısıtlı olanaklar ortaya koymaktadırlar. Bundan yola çıkarak önerilecek sistemin sabit ve sınırlı bir sistem olarak değil de dinamik ve genişletilebilir, sistem dışından eklenmek istenen donanım ve yazılımları kendine entegre edebilecek bir çerçeve şeklinde tasarlanması düşünülmüştür.

Etkileşim söz konusu olduğunda üç ana seviyeden söz etmek gerekmektedir:

- Giriş: Sisteme verilerin alınması evresi
- İşlem: Girişten alınan verilerin yorumlanıp çeşitli kararların alınması ve tasarıma uygun sonuçların üretilmesi
- Çıkış: İşlem aşamasında üretilen sonuçların yazılım veya donanıma aktarılması

Bu üç seviye etkileşimin yaşandığı anda sistemde olan olayları temsil etmektedir. Ancak etkileşim yaşanmadan önce gerçekleşen ve girişten alınan verilerin nasıl işleneceği ve çıkışa nasıl aktarılacağına tasarlandığı bir süreç daha vardır, kompozisyon. Kompozisyon sürecinde etkileşim sisteminin tam anlamıyla tasarımı yapılmakta. Girişten alınan verilerin yorumlanmasıyla oluşan durumlar karşısında hangi kararların alınacağı, bu verilerin sistemdeki hangi parametreleri etkileyeceği kompozisyon sürecinde tasarlanır. (Şekil 3.1)



Şekil 3.1 Etkileşimin genel yapısı

Önerilen sistem iki parça halinde tasarlanmıştır. Bu parçalar birbirlerinden ayrı dağınık halde duran modüllerden ve modülleri bir araya getirip onlara işlevlerinin yüklenebileceğinin, giriş verisinin nasıl yorumlanacağı ve çıkış verisinin nasıl üretileceğinin belirlendiği kompozisyon arayüzüdür.

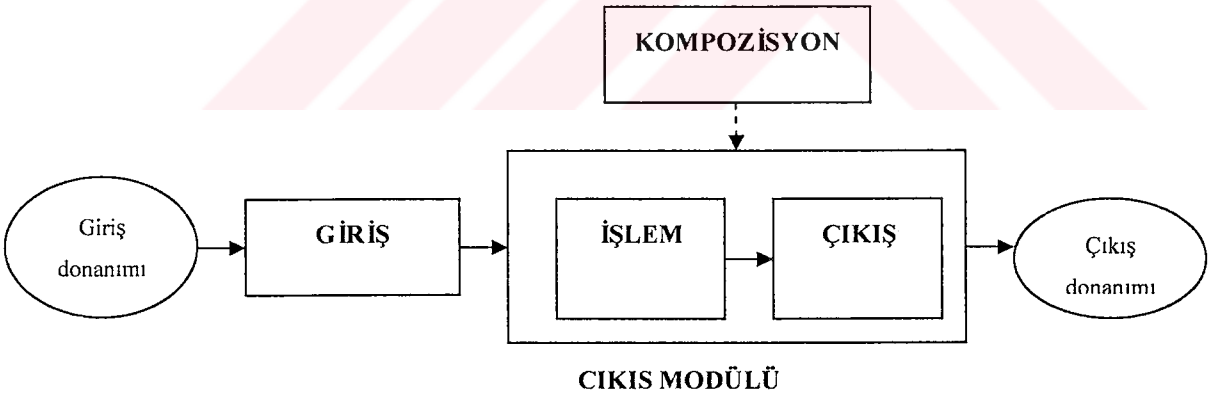
Yapılan araştırmalar sonucu önerilen bu sistemin olmazsa olmaz bazı özellikleri barındırması gerektiği ortaya çıktı. Bunları kısaca listelemek gerekirse:

- **Birden çok ortamdan girdisi olmalı:** Sahneden sadece tek bir aygıt kullanılarak alınan veriler yeterli beslemeyi sağlamaktan çok uzak olacaktır. Alınan verinin çeşitliliğinin artması dansçının sahne üzerindeki elemanları kontrol etmesi için çok daha fazla sayıda seçenek verilmesi anlamına gelecektir.
- **Birden çok ortama çıkışı olmalı:** Yapılan araştırmaların gösterdiği üzere sahnede pek çok çeşit çıkış ortamı kullanılmaktadır. Koreografa mümkün olduğu kadar çok çıkış vermek sahne üzerindeki kontrolü arttıracak gibi yaratıcılığı da tetikleyecektir.
- **Etkileşim seçenekleri düzenlenebilmeli:** Kareografin alınan bilginin nasıl işleneceği, sahneye aktarılacak medyanın seçimi ve düzenlenmesi ve her türlü diğer özellik üzerinde kontrolü olması gerekmektedir.
- **Genişlemeye açık olmalı:** Oluşturulacak olan sistem gelişen teknoloji ve yeniliklere açık olmalıdır. Sistem yeni giriş ve çıkış donanımlarının bünyesine entegre edilebilmesine izin verebilmeli, bunun yanında hazırda olan etkileşim seçeneklerinin çoğaltılması da mümkün olmalıdır.
- **Zamanlama sistemi olmalı:** Performans sanatları gerçek zamanlı icra edildiğinden olaylar zamana bağlı olarak planlanabilmeli ve etkileşimlerin başlama ve bitiş zamanı ayarlanabilmelidir.
- **Tetikleme sistemi olmalı:** Dansçının hareketleri etkileşimin normal seyri içinde çeşitli anlamlara göre yorumlanabildiği gibi, etkileşimler arası geçişlerde tetik olarak da kullanılabilmelidir.
- **İş yükü dağıtılabilmeli:** Hazırlanan sistemde gerek performans sebepleri gerekse donanımsal bazı kısıtlar yüzünden sistemin birden fazla bilgisayar üzerine dağıtılabilmesi gerekebilir.
- **Kullanıcı dostu olmalı:** Piyasadaki pek çok programın getirdiği karmaşıklıktan uzak ve sade olmalı. Kullanan kişinin üstün sayısal bilgiye gerek duymadan sonuç alabilmesi sağlanmalıdır. Basit arayüzler kullanarak karmaşık sonuçlara ulaşılabilmeli.

Yukarıda sayılan esaslara göre hazırlanan bir sistem özellikle esnek ve ölçeklenebilir olmasıyla hem geniş bir kitlenin kullanımına olanak vermesi açısından hem de geliştiricilerin üzerinde modifikasyon yapabilmesi açısından tercih sebebi olacaktır. Bu bölümün devamında önerilen sistemin özelliklerinin gerçekleşmesine ilişkin önermelerde bulunulacak ve bu tez bünyesinde gerçekleştirilen kısımları ayrıntılı bir biçimde anlatılacaktır.

3.2 Yapı

Sistemin yapısı etkileşim sisteminin tanımlanmasında olduğu gibi dört ana bölümden oluşacaktır. Bunlar; giriş, işlem, çıkış, ve kompozisyon bölümleridir. Bu bölümlerin birbirinden bağımsız çalışması sistemin modülerlik ve genişletilebilirlik özelliğini sağlaması açısından gerekmektedir. Bu yüzden bölümler, birbirleri arasında veri geçişine olanak sağlayan modüller şeklinde tasarlanmıştır. Yani her bölüm, sistem dışından yada bir önceki modülden aldığı girdilere göre kendi içinde işini gördükten sonra bir sonraki modüle yada çıkışa elde ettiği sonuçları iletmektedir, bu da bölümlerin modüler olmasını sağlamaktadır. Böylece bu modüller benzer girdi ve çıktılara sahip diğer modüllerle değiştirilebilecektir. Aşağıdaki şekilde, hazırlanan modüler yapının kabaca temsili görülmektedir. (Şekil 3.2)



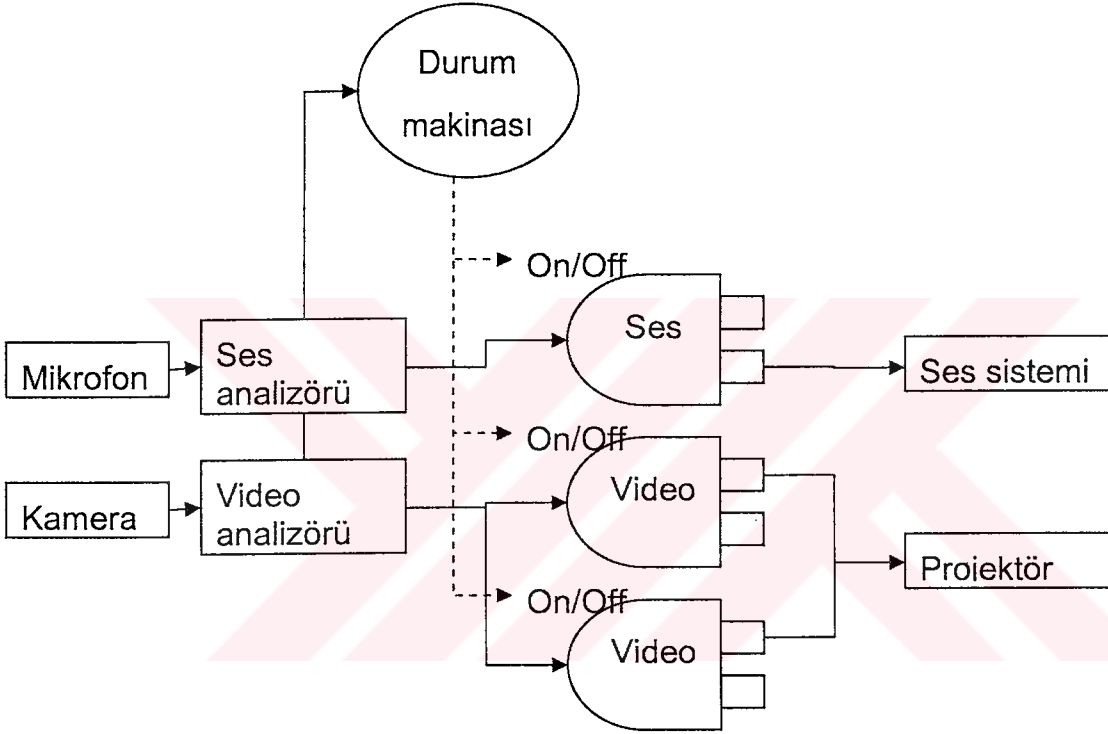
Şekil 3.2 Sistemin genel yapısı

Modüler yapının sağlanabilmesi için sistemin çeşitli modülleri tek bir çatı altında toplayıp yönetebilecek bir iskelet şeklinde düşünülmesi gerekmektedir. Bu iskelet yardımıyla koreograf kompozisyon sürecinde gerekli duyduğu modülleri bir araya getirip istediği etkileşim ortamını elde edebilecektir.

3.3 Sistemin Çalışması

3.3.1 Genel Yapı

Sistemin yapısına daha ayrıntılı bakacak olursak modüler yapının getirdiği faydalar açıkça görülmektedir. Bu sayede sistemin tamamen genişletilebilir ve özelleştirilebilir olmasının yanı sıra hazırlanan modüllerin sisteme eklenilip çıkartılması ile gerçek zamanda etkileşim sırasında kullanılması kolaylaştırılmıştır. Aşağıdaki şekilde sistemin çalışma esnasındaki veri akışı resmedilmiştir. (Şekil 3.3)



Şekil 3.3 Sistemin modüler yapısı

Kontrol modülü: Çalışma yapısı kompozisyon sırasında hazırlanan bu modül giriş işlemcilerinden gelen bilgiyi yorumlayarak ve zamanı izleyerek hangi çıkış modülünün çalışacağına karar vermek için kullanılır.

Giriş modülü: Sorumlu olduğu giriş donanımından gelen veriyi işler ve elde ettiği anlamlı bilgileri etkileşim modülleri ve durum makinesine aktarır.

Çıkış modülü: Bir veya daha fazla giriş işlemcisinden aldığı veriyi kendi yöntemleriyle işleyerek çıkış donanımına vereceği medya için parametre olarak kullanır. Çıkış donanımlarına gönderilecek her türlü animasyon, ses, efekt ve diğer etkiler çıkış modülleri tarafından üretilir. Belirli bir anda hangi çıkış modülünün çalışacağına durum makinesi karar

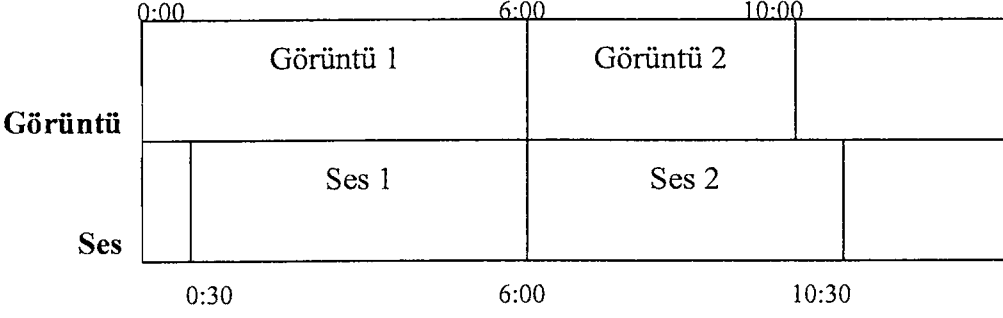
verir.

3.3.2 Kontrol Modülü

Kontrol modülü sahneden gelen veriyi ve zamanı sürekli izleyerek ortamın tasarımı sırasında kareografin belirlediği esaslara dayanarak hangi çıkış modülünün çalışacağına karar verir. Devamlı suretle sahneden gelen veriyi izleyen bu modül çalışma prensibi olarak iki ana kavramı iç içe barındırır; zaman ve durum makineleri. Bu aşamada daha fazla ilerlemeden aşağıdaki basit örnek senaryoyu incelemek faydalı olacaktır:

Dansçının gösterisi başlar. İlk başta dansçının sahnedeki yatay konumuna bağlı olarak projektörle hareketli çizgiler yansıtılmaktadır. Dansçı sağa gittiğinde çizgilerin sayısı artmakta, sola gittiğinde azalmaktadır. Sahnenin ortasında bir kutu durmaktadır. Dansçı kutunun önünden her geçtiğinde çizgilerin rengi değişmektedir. Sahnenin en solunda ve en sağda birer direk durmaktadır. Dansçı soldaki direğin yanına gittiği zaman bu sefer çizgilerin sahnede döndüğünü gösteren başka bir animasyon başlamaktadır. Dansçının ortadaki kutunun önünden geçmesi yine çizgilerin renklerini değiştirmektedir. Dansçı sağdaki direğin yanına geldiğinde ise başlangıç animasyon ve etkileşimine geri dönlmektedir. Performans başladıktan 30 saniye sonra müzik devreye girer. Dansçı 6 dakika bu etkileşimle dans ettikten sonra sistem başka bir sayfaya geçer. Müzik değişir. Bu sefer projektörle zıplayan beyaz daireler yansıtılmaktadır. Dansçı yerinden her zıpladığında yansıtılan görüntüye bir daire daha eklenir her daire eklenişinde bir ses efekti duyulur. 4 dakika daha devam eden dansçının performansı toplam 10 dakikada sona erer, müzik 30 saniye daha çalmaya devam eder.

Örnekten de görülebileceği gibi performans sanatları gerçek zamanlı icra edildiği için zamanla ilişkisiz olması düşünülemez. Koreograf kendisine sağlanan tasarım arayüzünü kullanarak hangi etkileşim sürecinin ne kadar süreceğini ve yarattığı etkileşim süreçlerinin sıralarını belirtebilmelidir. Örnekte görüldüğü üzere görüntü ve müzik iki ayrı safhada planlanmıştır. Şekil 3.4'te de görüldüğü gibi yansıtılan görüntünün başlamasından 30 saniye sonra başlayan müzik 5:30 dakika sonra bitmiştir, ve görüntü de toplam 6 dakika boyunca sahnede kalmıştır. Bir sonraki evrede ise 4 dakika görüntü sürmüş, bunun yanında arkada çalan müzik 4:30 dakika boyunca çalmaya devam etmiştir.



Şekil 3.4 Örnek senaryonun zamanlama yapısı

Böylece kontrol modülünün yapması gereken ana görev ortaya çıkmış olmaktadır. Kontrol modülünün ana görevi zamana bağlı olarak etkileşimlerin başlama ve bitiş zamanlarını kontrol etmektir. İçinde belli sayıda zamanlayıcı bulunduran kontrol modülü kareografin tasarım programında hazırladığı kompozisyondaki ardışıklık ilkesine dayanarak belli bir anda hangi modülün çalışması gerektiğine karar verir.

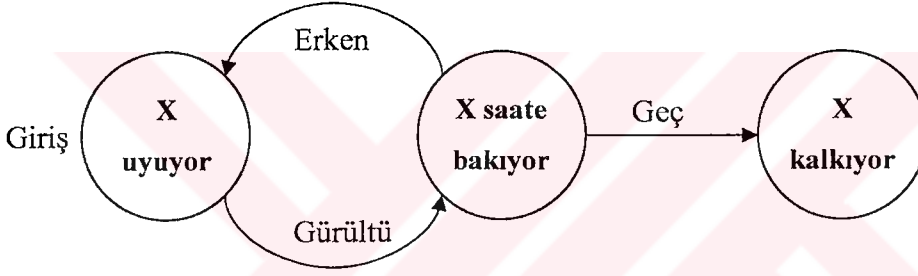
Zamanlayıcı kontrol modülünün çok önemli bir ögesidir ancak etkileşimin gerçekleşmesi için yeterli desteği vermekten uzaktır. Zamanlayıcının yalnız başına kullanılmasıyla sadece klasik gösterilerin yapılması olası olmaktadır, çünkü zamanlayıcının sunduğu lineer yapı etkileşim ortaya çıkarmak için kullanılabilecek bir yapı değildir. Burada gerekli olan şey etkileşimin gerçekleşmesi için daha etkin kullanılabilecek lineer olmayan bir mantıksal yapıdır. Yukarıda anlatılan örnek senaryoya tekrar dönersek lineer zaman dilimlerinin içinde etkileşimin gerçekleştiği fakat lineer olmayan daha farklı bir yapının olduğunu görmekteyiz. Sürekli çeşitli animasyonlar oynamasına rağmen bazı koşullar animasyonların içinde değişiklikler olmasına, bazı koşullarsa tamamen başka animasyonlara geçişe sebep olmaktadır. Bu olaylar zincirini basit olarak bazı koşullara göre bir durumdan başka bir duruma geçiş yapılan bir yapı olarak tanımlarsak aslında bu çeşit bir etkileşimin bir sonlu durum makinesine benzetebileceğimizi görürüz.

Sonlu durum makineleri belirli koşulların gerçekleşmesine göre durum değiştiren sistemlerdir. Gerçek hayattaki pek çok sistem sonlu durum makinesi şeklinde modellenebilir. Sonlu durum makinesindeki ana iki eleman durumlar ve geçişlerdir. Sonlu durum makinelerinin önemli bir özelliği şekil yanında tablo şeklinde de temsil edilebilmeleridir, bu sayede sayısal sistemlerde kullanılabilmektedirler. Sistem başlatıldığında hangi durumdan başlanacağı belirtilmelidir. Çok basit bir örnek vermek gerekirse şöyle bir senaryoyu sonlu durum makinesine

çevirebiliriz:

X kişisi uyumaktadır ve sabah 9:00'da uyanması gerekmektedir. Ancak X kişinin evi çok işlek bir caddedir ve gürültü yüzünden sık sık uyanmaktadır. X her uyandığında saate bakmakta eğer saat 9:00 olmamışsa uymasına devam etmekte, aksi taktirde yataktan kalkmaktadır.

Böyle bir sistemi durum makinesi olarak modellemek için durumları ve geçiş koşullarını ortaya çıkarmak gerekmektedir. Senaryoya bakarsak X'in üç farklı durumunun olduğu görülmektedir: Uyuyor, saate bakıyor ve uyanıyor. Bununla birlikte üç farklı geçiş koşulu mevcuttur: Gürültü, saatin erken olması, saatin geç olması. Bu sistemi sonlu durum makinesi olarak Şekil 3.5'deki gibi modelleyebilir ve Tablo 3.1'deki gibi tablo haline getirilebilir.



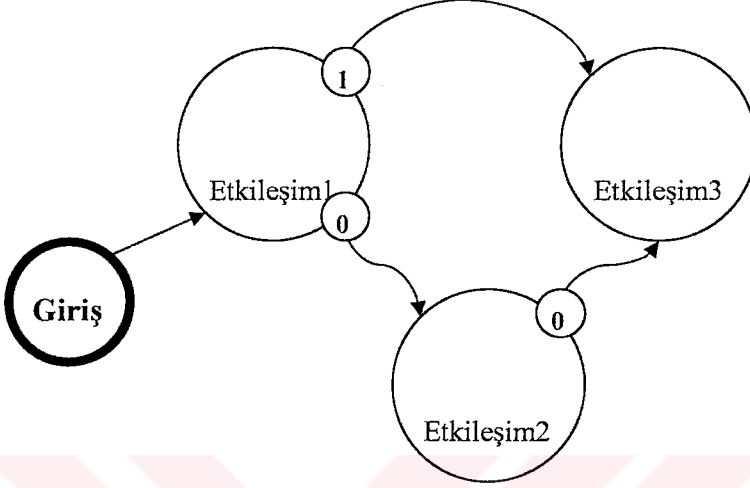
Şekil 3.5 Sonlu durum makinesi örneği

	Gürültü	Erken	Geç
X uyuyor	X saate bakıyor	-	-
X saate bakıyor	-	X uyuyor	X kalkıyor
X kalkıyor	-	-	-

Tablo 3.1 Sonlu durum makinesinin tablo gösterimi

Sistemde durum makinesi kullanımının en uygun çözüm olduğu görülmektedir. Durum makinesinde her bir etkileşim bir durumu temsil etmekte ve bazı koşulların sağlanması halinde durumlar arasında geçişler yaşanmaktadır. Şekil 3.6'da sistemde oluşturulabilecek örnek bir durum makinesi modeli görülmektedir. Bu modele göre sistemde üç adet etkileşim yer almaktadır. Etkileşim1 ile sisteme giriş yapılmakta, 0 veya 1 koşulu gelmediği sürece bu durumda kalınmaktadır. 0 koşulu gerçekleştiğinde Etkileşim2'ye 1 koşulu gerçekleştiğinde

Etkileşim3'e geçilmektedir. Etkileşim2'deyken bu durumun 0 numaralı geçiş koşulu gerçekleştiğinde bu sefer Etkileşim3'e geçilmektedir. Etkileşim3 için tanımlanmış hiçbir geçiş koşulu olmadığı için sistem bu duruma geldiğinde en sona kadar başka bir etkileşime geçemez. (Şekil 3.6)



Şekil 3.6 Sistemde durum makinesinin kullanımı

Kontrol modülünde durum makinesi veri yapısı olarak iki boyutlu bir dizide tutulmaktadır. Şekil3.7'de görüldüğü üzere bu dizide sütunlar durumları yani etkileşimleri, satırlar geçiş koşullarını, değerlerse belli durumun belli koşulu gerçekleştiği zaman hangi duruma yani etkileşime geçileceğini ifade etmektedir. Böyle bir yazım biçimi iki durum arasında birden fazla geçiş koşulu ile geçişe olanak sağlamaktadır ki bu sistemin esnekliğini daha da arttıran bir özelliktir.

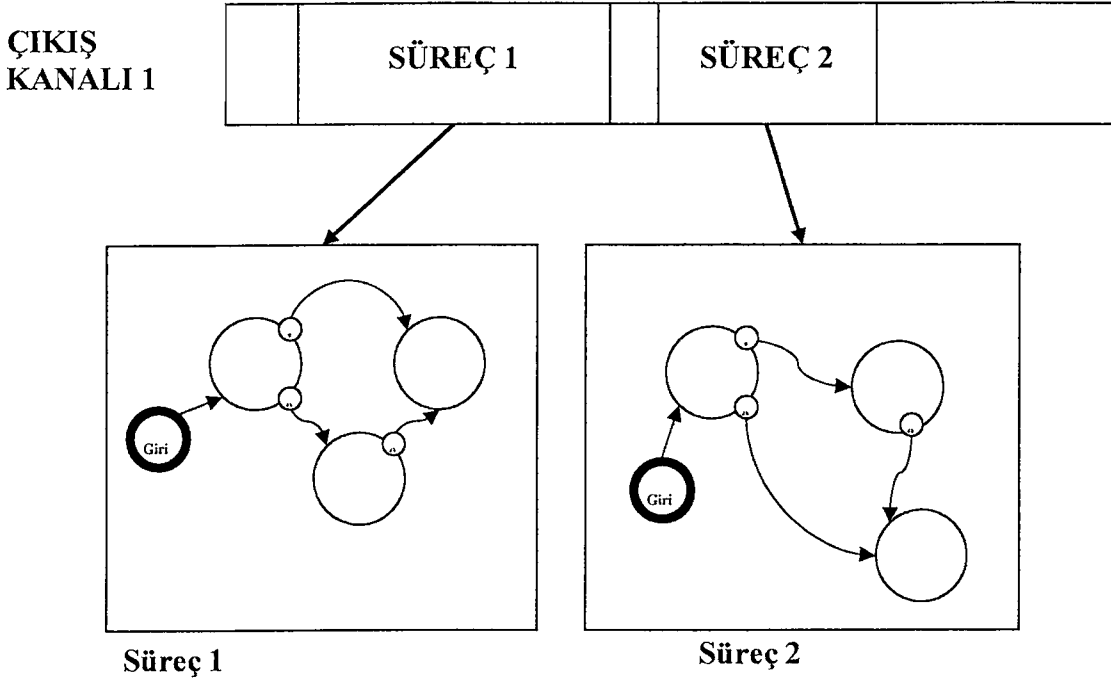
		Durumlar				
		0	1	2	3	4
Koşu lar	0	4	0	1	4	2
	1	3	-1	3	-1	-1
	2	2	-1	1	-1	-1
	3	-1	-1	-1	-1	-1
	4	-1	-1	-1	-1	-1

Geçiş Durumları

Şekil 3.7 Durum makinesinin sabit matris şeklinde gösterimi

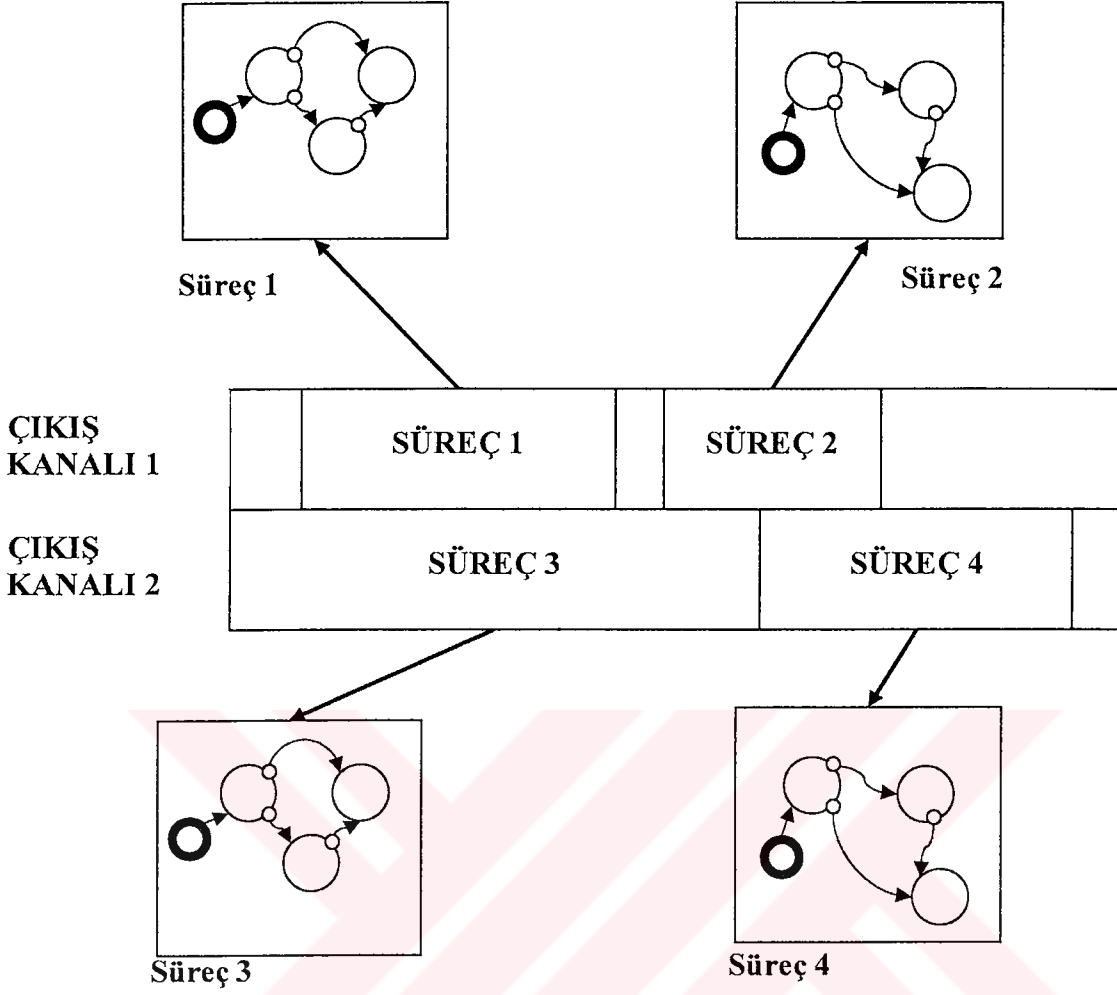
Şekil 3.7'deki örnek matrisi ele alırsak 0 durumundayken bu durumun 2 no'lu koşulu sağlandığında 1 etkileşimine geçilecektir. Yine 1 durumu devam ederken 2 koşulu sağlandığında 3 durumuna geçilmektedir.

Durum makinesi mantığını bir şekilde zamanlamayla birleştirmek gerekmektedir. Aksi takdirde bir etkileşim başladığında sonsuza kadar durum makinesinin içinde dönmek mümkündür. Sistem bu yüzden zamanlayıcıları kullanmaktadır. Zamanlayıcılar işin içine girdiğinde birbirinden farklı durum makineleri kullanıp bunları birbirine ardışık biçimde bağlamak mümkün olmaktadır. Böylece bir durum makinesinin işleme zamanı bittiğinde o an hangi durumda olursa olsun işini durdurur ve eğer arada bir zaman boşluğu yoksa yerini bir sonraki durum makinesinin giriş etkileşimine bırakır. (Şekil 3.8) Zaman boşluğu olması durumundaysa bir sonraki durum makinesinin vakti gelene kadar sistem boşa çalışır.



Şekil 3.8 Durum makinesinin süreçlerle birlikte kullanımı

Ancak tek bir durum makinesi belli bir andaki etkileşimi tarif etmek için yeterli olmayacaktır. Aynı anda sadece tek bir durum makinesinin kullanılması durumunda örneğin görüntü için bir koşulun sağlanıp da diğer etkileşime geçtiğinde ortamdaki bütün medyaların aynı anda durum değiştirmesi gerekecektir ve bu istenmeyen bir etkidir. Bunun önüne geçebilmek için ortamda birbirinden bağımsız her çıkış için ayrı durum makinesi kullanılması ve zamanlamalarının da ayrı ayrı yapılması gerekmektedir. Yeni her medya kendi kanalı boyunca işlevini yerine getirecek, böylece her biri ayrı ayrı planlanabilecektir. Tabi ki bunun için çok daha karmaşık veri yapıları kullanılmak zorundadır ama sonuç olarak çok daha serbest bir yapı elde edilecektir. Şekil 3.9’da birden fazla zaman çizgisinin içinde ardışık birden fazla durum makinesinin nasıl tutulduğu görünmektedir.



Şekil 3.9 Birden fazla çıkış kanalının kullanımı

Sistemin etkileşim sırasında çalışacak olan kısmı Processing ortamında Java dilinde yazılmıştır. Processing ortamı kamera, mikrofön gibi giriş ortamlarının, projektör, ses sistemi gibi çıkış ortamlarının ve seri port, ağ gibi giriş/çıkış ortamlarının çok kolay kullanılabileceği kütüphaneleri ve desteği içerdiği için tercih edilmiştir. Bunun yanında Java gibi nesneye yönelik bir dilin kullanılması modüllerin hazırlanması ve birbiriyle etkileşime girmesine imkan tanımıştır. Processing'in sağladığı rahat donanım arayüzü hem uygulamanın tez bünyesinde geliştirilmesi için hem de ileride başkaları tarafından eklenecek modüllerin geliştirilmesinde sağlayacağı rahatlıklar açısından fayda sağlamaktadır. (MIT Media Lab, 2005)

3.3.3 Giriş Modülü

Giriş modülleri giriş donanımından gelen verileri çeşitli filtrelerden geçirip yorumlayıp anlamlı bilgiler haline çeviren modüllerdir. (Şekil 3.10) Sistemde hangi modüllerin olacağına

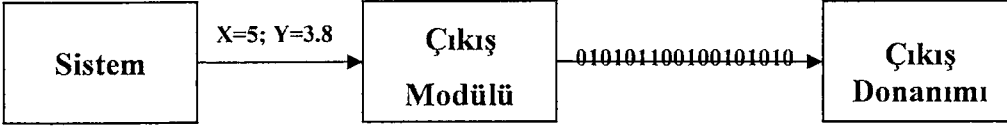
ortamdaki giriş donanımı çeşitlerini kareografin kullanma isteği belirler. Giriş modüllerinin yolladığı verileri kontrol modülü ve çıkış modülleri alır ve kullanırlar. Giriş modüllerinin temel işlevi, donanım ile sistem arasında bir arayüz olmalarıdır. Böylece ortamda yeni bir giriş donanımı kullanılmak istendiğinde toptan tüm sistemin değişmesi gerekmez. Sistemde kullanılabilecek giriş donanımları ilk bölümde ayrıntılı bir biçimde açıklanmıştır. Örnek vermek gerekirse kameradan alınacak görüntü için kullanılacak bir giriş modülünün kameradan gelen veriyi alabilecek API'lere sahip olması gerekmektedir. Daha sonra çeşitli filtrelerden ve özel işlemlerden geçirilen görüntü verisinin içinden dansçıya ait verilerin bulunması ve sistemin diğer modüllerine aktarılması gerekmektedir. Bulunacak veri sahnedeki dansçının (x,y) cinsinden noktasal koordinatı olabileceği gibi, (x_1, y_1, x_2, y_2) cinsinden sahnede dansçının kapladığı alanı belirten bir dikdörtgen de olabilir. İşlenmiş verinin dışında eğer ortaya çıkarılacak medyada kullanılacaksa giriş donanımına gelen ham verinin aktarılması da mümkün olmalıdır. Bütün bunlar giriş modülünü yazacak uzmanın getireceği esnekliklerdendir. (Şekil 3.10)



Şekil 3.10 Giriş modülü

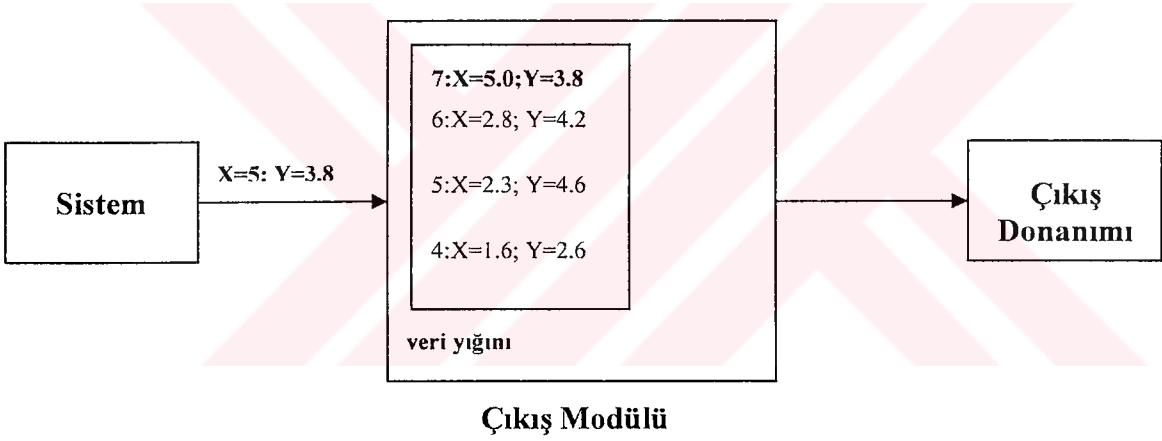
3.3.4 Çıkış Modülü

Çıkış modülleri sistemin sahneye aktaracağı medyayı üretmekle yükümlü olan modüllerdir. Giriş modüllerinden gelen verileri belli oranda parametre olarak kullanarak sahnedeki görsel, işitsel ve fiziksel ortamı oluşturmayı sağlarlar. (Şekil 3.11) Çıkış modülleri sistemde bir donanım için birden fazla olabilir. Her biri kendi ilgilendiği donanıma zamanı gelince aktaracağı bilgiyi aktarır. Belli bir anda hangi çıkış modülünün çalışacağına kontrol modülü kendi iç dinamiklerinden faydalanarak karar verir. (Şekil 3.11)



Şekil 3.11 Çıkış modülü

Giriş donanımından gelen veri sürekli olmadığından ve verinin ne zaman geleceği belli olmadığından çıkış donanımına veri geldiği zaman alıp bir kuyruğa yazmaları gerekmektedir. (Şekil 3.12) Böylece modülün her an veriyi kontrol etmeleri gerekmemekte ve verilerin zamana bağlı değişimini izlemek için bir fırsatı olmaktadır.



Şekil 3.12 Çıkış modüllerinin veri yığınları

3.4 Dağıtık Yapı

3.4.1 Genel Mantık

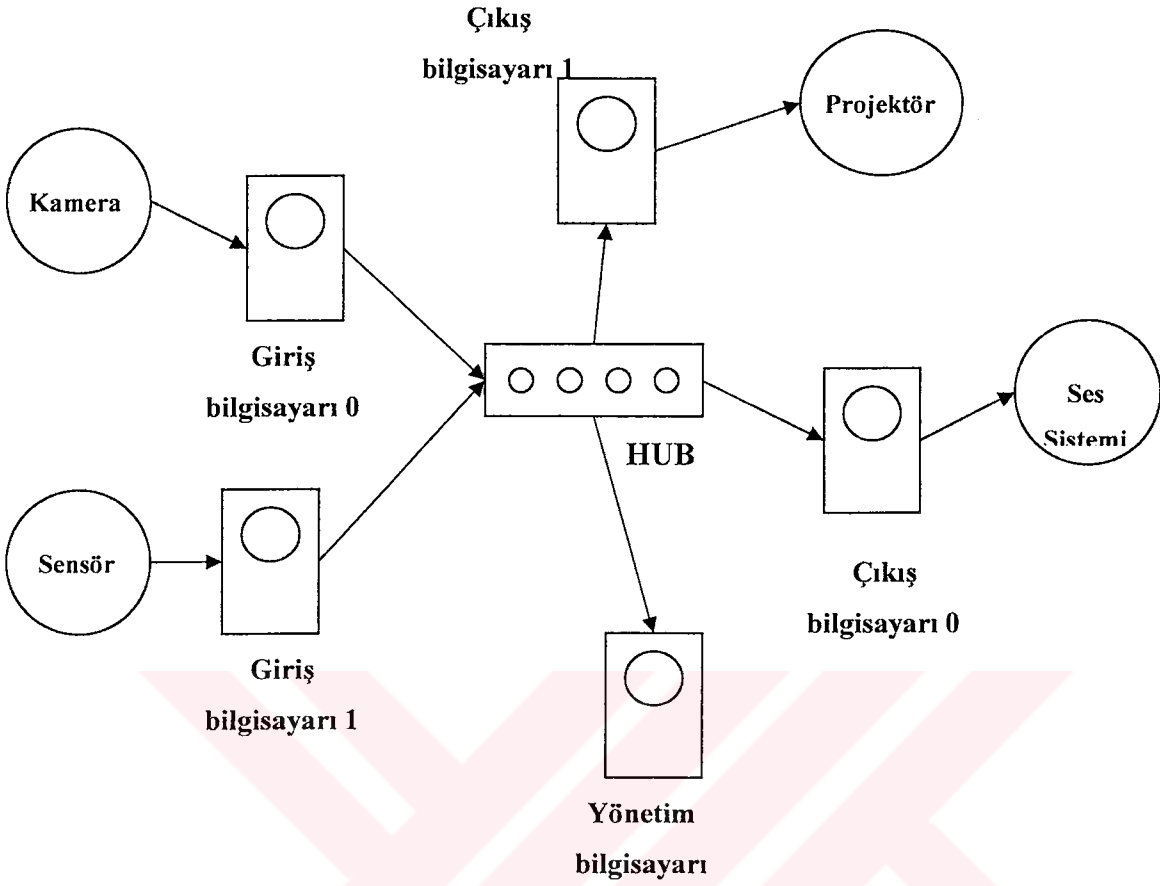
Sistemin tüm giriş ve çıkış işlemlerinin tek bilgisayar kullanacak şekilde tasarlanması uygulamaya geçildiğinde çok önemli bazı kısıtlamalara sebep olabilmektedir. Ortaya çıkabilecek bu kısıtlamaları kaynaklarına göre üç ana başlıkta toplayabiliriz:

- Bilgisayar işlem gücü: Tasarlanan etkileşim sisteminde gerek alınan verinin işlenmesi (özellikle video) gerek etkileşim sonucu oluşacak medyanın hazırlanması için gereken

işlem gücü tek bilgisayarın kaldıramayacağı güçte olabilir. Böyle bir durum sistemde yavaşlamalara sebep olacağı için gerçek zamanlı etkileşimi tehlikeye atabilir.

- Bilgisayar donanımı: Sistemde kullanılacak bilgisayarın çıkış donanımı yeterli olmayabilir. Örneğin, sahneye birden fazla projektörle farklı görüntüler yansıtılmak istendiğinde kullanılan bilgisayarın VGA çıkışları yetersiz olduğundan böyle bir sunuma izin vermeyebilir. Aynı şey giriş donanımı için de geçerlidir. Örnek olarak, birden fazla kameradan gelen veriyi inceleyebilmek için özel donanım kullanılması zorunludur.
- Çıkış donanımı: Bilgisayara girecek ve bilgisayardan çıkacak olan verinin aktarım yapılacağı donanımların sahnenin değişik bölgelerinde konumlandırılması gerekebilir. Böyle bir durumda tek bir bilgisayardan bütün bu donanıma veri gönderecek kabloların olması gerekmektedir. Kablo uzunluğu pek çok donanımda sorun olmamasına rağmen VGA, USB ve diğer bazı kablo çeşitlerini kullanan çıkış donanımlarının kullanımında için ciddi kısıtlamalar getirmektedir.

Bütün bu kısıtlamaları ortadan kaldırmak için sistem tek bilgisayara bağımlı olmaktan kurtarılmış ve koreografa ağ üzerinde birleştirilmiş birden fazla bilgisayar kullanarak dağıtık bir sistem yaratma imkanı verilmiştir. (Şekil 3.13)



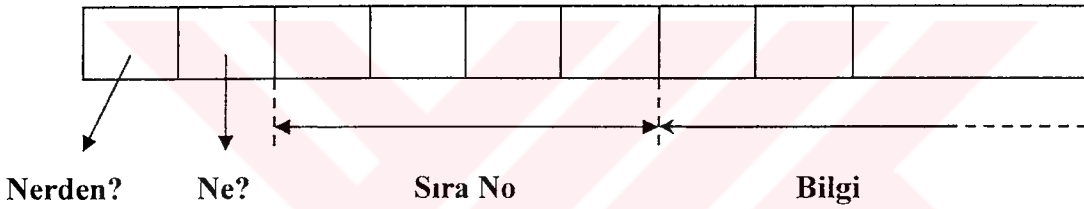
Şekil 3.13 Dağıtık yapı

Dağıtık yapının kullanılabilmesi için kareografi programında etkileşim sistemi hazırlanırken değişik bilgisayarlara dağıtılacak etkileşimlerin farklı çıkış kanallarında gruplanması gerekmektedir. Derleme işleminde kullanıcı karşısına gelen diyalog penceresi kullanılarak çıkış kanalları için derlenen kod ve veri dosyaları diğer bilgisayarlara dağıtılabilir ve dağıtılan bu kodlar uzaktan çalıştırılabilir. Tek bilgisayardan ortamdaki diğer bilgisayarlara ulaşabilmek için her bilgisayarda ağdan dosya alıp kaydetme ve emir alıp uygulama işlerini yapabilecek küçük ajan programlar çalıştırılmaktadır.

3.4.2 Mesaj Yapısı

Sistemin ağ üzerinde dağıtık bir yapıda çalışabilmesi için giriş, çıkış ve karar verme modüllerinin de bu yapıya uyum sağlaması gerekmektedir, zira modüllerin arasında normalde çeşitli fonksiyonlar çağırılarak yapılabilecek iletişimin bu ağ koşullarında gerçekleşmesi mümkün değildir. Bu yüzden modüllerin ağ üzerinde haberleşebilmesi mesajlar üzerine kurulmuştur ve bu iş için basit bir mesaj yapısı tasarlanmıştır. Mesajların ağ üzerinden iletimi

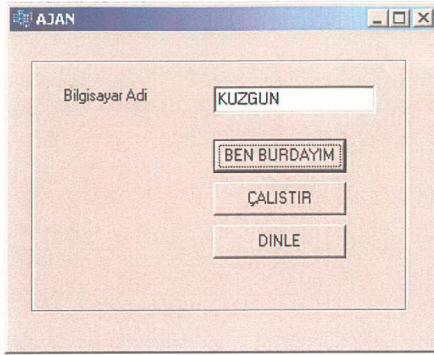
için Processing için yazılmış OSC tabanlı veri gönderimini destekleyen oscP5 kütüphanesi kullanılmıştır. (Wright vd, 2003) Şekilde hazırlanan mesaj yapısı görülmektedir. Bu yapıda ilk byte bilginin sistemin hangi giriş modülünden geldiğini göstermektedir. İkinci byte ise gönderilen bilginin birinci bytetaki giriş donanımının ne tip bir parametresi olduğunu gösterir. Bu iki veriye dayanarak örneğin gelen paketin ses donanımından gelen yükseklik bilgisi olduğu anlaşılabilir. Alıcı modülün o giriş modülünün adı geçen giriş parametresiyle işi yoksa paketi dikkate almaz. Bundan sonraki iki byte gönderilen paketin sıra numarasıdır. Her giriş modülü ağa veri çıkarırken gönderdiği pakete 0-65535 arası bir sıra numarası verir. Böylece paketlerin X+1. sırada yollanan bir paketin X. sırada yollanan bir paketten daha önce varması durumunda bu durum sezilebilir. Sıra numarasından sonraki bytelar veri bytelarıdır. İlk iki bytetan veri tipi anlaşıldığı için bu byte dizisinden gerekli veri çıkartılabilir. (Şekil 3.14)



Şekil 3.14 Ağ mesajı yapısı

3.4.3 Ajan Programı

Kareografi programı tarafından ağdaki diğer bilgisayarlara dosya yollamak, oradaki programların istendiği zaman çalışmasını sağlamak için her bilgisayarda küçük bir ajan programının çalışması gerekmektedir. Bu program yardımıyla kareografi programı ağdaki her bilgisayarı görüp üzerinde işlem yapılabilir. Bir bilgisayarı sisteme tanıtmak için ajan programını çalıştırıp bilgisayara bir isim vermeli ve “BEN BURDAYIM” düğmesi tıklanmalıdır. Bu işlem sonucunda ajan programı ağa kendi adını ve ip adresini içeren bir mesaj yayınlamaktadır. Bu mesaj kareografi programının ipsi bilinmediğinden ağdaki bütün bilgisayarlara gönderilmektedir. Mesajı alan kareografi programı artık sistemde hazır bir bilgisayar olduğunu anlamakta ve kullanıcının isteği doğrultusunda bu bilgisayara dosya ve emir yollayabilmektedir. Ajan programı iki portu devamlı olarak dinleyerek birinden gelen dosyaları sisteme yazmakta, diğerinden gelen emirleri ise uygulamaktadır. (Şekil 3.15)



Şekil 3.15 Ajan programının arayüzü

3.5 Sahne Etkileşimi Tasarım Programı

3.5.1 Arayüzün Özellikleri

Kareografin önerilen sistemde yer alan modülleri kurgulayabilmesi ve sistemi çalışır hale getirebilmesi için bir kurgu arayüzü hazırlanması gerekmektedir. Kurgu için görsel bir arayüzün kullanılması pek çok fayda sağlamaktadır. Hazırlanan kurgu uygulaması efektlerin hangi sırayla gösterileceğini, görsel ve işitsel parametreleri ve efektlerin birbirleri arasında nasıl geçiş yapacaklarını kareografin programlamaya bulaşmadan gerçekleştirmesini sağlamaktadır. Bu uygulama koreografa görsel bir arayüz sunar. Uygulamanın görsel arayüzünü kullanarak koreograf ortama yeni efektler ekleyebilir, eklediği efektlerin parametrelerini ve etkileşim seçeneklerini değiştirebilir, ve efektlerin birbirleri arasında hangi koşullarda geçiş yapabileceğini ayarlayabilir. Bunun dışında kullanıcı tarafından ortama yeni giriş donanımları için giriş modülleri eklenmesi de mümkün olmaktadır. Hazırlanan giriş ve çıkış modülleri olanaklara göre birden fazla bilgisayara dağıtılabilir. Kullanıcı tek bir tuşa basarak hazırlamış olduğu ortamı kod olarak çıktı alabilmekte ve ortamdaki diğer bilgisayarlara dağıtabilmektedir. Yani kareografinin her türlü yönetimi bu arayüzde yapılabilmektedir. Arayüzün temel işlevlerini listeleyecek olursak:

- Yeni giriş ve çıkış modülleri ekleme
- Çıkış kanalları yaratma
- Zamanlayıcı üzerinde yeni süreçler tanımlama
- Yaratılan süreç dilimlerinin içinde etkileşimlerin tanımlanması

- Etkileşimleri birbirine bağlayan geçiş koşullarının tanımlanmasıyla durum makinesi yapısının oluşturulması
- Hazırlanan çıkış kanallarının bilgisayarlara atanması
- Tasarlanan sisteme göre çıkış kodunun oluşturulması
- Hazırlanan çıkış kodunun bilgisayarlara dağıtılması

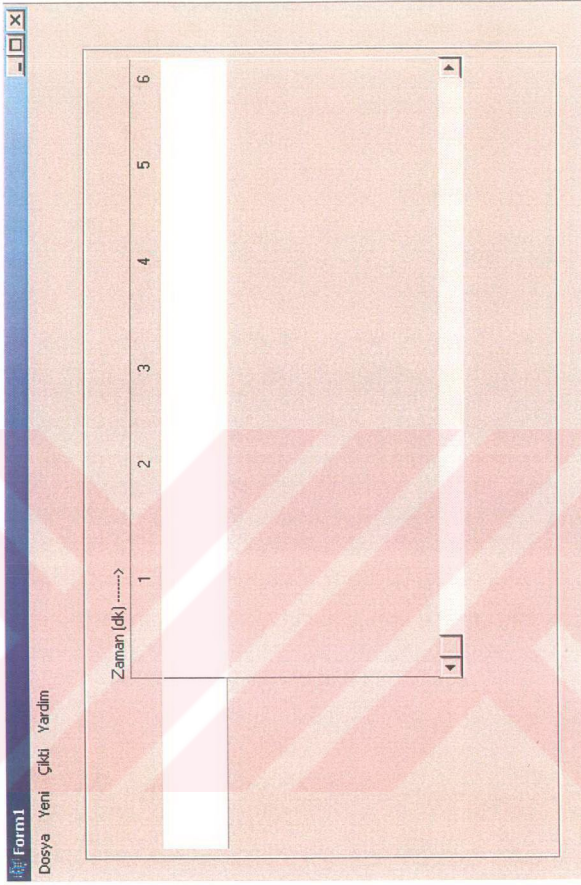
Sisteme eklenen her giriş modülü aynı zamanda kendine özgü geçiş koşullarını da yanında getirir. Kabul etmek gerekir ki görsel bir aygıttan alınan girişin getireceği koşullarla işitsel bir aygıttan alınan girişin getireceği koşullar aynı olmayacaktır. Sisteme yeni bir giriş modülü eklemek için “inputModules” klasörünün altına giriş modülü dosyalarını eklemek yeterlidir. Arayüz programı her açıldığında bu klasörün altındaki dosyaları tarayarak ayar dosyalarından gerekli bilgileri alır ve kendi içinde veri yapılarını oluşturur. Böylece kareografin geçiş koşullarını tanımlarken kullanacağı seçeceği giriş aygıtları ve arayüzler için gereken temel bilgiler hazırda bulundurulmuş olur.

Giriş modülleri gibi çıkış modüllerinin dosyaları da “outputModules” klasörüne konmalıdır. Böylece program aynı giriş modüllerinde olduğu gibi çıkış modüllerinin ayar dosyalarını da açılışta okuyarak içindeki veri yapılarına yazar. Programın çıkış modüllerinden öğreneceği bir diğer şey de ortamda hangi çıkış modülü çeşitleri olduğudur. Bu bilgi koreograf yeni bir çıkış kanalı yarattığında bu kanalda sadece aynı çıkış donanımına veri gönderen modüllerin bulunmasını sağlamada kullanılacaktır.

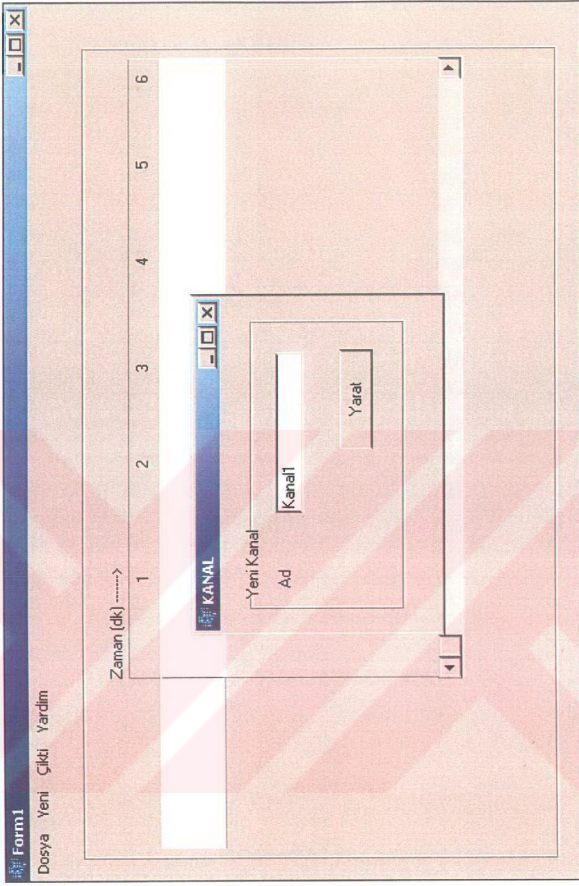
3.5.2 Arayüzün Kullanımı

3.5.2.1 Yeni Çıkış Kanalı Yaratma

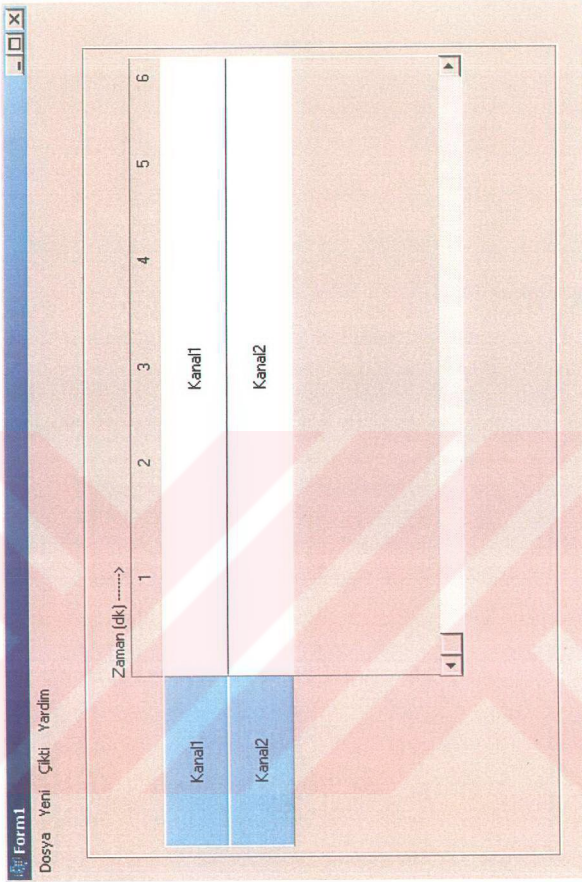
Sisteme mönüdeki “Yeni” seçeneğinden yeni bir çıkış kanalı eklenebilmektedir. Bu yeni çıkış kanalı eklenirken kanala bir isim verilmeli ve bir çıkış donanımı seçilmelidir. Yeni bir çıkış kanalı eklendiğinde bu kanal için sistemde bir zaman çizgisi oluşturulur. (Şekil 3.16) (Şekil 3.17) (Şekil 3.18)



Şekil 3.16 Boş tasarım programı



Şekil 3.17 Yeni kanal yaratma arayüzü



Şekil 3.18 Yaratılan kanallar

3.5.2.2 Yeni Süreç Yaratma

Sisteme eklenen yeni kanalın üstüne sağ tıklamak suretiyle kanal içinde süreçler yaratmak mümkündür. (Şekil 3.19) (Şekil 3.20) Süreçler ilk yaratıldıklarında içleri boş olarak yaratılırlar. Yaratılan bu süreçleri üstüne tıklayıp taşıyarak zaman içinde yer değiştirmek, ve süreçlerin başlangıç ve bitiş zamanlarını kenarlarındaki aktif bölgelerden çekiltilerek değiştirmek mümkündür. Bir çıkış kanalının içindeki süreçlerin birbiriyle çakışması mümkün değildir. Örnek vermek gerekirse, Süreç1 4-6 dakika arasında yer alıyorsa, aynı çıkış

kanalında yeni yaratılacak süreç bu aralıkla kesişemez, 4. dakikadan önce bitmek yada 6. dakikadan sonra başlamak zorundadır. Eğer bir çakışma yaratılmak isteniyorsa yeni süreç için başka bir çıkış kanalı yaratılması gerekmektedir. Bu yüzden kareografi planlanırken çıkış kanalları ve süreçler dikkate alınmalıdır. Burada unutulmaması gereken bir konu da farklı bilgisayarlarda çalışacak olan süreçlerin aynı çıkış kanalı içinde yaratılmamasıdır. Sistemin varsayımı üzerine bir bilgisayarda birden fazla çıkış kanalı çalıştırılabilir, ancak bir çıkış kanalı birden fazla bilgisayara dağıtılamaz. Yaratılan süreçlerin üzerine sağ tıklayınca çıkan mönüden silme işlemi yapılabilir.



Form1 Dosya Yeni Çıktı Yardım

Zaman (dk) ----->

1 2 3 4 5 6

Kanal1 Kanal2

Yeni Süreç

Ad SUREC00

Kanal Kanal1

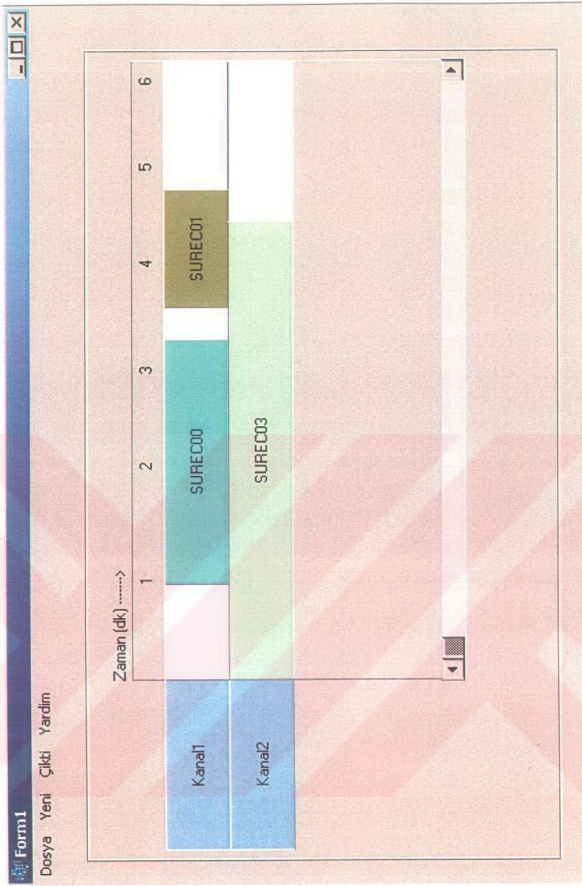
Kaçtan 1

Kaç 3

Renk Sarı

Yarat

Şekil 3.19 Yeni süreç yaratma arayüzü

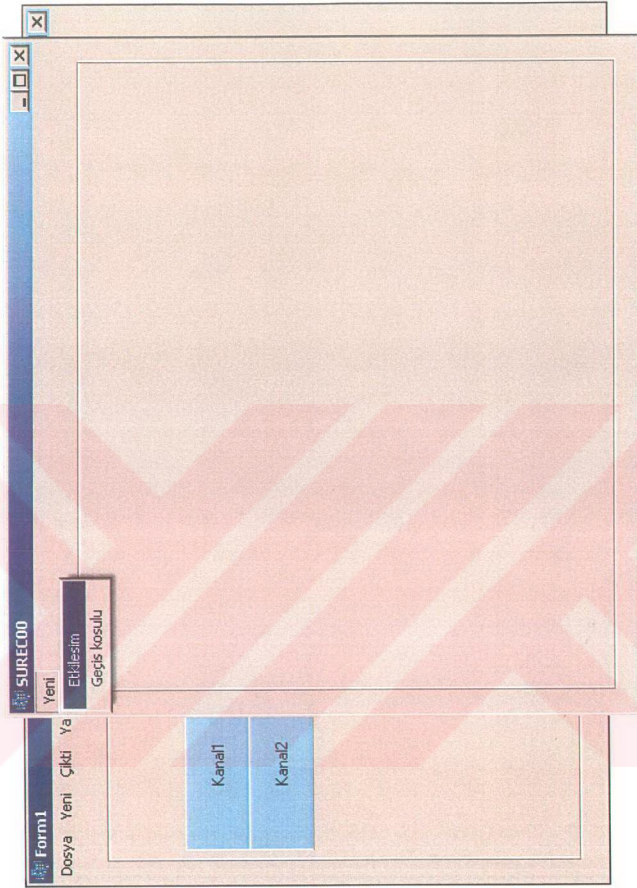


Şekil 3.20 Yaratılan yeni süreçler

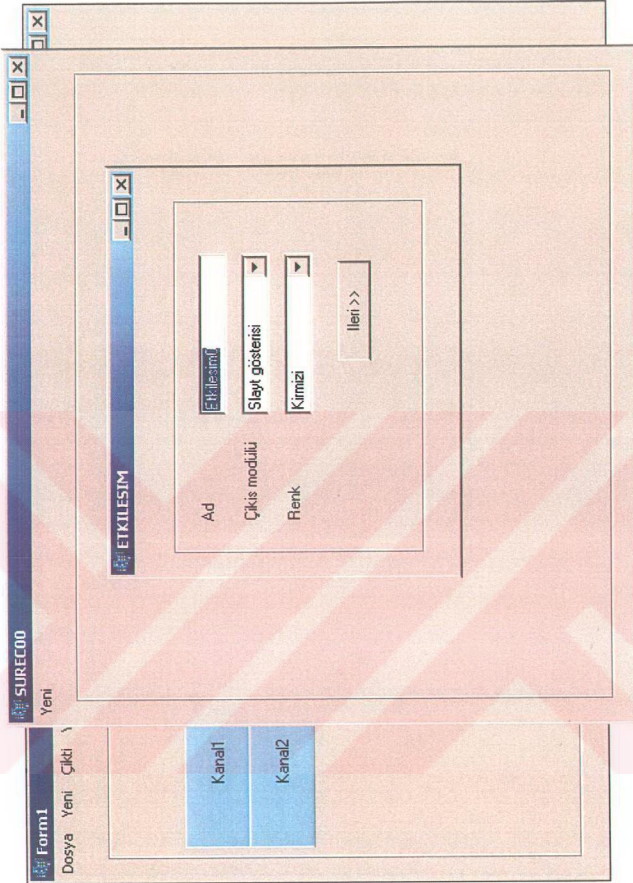
3.5.2.3 Yeni Etkileşim Yaratma

Süreçlerin içinde durum makinelerini yaratmak için sürecin üzerine çift tıklamak gerekmektedir. Bu işlem ile açılan pencereye yukarıdaki mönüyü kullanarak yeni etkileşim eklemek mümkündür. (Şekil 3.21) Yeni etkileşim eklemek için öncelikle yukarıdaki mönüden seçim yaptıktan sonra gelecek pencerede eklenecek etkileşimin türünü ve ismini belirtmek gerekmektedir. Bu mönü sabit olmayıp “ouputModules” klasöründen okunan etkileşim çeşitlerinden oluşturulmuştur. Etkileşimin türünü seçip sistem içinde kullanılacak bir isim

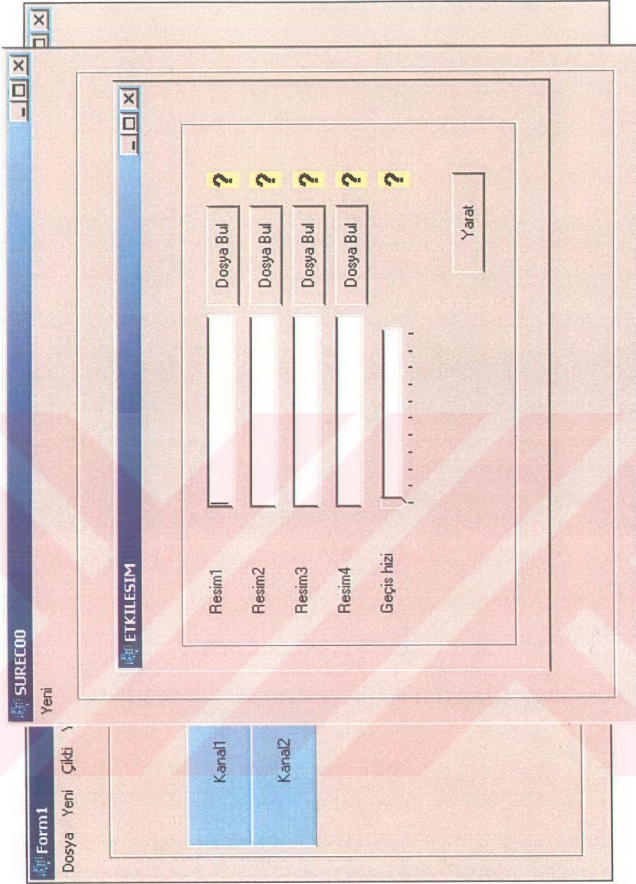
verdikten sonra “İleri” butonuna basıldığında kullanıcının karşısına etkileşim parametrelerinin ayarlanabildiği bir başka pencere açılır. (Şekil 3.22) (Şekil 3.23) Bu penceredeki giriş elemanları seçilen etkileşime ait çıkış modülünün ayar dosyasından elde edilmiştir. Kullanıcı sahneye istediği sayıda etkileşim ekleyebilir, sistem dinamik veri yapılarıyla tasarlandığından herhangi bir sayı kısıtlaması yoktur. Yeni bir etkileşim eklendiğinde süreç içinde büyük bir daire şeklinde gösterilir, bu gösterim şekli durum makinesi yaklaşımından ileri gelmektedir. (Şekil 3.24) Etkileşimler yaratıldıktan sonra sürükle-bırak yöntemiyle pencere içinde istenildiği gibi taşınabilir. Etkileşim seçenekleri değiştirilmek istendiğinde etkileşim dairesi üstüne çift tıklanması başta karşılaşılan etkileşim seçeneklerini kullanıcının karşısına tekrar getirmektedir, böylece seçenekler tekrar tekrar değiştirilebilmektedir. İlk yaratılan etkileşim varsayılan olarak sürecin giriş etkileşimi olarak atanır. Giriş okunun gösterdiği etkileşim giriş etkileşimidir. Bunu değiştirmek için giriş yapılmak istenen etkileşimin üstüne sağ tıklanıp ve ilgili seçeneğin seçilmesi gerekmektedir. Etkileşimi silmek içinse üzerine sağ tıklayınca açılan mönüden ilgili seçeneği seçmek yeterlidir. Birden fazla aynı türden fakat farklı parametrelili etkileşim yaratılması o etkileşimin kullanıcıya tanıdığı seçenekleri arttırmaya ve çeşitlilik sağlamaya yardımcı olur.



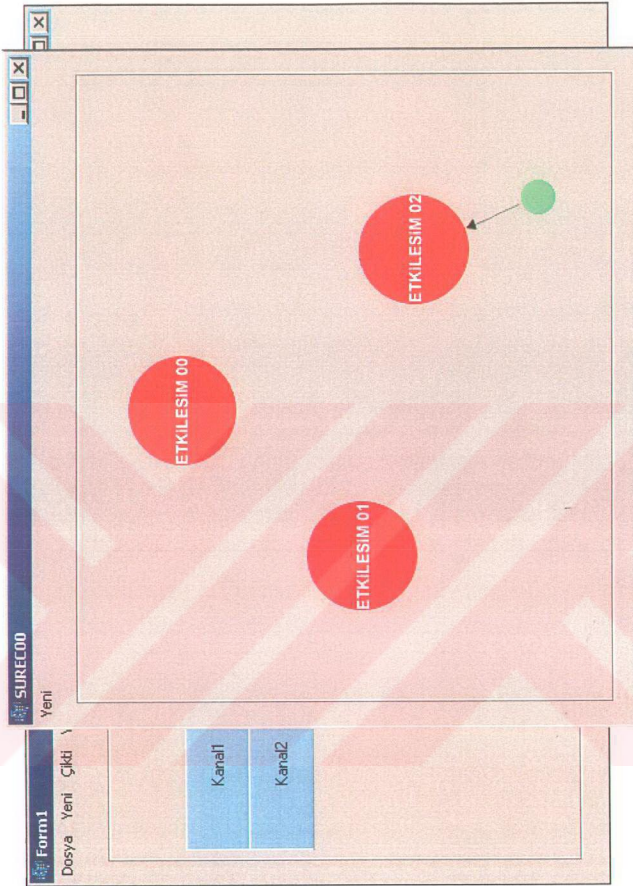
Şekil 3.21 Boş süreç penceresi



Şekil 3.22 Yeni etkileşim yaratma arayüzü - 1



Şekil 3.23 Yeni etkileşim yaratma arayüzü - 2



Şekil 3.24 Yaratılan yeni etkileşimler

3.5.2.4 Yeni Geçiş Koşulu Yaratma

Sürece eklenen etkileşimlerin birbiri arasında geçiş yapıp durum makinesi modelini oluşturmaları için geçiş koşullarına ihtiyaç vardır. Geçiş koşulları bir etkileşimden başka bir etkileşime geçmek için giriş modüllerinden veya sistem içindeki zaman sayaçlarından gelen verinin tetiklediği küçük modüllerdir. Tetiklenen bu modüller etkileşim sırasında gerekli koşullar sağlandığında ait oldukları sürecin durum makinesine durumun değişmesi gerektiği

mesajı ile kendisinin ve ait olduğu etkileşimin numarasını iletir. Geçiş koşullarının bir çıkış bir de varış etkileşimi vardır. Bu iki parametre farklı olabildiği gibi aynı etkileşimin yeniden başlatılması bir anlam ifade edecekse aynı da olabilir. Herhangi bir etkileşime hiçbir geçiş koşulu varmıyorsa o etkileşim hiçbir zaman devreye girmeyecek demektir, bu yüzden etkileşimleri birbirine bağlarken dikkatli olmak gerekmektedir. Var olan bir etkileşime yeni bir durum eklemek için yine yukarıdaki m n den ilgili se enek se ilmeli ve gelen yeni m n de hangi iki durum arasında hangi giriř mod l n  kullanarak bir ge iř yapılacađı se ilmelidir. (řekil 3.25) “İleri” butonuna basıldıđında kullanıcının karřısına se tiđi giriř mod l n n ayar dosyalarından okunan ge iř kořulu se enekleri gelir. (řekil 3.26) Bu pencereden kořul ayarlanıp “Yarat” d đmesine basıldıđında se ilen kaynak ve hedef etkileřiminin arasında bir ok iřareti ve de kaynak etkileřimin yanında ufak bir daire řeklinde ge iř kořulu durum makinesi i indeki yerini almıř olur. (řekil 3.37) Ge iř kořulunda herhangi bir deđiřiklik yapılması gerektiđi zaman durum makinesinde kendi dairesinin  st ne  ift tıklamak ilk bařtaki ayar penceresini geri getirir. Giriř kořulunu silmek i inse sađ tık m n s nden ilgili se eneđi se mek yeterlidir.

SUREC00

Yeni

Form1

Dosya Yeni Çıktı

GEÇİŞ KOSULU

GİRİŞ KOSULU

Giris modülü: Kamera giris

Geçiş kosulu türü: Cisim yüksekligi

Hangi etkileşimden?: ETKİLESİM00

Hangi etkileşime?: ETKİLESİM02

İleri >>

Kanal1

Kanal2

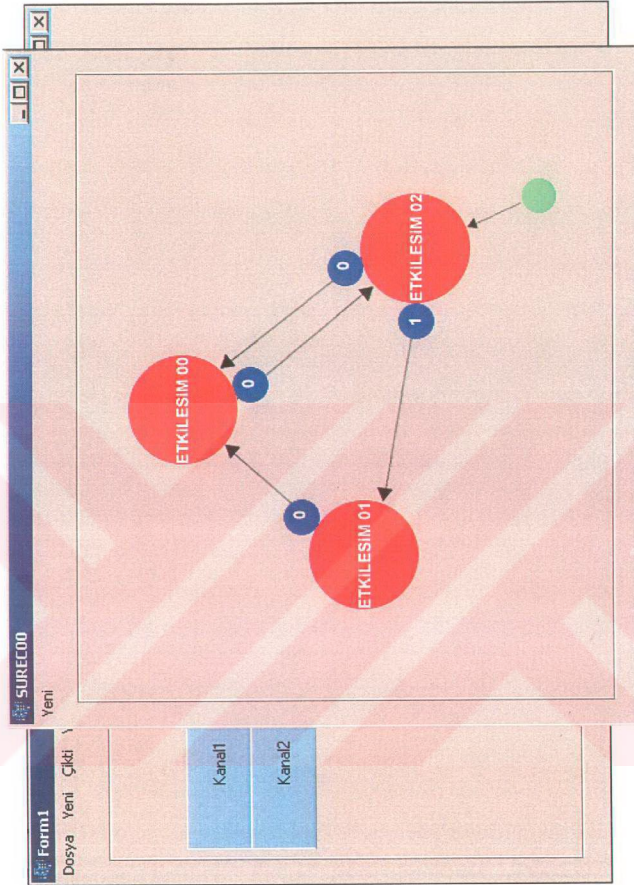
ETKİLESİM00

ETKİLESİM00

Şekil 3.25 Yeni geçiş koşulu yaratma arayüzü - 1

The image shows a software interface for creating a new transition rule. The main window is titled 'Yeni' (New) and has a menu bar with 'Form1', 'Dosya', 'Yeni', and 'Çıktı'. The main area contains a red circle with the text 'ETKİLEŞİM 00' and a blue bar with the text 'GEÇİŞ KOŞULU'. Below this, there is a section for defining the transition rule, including a question mark icon, a text input field, a dropdown menu labeled '< Cisim yüksekliği <', and a 'Yarat' button. The background shows a partially visible 'Form1' window with 'Dosya', 'Yeni', and 'Çıktı' menus, and two blue buttons labeled 'Kanal1' and 'Kanal2'.

Şekil 3.26 Yeni geçiş koşulu yaratma arayüzü - 2



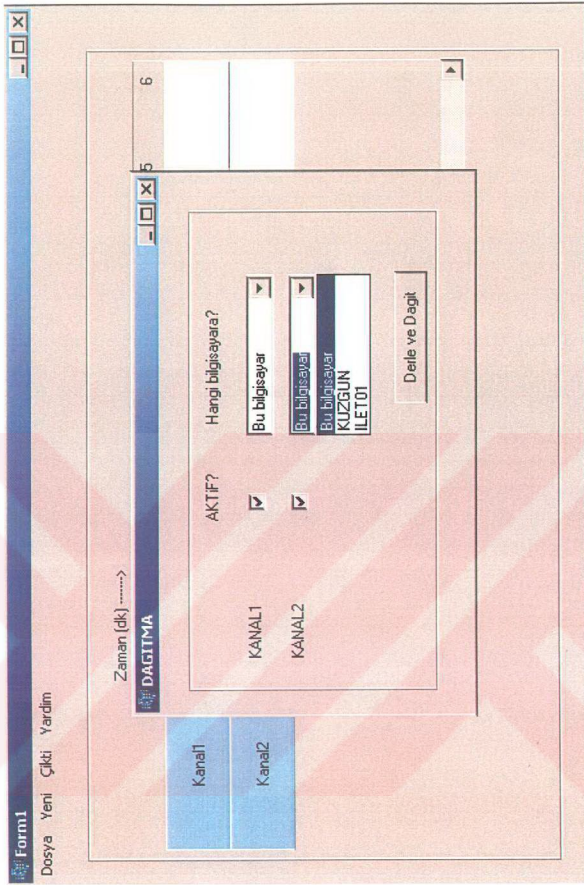
Şekil 3.27 Yaratılan yeni geçiş koşulları

3.5.2.5 Derleme ve Dağıtma

Hazırlanan sistemin kod olarak çıktısının alınması için ana pencerenin üst mönüsünden “Derle” seçeneğinin seçilmesi gerekmektedir. Bu seçenek seçildiğinde sonuç kodu oluşturulmaya başlanmadan önce program kullanıcıya hangi çıkış kanallarının derleneceğini ve kanalların sistemdeki bilgisayarlara dağılımını sorgulayan bir pencere açacaktır. (Şekil 3.28) Eğer daha önceden tanımlanmış birden fazla bilgisayar yoksa derlenen kanal tasarımın

yapıldığı bilgisayarda çalıştırılır. Sistemde farklı bilgisayarlar varsa, yani dağıtık çalışma yapılacaksa, derlenen kod belirtilen bilgisayarlara dağıtılır. Ancak derleme sırasında ortaya çıkan kodun aktarımı için diğer bilgisayarlarda ajan programların çalıştırılması gerekmektedir. Bütün bu ayarlar tamamlandıktan sonra “Derle&Dağıt” tuşuna basıldığında kareografi editöründe hazırlanan sistemin seçilen parçaları, ilgili dosyalardan toplanan kod kullanılarak Java kodu halinde bir araya getirilir. Bu hazırlanan kod Java derleyicisiyle derlenir ve ortaya çıkan sınıf dosyaları ile kullanılan veri dosyaları paketlenerek ajan programlarla bağlantıya geçen kareografi programı tarafından bilgisayarlara dağıtılır.

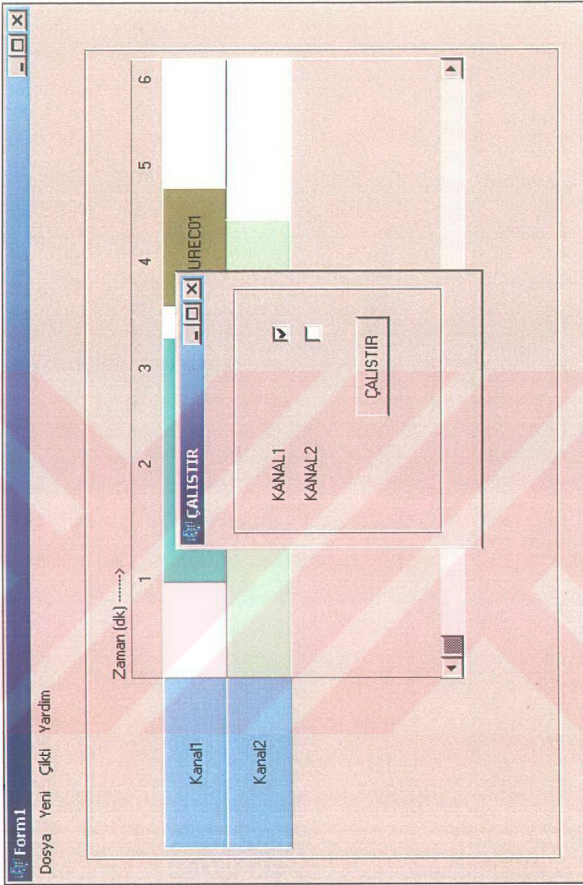




Şekil 3.28 Derle ve Dağıt arayüzü

3.5.2.6 Çalıştırma

Üretilen kodun dağıtılması bittikten sonra son yapılacak iş kodun çalıştırılmasıdır. Kullanıcı ana pencerenin üst menüsünden “Çalıştır” butonuna basarak karşısına gelen pencereden seçeceği çıkış kanallarından bazısını yada hepsini çalıştırabilir. (Şekil 3.29) “Çalıştır” butonuna basıldığında seçilen çıkış kanallarının dağıtıldığı bilgisayarlardaki ajan programlarına ellerindeki kanalları çalıştırmaları emri gider. Bu emir karşısında ajan programları Java programını kullanarak ellerindeki paket kodu çalıştırmalar.



Şekil 3.29 Çalıştırma arayüzü

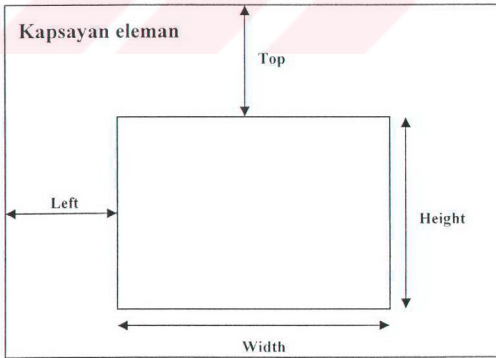
3.5.3 Teknik Ayrıntılar

Kareografi programının tasarımında pek çok teknik güçlükle uğraşmıştır. Bunlardan biri de tasarım arayüzünün hazırlanmasıdır. Tasarım programı Borland C++ Builder 6.0 kullanılarak hazırlanmıştır. Ekran kayıtlarından da görüldüğü üzere kullanılan arayüz elemanlarının çoğu standart elemanlar değildir. İhtiyaç duyulan arayüz elemanları var olan arayüz elemanlarında

türetilerek hazırlanmıştır.

Arayüzün ana penceresinde çıkış kanallarının içinde yaratılan her sürecin kanal içinde ileri veya geri hareket ettirilebilmesi ve sol ve sağ kenarlarından boyutunun değiştirilebilmesi gerekmektedir. Bunun dışında, hazırlanan bir projede pencere içinde yaratılacak tüm süreçleri aynı anda görmek mümkün olmayacaktır. Bunun için çıkış kanallarının altına bir sürgü konmuş ve süreçlerin belli bir anda sadece belirli bir miktarının görünmesi sağlanmıştır. Altaki sürgüyü sağa sola hareket ettirince çıkış kanalı üzerinde zamanca daha geri veya ilerideki süreçleri de görmek gerekmektedir. Bu özellikleri sağlayan bir temel eleman bulunmadığı için var olan elemanların birleşimiyle yeni bir eleman elde etmek gerekmektedir. Borland'ın "Panel" isimli temel elemanı bu iş için uygundur, çünkü iç içe birden fazla panel yaratarak her panele farklı görevler yüklemek mümkündür, iç içe geçmiş panellerde içteki panelin sadece belli bir kısmının görünmesi de sağlanabilmektedir. Borland arayüz elemanlarının yerini ve boyutunu kontrol edebilmek için 4 ana değişken kullanmaktadır (Şekil 3.30):

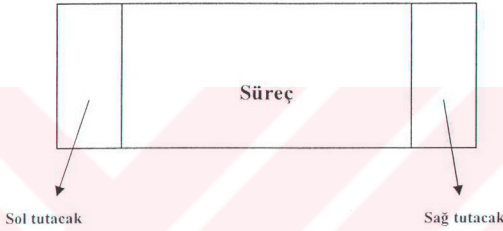
- Left: Elemanın içinde bulunduğu kapsayan elemanın sol kenarına olan uzaklığı
- Top: Elemanın kapsayan elemanın üst kenarına olan uzaklığı
- Width: Elemanın piksel cinsinden genişliği
- Height: Elemanın piksel cinsinden yüksekliği



Şekil 3.30 Borland C++ Builder'da arayüz elemanlarının temel özellikleri

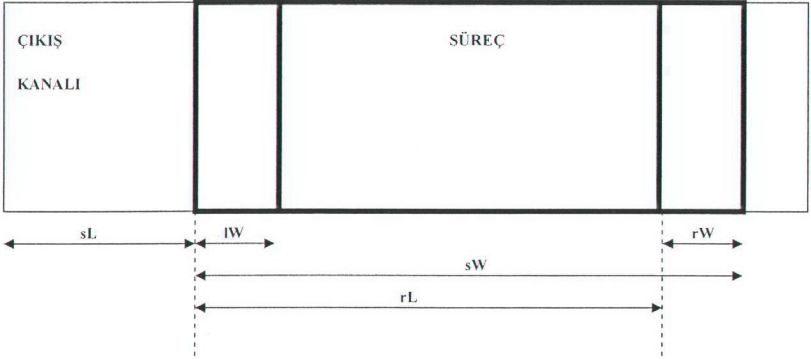
Bu değişkenleri kullanarak süreç modülü için istenen taşınabilirlik ve ölçeklenebilirlik

özelliklerini gerçekleştirmek gerekmektedir. Elemanın sağ ve soldan çekiştirilerek ölçeklenebilmesi için süreç panel'inin içinde sağ ve sol yanlarda aktif bölgelerin bulunması gerekmektedir. Bu aktif bölgeleri sağlamak için panelin içinde sol ve sağa dar tutacak paneller eklenmiştir, ve farkedilmesin diye panellerin rengi ana panelle aynı renkte düzenlenmiştir. (Şekil 3.31) Bu panellerin üstüne tıklayarak kullanıcı ölçekleme işlemini de başlatmış olmaktadır. Fareyi sağa sol sürükleyen kullanıcı tutacakların yerini değiştirdiği gibi süreç panelinin genişliği ve sol kenara olan uzaklıklarını da değiştirmektedir.



Şekil 3.31 Süreç elemanının yapısı

Tutacaklardan tutulup çekiştirilirken hangi tutacağın, ne tarafa doğru çekiştirildiğine göre çeşitli hesaplamalarla yeni boyutlar hesaplanır ve farenin hangi elemanın üstünde olduğuna göre de hesaplamaların tekrarlanması gerekmektedir. Aşağıda şekilde görünen yapı için çeşitli durumlara göre elemanların yeni boyutlarının hesaplanması için gereken formüller verilmektedir. (Şekil 3.32)



Şekil 3.32 Hesaplama da kullanılacak parametreler

sL: Sürecin çıkış kanalının sol kenarına olan uzaklığı

sW: Sürecin genişliği

sM: Farenin sürecin sol kenarına olan uzaklığı

IW: Sol tutacağın genişliği

IM: Farenin sol tutacağın sağ kenarına olan uzaklığı

rL: Sağ tutacağın sürecin sol kenarına olan uzaklığı

rW: Sağ tutacağın genişliği

rM: Farenin sağ tutacağın sol kenarına olan uzaklığı

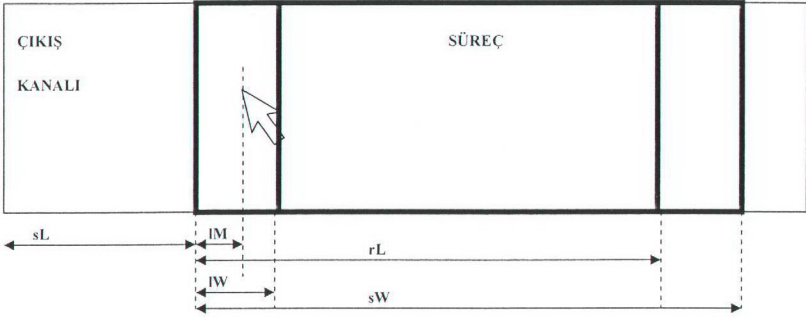
kM: Farenin çıkış kanalının sol kenarına olan uzaklığı

Sol tutacaktan sola veya sağa doğru çektiştirildiğinde fare tutacağın üstündeyse (Şekil 3.33):

$$sL = sL + (IM - IW/2)$$

$$sW = sW - (IM - IW/2)$$

$$rL = rL - (IM - IW/2)$$



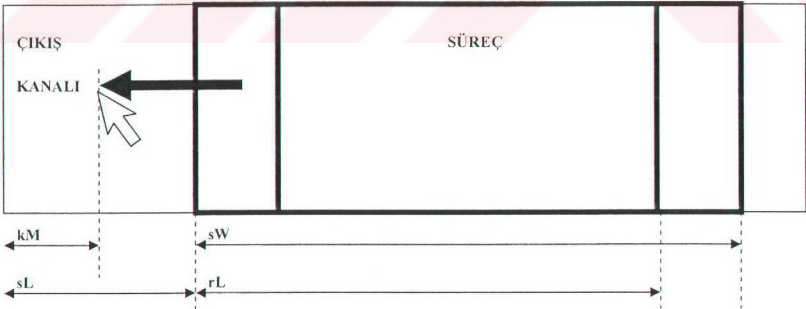
Şekil 3.33 Sol tutacaktan çekiştirirken fare sol tutacaktırsa

Sol tutacaktan sola doğru çekiştirildiğinde fare çıkış kanalının üstündeyse (Şekil 3.34):

$$sL = sL - kM - IW/2$$

$$sW = sW + kM + IW/2$$

$$rL = rL + kM + IW/2$$



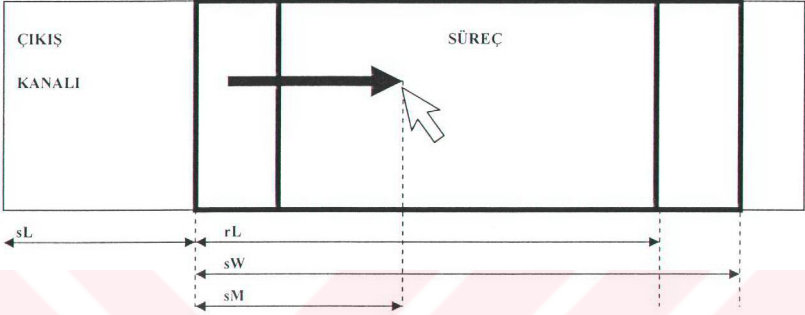
Şekil 3.34 Sol tutacaktan çekiştirirken fare çıkış kanalındaysa

Sol tutacaktan sağa doğru çekiştirildiğinde fare sürecin üstündeyse (Şekil 3.35):

$$sL = sL + sM - IW/2$$

$$sW = sW - sM - lW/2$$

$$rL = rL - sM - lW/2$$

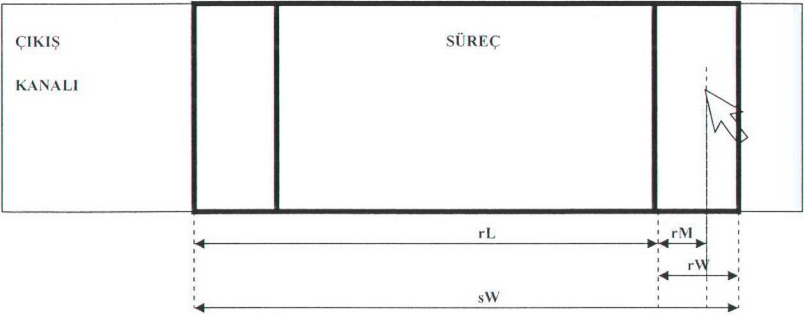


Şekil 3.35 Sol tutacaktan çekiştirirken fare süreçteyse

Sağ tutacaktan sola veya sağa doğru çekiştirildiğinde fare tutacağın üstündeyse (Şekil 3.36):

$$sW = sW + (rM - rW/2)$$

$$rL = rL + (rM - rW/2)$$

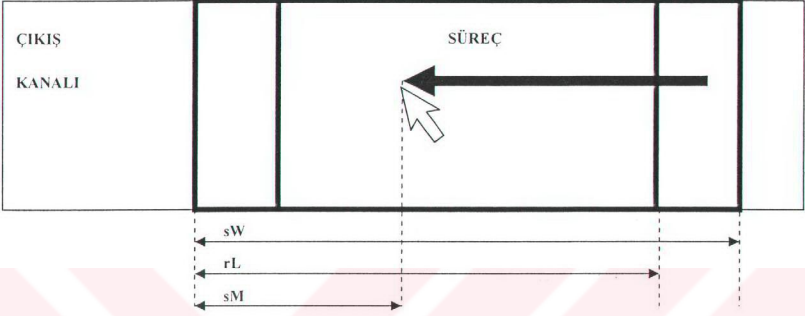


Şekil 3.36 Sağ tutacaktan çekiştirirken fare sağ tutacaktaysa

Sağ tutacaktan sola doğru çekiştirildiğinde fare sürecin üstündeyse (Şekil 3.37):

$$sW=sM+rW/2$$

$$rL=sM-rW/2$$

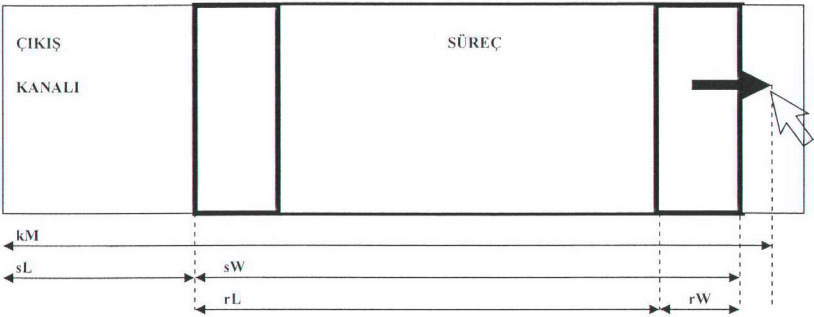


Şekil 3.37 Sağ tutacaktan çekiştirirken fare süreçteyse

Sağ tutacaktan sağa doğru çekiştirildiğinde fare çıkış kanalının üstündeyse (Şekil 3.38):

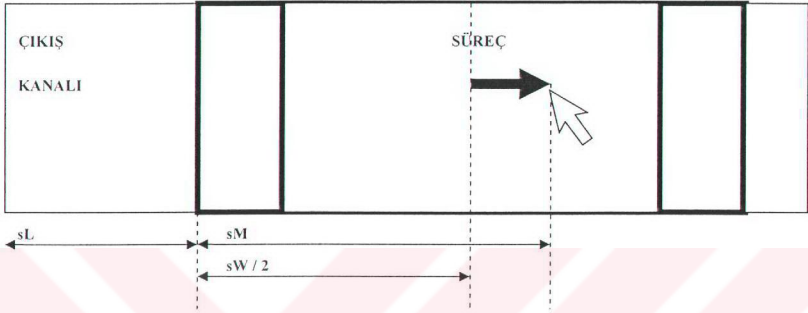
$$sW=kM+rW/2-sL$$

$$rL=rL+kM-sW-sL+rW/2$$



Şekil 3.38 Sağ tutacaktan çekiştirirken fare çıkış kanalındaysa

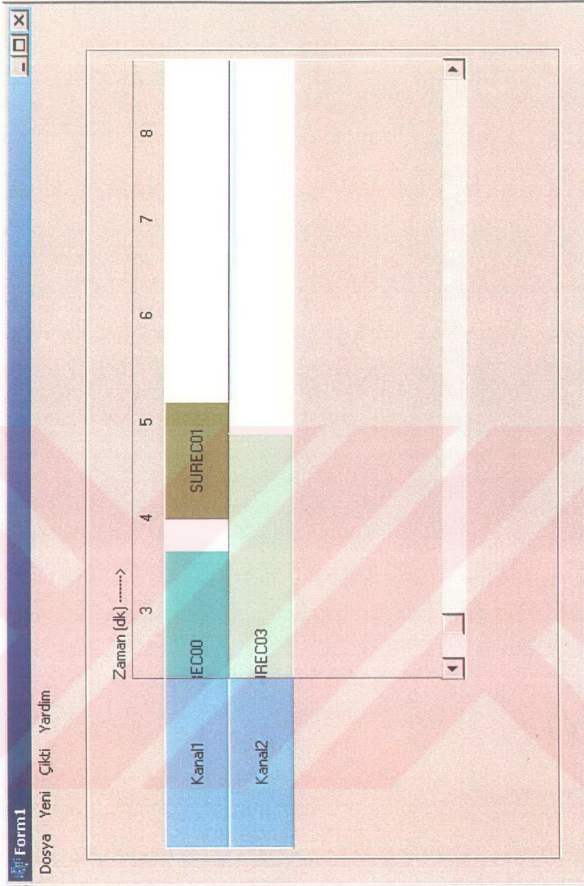
Süreç çıkış kanalı içinde hareket ettirilmesi halinde hesaplaması daha kolaydır. Çünkü sürecin sadece kanalın sol kenarına olan uzaklığı değişmektedir. Bu hesapta temel ilke fareyi devamlı surette sürecin merkezinde tutacak şekilde süreci hareket ettirmektir. (Şekil 3.39)



Şekil 3.39 Süreci taşırken

$$sL = sL + sM - sW/2$$

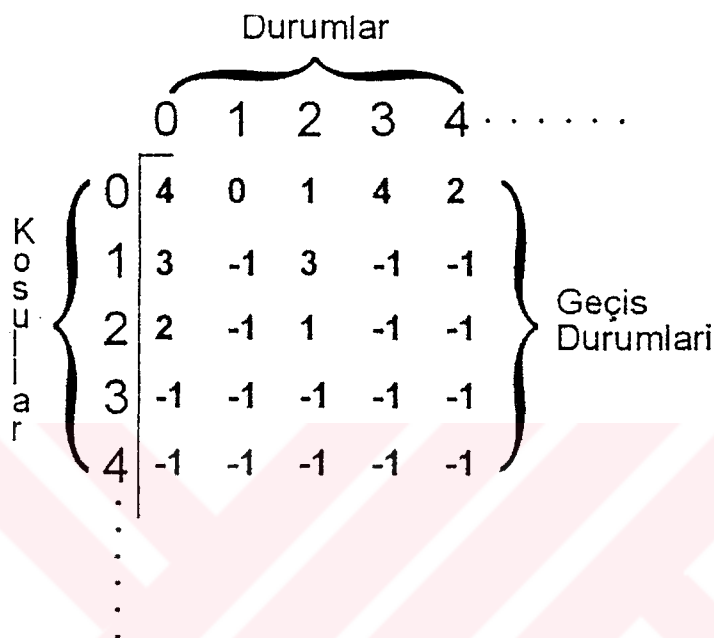
Altta ki kaydırma çubuğunun hareketiyle süreçlerin de hareket etmesi gerekmektedir. Bunun için kullanıcı kaydırma çubuğunun değerini arttırdığında her sürecin sL değerinden belli bir değer çıkarmalı, azalttığında ise her sürecin sL değerine belli bir değer eklenmelidir. Bunun sonucunda bazı süreçlerin sL değerleri sıfırın altına inecektir. Bu olay ya süreçlerin görüntüden çıkmasına yahut sadece belli bir kısmının kırılmış olarak görünmesine sebep olacaktır ki bu da Panel elemanının bize sağladığı istenen bir özelliktir. (Şekil 3.40)



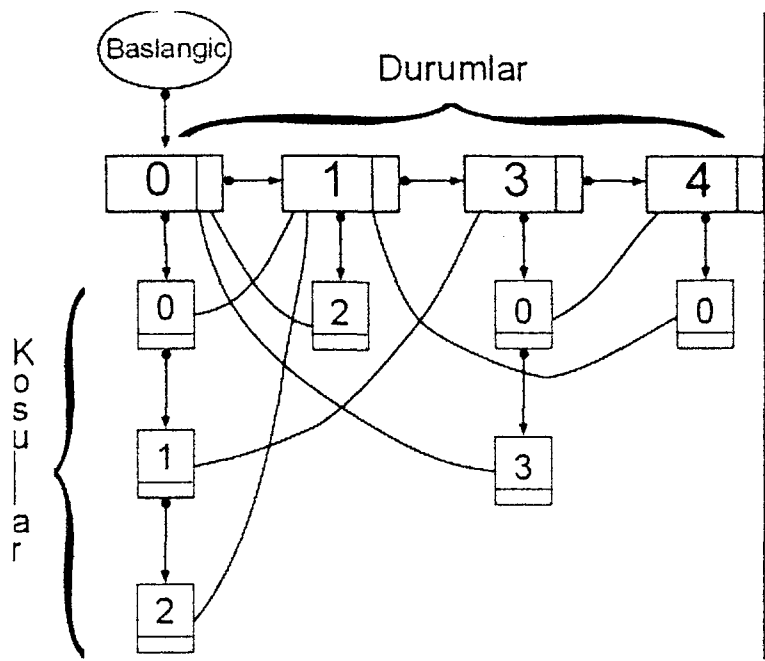
Şekil 3.40 Zaman sürgüsünün kullanımı

Süreçlerin içinde kullanılacak durum makineleri sonuç kodunda olduğu gibi statik iki boyutlu diziler şeklinde olamaz, çünkü kullanıcının kaç tane etkileşim yaratacağı ve her etkileşim için ne kadar geçiş koşulu yaratacağını önceden bilmek imkansızdır. Bu yüzden kareografi arayüzünde durum makinesi için dinamik bir veri yapısı hazırlanmıştır. (Şekil 3.41) (Şekil 3.42) Bu yapı aslında satır ve sütun boyutları kendi aralarında ve birbirlerinden bağımsız iki boyutlu bir diziyi simule etmektedir. Durum makinesi için düşünülen iki boyutlu matrisin dinamik veri yapısıyla oluşturulması işaretçiler kullanarak çözülmüştür. Buna göre

durumların hepsi bir durum bağlantılı listesinde bulunmaktadır. Her durum aynı zamanda kendi geçiş koşullarını içeren bir koşul bağlantılı listesine sahiptir, ve de koşul bağlantı listesindeki her koşul hedeflediği duruma bir işaretçi içermektedir. Böylece istendiği kadar durum ve koşulun eklenip çıkarılmasına olanak sağlanmıştır.



Şekil 3.41 Durum makinesinin iki boyutlu dinamik dizi şeklinde gösterimi



Şekil 3.42 Durum makinesi dinamik veri yapısı

Processing koduna çıktı alınırken dinamik bir yapıya ihtiyaç duyulmadığından hazırdaki durum makinesi iki boyutlu statik bir diziye çevrilerek çıkış kodundaki yerini alır. (Şekil 3.43)

		Durumlar				
		0	1	2	3	4
K o s u l a r	0	4	0	1	4	2
	1	3	-1	3	-1	-1
	2	2	-1	1	-1	-1
	3	-1	-1	-1	-1	-1
	4	-1	-1	-1	-1	-1

Geçiş Durumları

Şekil 3.43 Durum makinesinin sabit matris şeklinde gösterimi

4. SONUÇLAR

Tez süresince performans sanatlarında teknoloji ve multimedyanın kullanımı incelenmiş, dünyadaki örneklerde yazılım ve donanımın kullanımı araştırılmıştır. Yapılan araştırmalar sonucu belli tipte donanıma bağlı kalıp sabit sistemler oluşturmanın efektif olmanın çok ötesinde olduğu anlaşılmıştır. Bunun sonucunda pek çok modülü kendi bünyesi altında toplayabilecek bir çerçeve yapı önerilmiştir. Böylece sistem giriş ve çıkış donanımı kısıtından kurtulmuş, etkileşim seçenekleriyle beraber geliştirilmeye ve genişletilmeye hazır bir duruma getirilmiştir. Önerilen yapıda modülleri bir arada tutmak için yeni bir yaklaşım getirilerek durum makinesi yapısının zaman kavramıyla entegre edilmiştir. Bu yapıda sistemdeki her etkileşim zamanın kısıtları altına sokulabilecek birer durum olarak düşünülmüştür. Her etkileşim tanımlanmış çeşitli geçiş durumları vasıtasıyla birbiri arasında geçiş yapabilmekte ve belirli zaman kısıtları sonunda başka yapılara geçiş imkanı verilmektedir. Önerilen bu sistemin diğer pek çok sistemde olduğu gibi bazı performans ve donanım kısıtlamaları yaşaması olasıydı. Bu kısıtlamaların üstesinden gelebilmek için sistem bir ağ üzerinde dağıtık çalışabilecek şekilde yeniden tasarlandı. Böylece modüllerin birbirleri arasında haberleşmesi ağ üzerinden gönderilen mesajlar yardımıyla yapıldı. Projenin performans anında çalışan bu kısmı Processing ortamında Java dili kullanılarak yazıldı. Böylece Processing'in medya konusundaki kolaylıklarından ve modülerliğinden faydalanılmış oldu.

Projenin diğer yarısında hazırlanan bu yazılım sisteminin nasıl işleyeceğini kurgulayacak bir arayüz tasarlandı. Bu arayüz yardımıyla kareografin ortamı durum makinesi-zaman mantığını kullanarak tasarlaması beklenmektedir. Tasarım bittikten sonra koreograf hazırlanan sistemi Processing kodu şeklinde derleyip çıktı alabilmektedir. Arayüzün bir başka işleviye hazırlanan kodun ağ üzerindeki bilgisayarlara dağıtılmasını sağlamaktır. Bu işlem için kareografi arayüzü dışında ağdaki her bilgisayarda çalışacak ve kareografi arayüzünün emirlerini yerine getirip dosya alışverişini sağlayacak bir ajan programı tasarlanmıştır. Arayüz Borland C++ Builder 6.0 ortamında geliştirilmiştir. Arayüzde Borland'ın standart elemanlarının sunduğu özellikler sistemde kullanılacak yapılara yetmediği için yeni elemanlar standart elemanların özellikleri kullanılarak geliştirilmiştir.

Sonuç olarak bu tez çalışması bir mühendisin bir sanat dalında teknolojinin nasıl kullanıldığını çözümlediği ve sayısal ortamda bu sanat dalında kullanılabilecek bir sistem ve arayüz tasarladığı disiplinlerarası bir çalışmadır. Hazırlanan sistem ve arayüzün tam anlamıyla kullanıma geçirilebilmesi için bir ekiple ciddi bir yazılım geliştirme safhasına ihtiyacı olmasına rağmen yapılan bu çalışma sistemin uygulanabilir olduğunu kanıtlamaktadır.

KAYNAKLAR

- Camurri, A., Hashimoto, S., Ricchetti, M., Ricci, A., Suzuki, K., Trocca, R., Volpe, G., (2000), "EyesWeb - Toward Gesture and Affect Recognition in Dance/Music Interactive Systems", *Computer Music Journal*, 24(10): 57-69.
- Dowling, P., Wechsler, R., Weiss, F., (2004), "EyeCon -- a motion sensing tool for creating interactive dance, music and video projections", *Motion, Emotion and Cognition at University of Leeds*, Leeds, İngiltere.
- Griffith, N. ve Fernstör, M., (1998), "LiteFoot – A floor space for recording dance and controlling media", *ICMC98*, Michigan, A.B.D.
- Holmes, Douglas B., (2003) "Jn4.Gesture: An Interactive Composition For Dance", *University of North Texas*, Texas, A.B.D.
- Mohnsen, B., (2005), "Dance Dance Revolution: The Next Big Technology Device.", *Teaching Elementary Physical Education*, 16(2): 36.
- Mulder, A., (1994a), "How to Build an Instrumented Glove Based on the PowerGlove Flex Sensors", *PCVR Magazine*, 16,10-14.
- Mulder, A., (1994b), "Human Movement Tracking Technology", *Hand centered studies of human movement project*, Teknik rapor 94-1, Burnaby, Kanada.
- Paradiso, J., Abler, C., Hsiao, K., Reynolds, M., (1997), "The Magic Carpet: Physical Sensing for Immersive Environments", *SIG CHI 97*, Atlanta, A.B.D.
- Paradiso, J. A., Hsiao, K., Benbasat, A. Y., Teegarden, Z.. (2000), "Design and implementation of expressive footwear", *IBM Systems Journal*, 39(3), 511-530.
- Richardson, B., Leydon, K., Fernstör, M., Paradiso, J., (2003), "Z-Tiles: Building Blocks for Modular, Pressure-Sensing Floorspaces", *CCCT2003*, Orlando, A.B.D.
- Rowan, J. B., Wechsler, R., Weiss, F., (2001), "Artistic Collaboration in an Interactive Dance and Music Performance Environment: Seine Hohle Form, a Project Report", *COSIGN2001*, Amsterdam, Hollanda.
- Siegel, W., Jacobsen, J., (1998), "The Challenges Of Interactive Dance: An Overview And Case Study", *Computer Music Journal*, 22(4): 29-44.
- Theobalt, C., Magnor, M., Schüler, P., Seidel, H., (2004), "Combining 2d Feature Tracking And Volume Reconstruction For Online Video-Based Human Motion Capture", *International Journal of Image & Graphics*, 4(4): 563-584.
- Wright, M., Freed, A., Momeni, A., (2003), "Open Sound Control: State of the Art 2003", *NIME-03*, Montreal, Canada.
- Warren, Jonah, (2003), "Unencumbered Full Body Interaction In Video Games", *Parsons School of Design*, New York, A.B.D.
- Zaks-Zilberman, M., (1999), "Merce Cunningham: Dance and Technology", *American University*, Washington D.C., A.B.D.

İNTERNET KAYNAKLARI

Eric Singer, “Cyclops”, < <http://www.ericssinger.com/cyclopsmax.html> > (Erişim tarihi: 14.04.2005)

EyeCon, “Video Motion Sensing”, < <http://www.frieder-weiss.de/eyecon/equipment.html> > (Erişim tarihi: 15.04.2005)

IGN.COM, “Hey You, Pikachu”, < <http://ign64.ign.com/articles/153/153734p1.html> > (Erişim tarihi: 02.05.2005)

InfoMus Lab, “DanceWeb”, < <http://www.infomus.dist.unige.it/Tools/DanceWeb.html> > (Erişim Tarihi: 04.04.2005).

MIT Media Lab, “Processing”, < <http://www.processing.org/> > (Erişim tarihi: 05.05.2005)

STEIM, “BigEye”, < <http://www.steim.org/steim/bigeye.html> > (Erişim tarihi: 14.04.2005)

Troikaranch, “Interactor LPT”, < <http://www.troikaranch.org/interactor.html> > (Erişim tarihi: 16.04.2005)

Troikaranch, “MidiDancer”, < <http://www.troikaranch.org/mididancer.html> > (Erişim tarihi: 17.04.2005)

Troikatronix, “Isadora”, < <http://www.troikatronix.com/isadora.html> > (Erişim tarihi: 16.04.2005)

Unsafe&Sound, “e-Merge”, < <http://www.unsafesound.com/e-merge1.html> > (Erişim tarihi: 15.04.2005)

ÖZGEÇMİŞ

Doğum Tarihi 10.08.1980

Doğum yeri İstanbul

Lise 1992-1998 Adile Mermerci Anadolu Lisesi

Lisans 1998-2002 İ.T.Ü. Elektrik Elektronik Fakültesi
Bilgisayar Mühendisliği Bölümü

Çalıştığı kurumlar

2000-2002 İ.T.Ü. Bilgi İşlem Merkezi, Öğrenci Asistan
2002-Devam ediyor Y.T.Ü. Enformatik Bölümü, Okutman

