

## Bilgisayarla Öğretim Programı (BÖP) ve Materyal Tasarımı: Akış Şemaları ve Stratejiler

### Giriş

Bilgisayarla Öğretim Programlarının (BÖP) tasarımı ve geliştirilmesi sürecinde izlenen basamaklar, her hangi bir öğretim materyalinin tasarımı ve geliştirilmesinde izlenen stratejiler ile benzerlik gösterir. Bunun böyle olması doğaldır. Çünkü en basit bir örnekle durumu tartışmak istersek kaliteli saydam hazırlama veya sayfa tasarımını ortaya koyabiliriz. Burada hazırlanan öğretim materyalinin kim için, hangi düzeydeki bireyler için geliştirileceği, burada kullanılan teknolojinin düzeyi, kullanılan şekiller, görsel ifadelerin canlandırılması, ekranın tasarımı ve şekli, verilmek istenen kavram ya da kavramlar, anahtar kelimeler, okunabilirlik düzeyi ve karmaşık görüşlerin nasıl açık hale getirilebileceği öncelikle ortaya konulur. Yani bir planlama yapılır. Tasarım sonrasında ise görsel materyal olarak bunun bir değerlendirilmesi yapılır. Değerlendirilme süreci sonrasında ilgili birimler için önerilerde bulunulur. Bu nedenle öğretim teknolojileri yardımcıyla diğer öğretim materyalleri tasarımı ve geliştirilmesinde izlenen teknikler birbirlerinin tamamlayıcısı durumunda olurlar. Bu öğretim araç ve materyallerinden biriside kaliteli Bilgisayarla Öğretim Programı (BÖP) olup, bu programın tasarımı ve geliştirilmesi çok önemli bir öğretim süreci etkinliğidir.

Bu çalışmanın amacı Bilgisayarla Öğretim Programı (BÖP) tasarımı ve öğretim materyallerinin geliştirilmesinde akış şemalarının (çizeneği) işlevlerini Öğretim Tasarımı ve Sistemleri bakımından tartışmaktır. Bu nedenle izlenen yol herhangi bir öğretim materyali tasarımı ve geliştirilmesinde izlenecek stratejileri ortaya koyarak bilimsel temeller etrafında akış şemaları ve ilgili diğer stratejilerinin işlevlerini öğretim süreçleri içindeki değeri yönünden örneklerle açıklamak ve sunmaktır. Bunun için yapılması zorunlu etkinlikler izleyen paragraflarda tartışılmaktadır.

Bilgisayarla Öğretim Programlarının (BÖP) Öğretim Tasarımı (ÖT) sürecini işlevsel kılarak daha görsel ve uygulanabilir duruma getirici çalışmalarından birisi akış şemaları ile sürecin tamamlanmasıdır. Bu süreç içinde programın yapısını ve öğretimin işleyişini görebilmek olanaklıdır. Bu kolaylığı sağlamak için, bilgisayar kullanıcıları ve program tasarımcıları önce akış şemaları (flowchart) geliştirerek ne yapacaklarına ilişkin bir plana sahip olurlar. Sonra bu planı izleyerek yeni aktiviteler ile ayrıntıları anlaşılır hale (storyboard-öykü panosu) getirirler ve şema içine yerleştirirler. Daha sonra da, derslerin ya da programın ilerleyişi ve akışına yardımcı olacak, yazılı metinlerden oluşan açıklamalar (scripting) kümesi oluşturularak, tasarım araçları BÖP tasarım ve geliştirme alanına ve sürecine girilmiş olunur (İpek, 2001).

ÖT sürecinde kullanılan akış şemasının içinde yer alan her şekil ve çizim biçimi program tasarımı içinde değişik rol ve özel anlama sahip bulunmaktadır. Akış şeması içindeki ilerleyiş bazen bir yönde, yani doğrusal "linear" ya da dal ve kollara ayrılarak "branching" devam edebilir. Bazı durumlarda bu iki yaklaşım aynı süreçte birlikte kullanılabilir. Bu akış şemalarının temel amacı, program için kısımları, seçenekleri ve bunlar arasındaki ilişkileri gösterme amacı taşır. Bu ilişkileri gösterme aynı amaca uygun olarak birlikte kullanıma hazır olabilmeyi olanaklı kılarlar.

Akış şemasının tasarımı basit şekillerin ifadesi ile başlar. Bu basit görünen ve gösterilen öğeler programın hedeflerini, ilgili soruları, önemli hususları ve dalları veya dallanmayı gösterir. Akış şemalarının konusu ve yapısı durağan değildir. Akış şemaları, tasarımcı ve programcıya yön verir. Kolaylık sağlayarak dükümanların sunuluşunu ve hareketlerini görsel olarak olanaklı kılarlar (bakınız şekil 1).

Program tasarımı esnasında, bilgisayar programcısı ve yazılımcıların, öğretim tasarımı akış şeması ile yazılım akış şemasını birbirlerine karıştırmamaları yararlı olur. Her iki uzman grup, akış şemalarını planlama aracı olarak kullanmakla beraber, bunlar birinden çok farklılık gösterirler. Programlama yapan (yazılımcı) kişi, her şeyi ayrıntılara döker. Gösterilen her bilgisayar öğretimi ve süreci kullanılan semboller ile çok özel anlamlar içermektedir. Buna karşın ÖT'nin geliştirdiği akış şeması çok fazla sembol içermez. Bilgisayarın kullanımı konusunda çok az bilgiye sahiptir. Bu nedenle akış şeması programın güvenliği (code) için yararlıdır. Bu olanak programcıya programa girmek için ne çeşit güvenlik sistemine (code) gereksinim olduğunu iletir. Yazılımcı programın yapısının bir haritasını tasarlayarak program içinde dalları, önemli noktaları, dönüşümleri, grafik ve ses gibi diğer program özelliklerini ortaya koyar.

Oysa programın tümüne bakıldığı zaman yazılımcının bazı hünerleri orada görülmekle beraber daha çok öğretim tasarımcısının izlediği yolları ve teknikleri görebiliriz. Zaten ÖT planlayıcısı bir proje lideri ve tasarımcısı olduğuna göre farklı etkinlikleri koordine ederek yüksek kalite ve verimlilikte bir ürünün proje takımı tarafından ortaya çıkarılması için uğraş verir. Ya da aynı zaman ve süreç içinde birden fazla öğretim tasarımcısının bulunması gerekebilir veya buna gereksinim duyulabilir.

### Akış Şemasının Çeşitleri ve Düzeyleri

Akış şemalarının oluşturulması, Alessi ve Trollip (1991) ve İpek (2001) tarafından içeriklerinin yoğunluklarına göre üç ayrı düzeyde tanımlanmaktadır. Bu farklı düzeyler basitten karmaşığa doğru bir yönde geliştirilir. Ayrıntıları fazla olan bir akış şemasının tasarımı oldukça zordur. Bu nedenle gereksiz sayılabilecek bazı ayrıntıları azaltarak uygun formlar hazırlanmalıdır. Akış şemaları günümüzde yapılan her iş ve görev için yapılacak analizlerde yararlı planlama araçlarıdır. Derslerin planlanmasında, sadece başlangıçtan dersin sonuna kadar olan sıralamayı göstermez. Aynı zamanda süreç içindeki olası kararları da ifade etmelidir. Bunun yanında akış şemaları bilgisayar programları için oluşturulan diyagramlar için tek bir teknik değildir. Bu konuda yapılan tartışmalar da vardır. Yani akış şemalarının program geliştirmeye ne kadar katkıda bulunduğuna ilişkin bilgi yetersizliği vardır. Bu husus küçümsenmemelidir.

Tüm bunlara karşın, akış şemalarının yapılan çalışmalar ve geliştirilen BÖP uygulamalarındaki önemi, akış şeması geliştirmenin, BÖP (software) tasarımıyla yol gösterici ve eksiklikleri tamamlayıcı rolü yakından görülmüştür. Tasarım süreci içinde araştırmacı ve tasarımcının yapılacak işler içinde kaybolmadan ve zaman israf etmeden tüm program tasarımı sürecini kontrolü altında tutması bakımından çok yararlı hatta zorunlu bulunmaktadır (İpek, 1995, 2001). Bu nedenle tüm araştırmacı ve tasarımcılara, öğretmenlere, mühendislere, öğrencilere ve eğitim uzmanlarına böyle bir program tasarımına girişmeden önce kesinlikle hangi düzeyde olursa olsun bir akış şeması ve diyagramı geliştirmeyi önerebiliriz. Bu açıklamalardan sonra akış şemalarının çeşitlerini ve özelliklerini öğretim sürecindeki önemi bakımından şöyle açıklayabiliriz.

Akış şemalarını oluşturma güçlükleri diğer BÖP'ler bakımından da farklılık göstermektedir. Örneğin, basit bir ders ya da konunun bir Özel Öğretici Program (ÖÖP) "tutorial" olarak tasarımını yapıyorsak, izlenecek adım *ilk düzey-basamak* olacaktır. Bu durum şöyle ifade edilebilir. Örneğin arabanızı sürerken bir lastiğinin patlaması sonrasında izlenecek işlemlerin sırası ve bazı tekrarlar söz konusudur. Böyle bir problemin çözümü için yapılacak işler için *ilk ve ikinci düzey* basamağında bir akış şeması geliştirilmelidir. Yani konunun içeriği ve karmaşıklık düzeyine göre akış şemasının zorluk seviyesine ve ne çeşit olacağına karar verilir.

Kısaca belirtmek gerekirse, akış şemalarının çeşitlerini belirlemek *görevi tasarımcıya ve takım üyelerine* düşen bir görevdir. Basit bir konu için genel gözden geçirme, ders işleme sırası ve tekniği sadece ekrandaki sayfa üzerinde gösterilebilir. Bu aşama *ilk (düzey-1)*

basamaktır. Eğer dersin konusu ya da yapılacak iş biraz daha karmaşık ise, süreç içinde karar verme noktalarının ve bazı ayrıntıların kol ya da dallara ayrılarak (branches) gösterilmesi söz konusu olur. Bu tasarım düzeyi ise ikinci düzey (*düzey-2*) olarak kabul edilmelidir. Yapılması gereken iş veya yapılacak görevlerin analizleri bir olay ya da konu için söz konusu ise, bu aşamada uygun öğrenmelerin ve işlemlerin gerçekleşmesi zorunluğu doğar. Bu düzeyde akış şeması sürecinde her türlü öğretim araçlarını öykü panosu (storyboard-senaryo, script), bilgi edinme kaynaklarını, hesaplamaları, ayrıntılara veya dallara ayırarak (branching), bilginin yönetimi (information management), kullanıcı kontrolünü (user kontrol) yeterli düzeyde ayrıntıları ile sürecin içine koymak ve sunmak gerekir. Bu seviyedeki ilişkileri yansıtan ve yönlendiren gösteren akış şeması *üçüncü düzey (düzey-3)* olarak tanımlanır ve kullanılır (Alessi ve Trollip, 1991; İpek, 1995, 2001; Luppá, 1984).

BÖP geliştirme deneyim ve çalışmalarına bakarak belirtmek gerekirse, yukarıdaki cümlelerde sözü edilen *düzey-3* şeması her BÖP için gerekli olmayabilir. Buradaki değerlendirme, tasarımcının konunun karmaşık oluşuna ve ne kadar ayrıntının tasarım için yeterli sayılacağına ilişkin görüşüne göre biçimlenir. Bazı durumlarda bilgisayarlarla kullanılan hazır sayılabilecek akış şemaları örnekleri de vardır. Bunlara ek olarak kısaca hangi BÖP'nin hangi düzeyde tasarlanmış bir akış şeması ile tasarımının gerektiğine ilişkin düşüncelerimizi şöyle belirtebiliriz (bakınız örnek şekiller 2, 3 ve 4).

- Özet olarak belirtildiği üzere, karmaşık olmayan dersler için *düzey-1* seviyesindeki akış şeması rahatlıkla ve kolayca geliştirilip kullanılabilir.
- Alıştırma, deneme ve test etkinliğine yönelik ders etkinliklerinin fazlaca karmaşık olabileceği düşünüldürse, fazlaca işlem ve görev gerektirdiği görülür. Bu nedenle *düzey-2* seviyesinde akış şeması geliştirilmesi önerilir.
- Simülasyon ve öğretim oyunları tasarımı oldukça karmaşık ve deneyim gerektiren bir çalışmadır. Bu düzeye çok dikkat etmek gerekir.

Burada fiziksel ve sosyal faktörlerin etkileşimi önem taşır. Bunun için simülasyonların *düzey-3* seviyesindeki akış şeması ile geliştirilmesi tasarımı daha verimli ve kolay kılabilir. Öğretimsel oyunların basamakları farklı içerik taşıdığı için, yani öğrencilerin öğrenme seviyelerine göre alıştırma (drill) tasarımı yapılır. Bu etkinlik için *düzey-2* seviyesinde akış şeması tasarımı ve diyagramı uygun görünmektedir. Fakat öğretim oyunları aynı zamanda simülasyon nitelikleri ile kullanılıp, program tasarımı gerektiriyorsa, *düzey-3* seviyesindeki bir akış şeması ile yapılacak işlemlerin gösterimi daha anlamlı olabilir. Yani daha iyi yön ve işlem yollarını gösterme işlevine sahip olur.

#### **Akış şemaları tasarımında kullanılan semboller ve nitelikleri**

Akış şemalarının düzeylerine girmeden önce, akış şeması için kullanılan basit sembolleri (icons) ve şekilleri, hem akış şeması hem de Öğretim Tasarımı (ÖT) uygulamasında kullanılmaları bakımından özellikleri ile okuyuculara anımsatmak gerekir. Her sembolün akış şeması veya şekillerin tasarımı süreci içinde özel bir rolü ve anlamı bulunmaktadır (bakınız şekil 1).

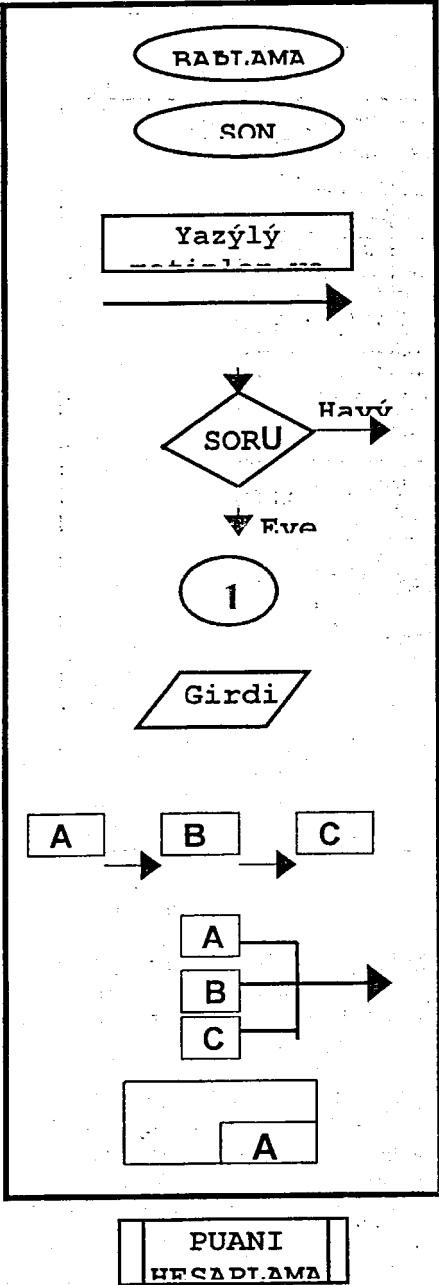
#### **Akış şemasını tasarlama yolları ve adımları**

Akış şemasını düzenlemeden önce, onun genel olarak ileriye doğru ilerleyen bir işlem olduğunu düşünmeliyiz. Daha sonra konu uzmanı (*Subject-MatterExpert*) bireyin o işlemi yaparken ne yaptığını ve düşündüğünü açıklar. Bu işlemin doğruluğu için süreç ve karar verme ile akış şeması geliştirilir. Bu işlemler ayrıca gözden geçirilmelidir. Bu işlemlere geçmeden önce yapılacak görevlerin bir listesi hazırlanır. Burada genel olarak *akış şeması* geliştirmede üç ayrı basamaktan söz edebiliriz (Seels ve Glasgow, 1998). Özetle belirtmek gerekirse;

1. Eđer elimizde yapılması gereken 10 sayıdan fazla görev ve iş varsa onların sayılarını 5-7 arasında sınırlayıp, onları öncelik sırasına göre düzenleyelim. Bu işleri akış şemasında tasarlayarak rahatça belirtebiliriz.
2. Başlama noktasından itibaren yapılması planlanan temel iş ve görevlerin (major tasks), alt düzeyindeki işlerin sırasını belirttikten sonra gerekli ise kollara-dallara (branches) ayırarak göstermek gereklidir.
3. Konu uzmanı (SME) ile akış şemasının gerçekleştirebileceđi işlemler açıklanarak görürlür. Bu aynı zamanda ilk iki basamağın birleşimidir. Uzman kişi ile beraberce akış şeması tamamlanmamış ise buradaki işlemlerin ve işlerin sıralanmasının doğruluğundan emin olmalıyız. Diđer bir durum ise tüm iş ve görevlerin uygun düzeydeki ayrıntıları ile gösterildiđine bakılmalıdır (Seels ve Glasgow, 1998).

#### **Akış şemalarının düzeyleri ve bazı örnekler**

Akış şemaları geleneksel olarak doğrusal yaklaşımla düzenlenmiş olmalarına karşın duyulan gereksinim ve teknolojik gelişmeler sonucu, BÖP tasarımı ve geliştirilmesinde farklı yaklaşımların kullanılmasına gerek olmuştur. Genel olarak akış şemaları doğrusal "linear" ve dallara-kollara ayrılma "branching" biçiminde tanımlanmıştır. Böyle olmakla beraber bu iki yaklaşımın özelliklerinin beraberce kullanıldığı ve geliştirildiđi birleştirilmiş akış şemaları da çok yaygın olarak kullanılmaya başlanmıştır. (Dills ve Ramiszowski, 1997; Gentry, 1994; Gibbons ve Fairweather, 1998; İpek, 2001; Jonassen, 1985, 1988, 1996; Merrill, 1994; Price, 1991; Seels ve Glasgow, 1998).



Programın veya dersin başlaması

Program veya dersin bitişi ise birden fazla yerde olabilir.

Bilgiler girilir, yanıtlar verilir. Ne yapılacağı belirtilir (süreç).

Programın gidiş/akış yönünü gösterir.

Elmas işareti, programın akış sürecinde

Akış şemasında ve tasarımda işlemler arasında veya sayfalar arasında bağlantıyı sağlar (sıralama)

Bu şekil öğretim tasarımı için (Input/ Output) bilgi girişi/bilgi çıkışı (ürün) için kullanılır.

Basamaklar arasında sıralanmayı gösterir.

Akış şeması, A, B, C işlemlerinin herhangi sıra düzeni içinde D işlemine girişmeden yapılması gerektiğini işaret eder.

Bu işi açıklamak için (not) konulur. Sayfa içinde alt köşeye numaralanarak belirtilir. Amacı bu bilginin daha önce verildiğini göstermektir.

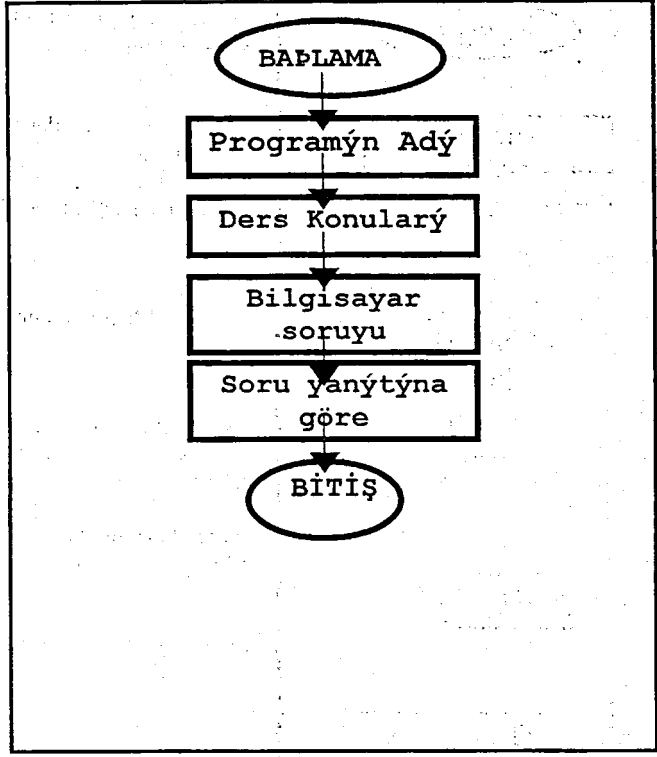
Bir işlemin alt işlemini ve işini ifade eder. Özel ifadeler halinde gösterilir.

**PUANI**  
**HEFADTAMA**

Şekil 1 Akış şemalarında kullanılan semboller (İpek, 2001)

Karmaşık olan ve üçüncü düzey akış şeması geliştirmede, program bilgilerini toplama ve nereye konulacağına ilişkin bilgiler yer alır. Diğer önemli bir nokta ise yeni alt basamakların oluşturulmasıdır. Bunlar dersi izlerken görsel olmayan, yani görünmeyen hususları içerir. Bu nedenle bilgisayar programları dilleri ile yeni bir program üretmeye yarayan daha çok sözel ifadelerle geliştirilen (authoring languages) programlar alt basamakları (subroutine)

geliştirebilme olanağına sahip olurlar. Fakat bunun yanında akış şemasının yetersiz ve yanlış düzenlenmesi nedeni ile bazı hatalar oluşabilir.



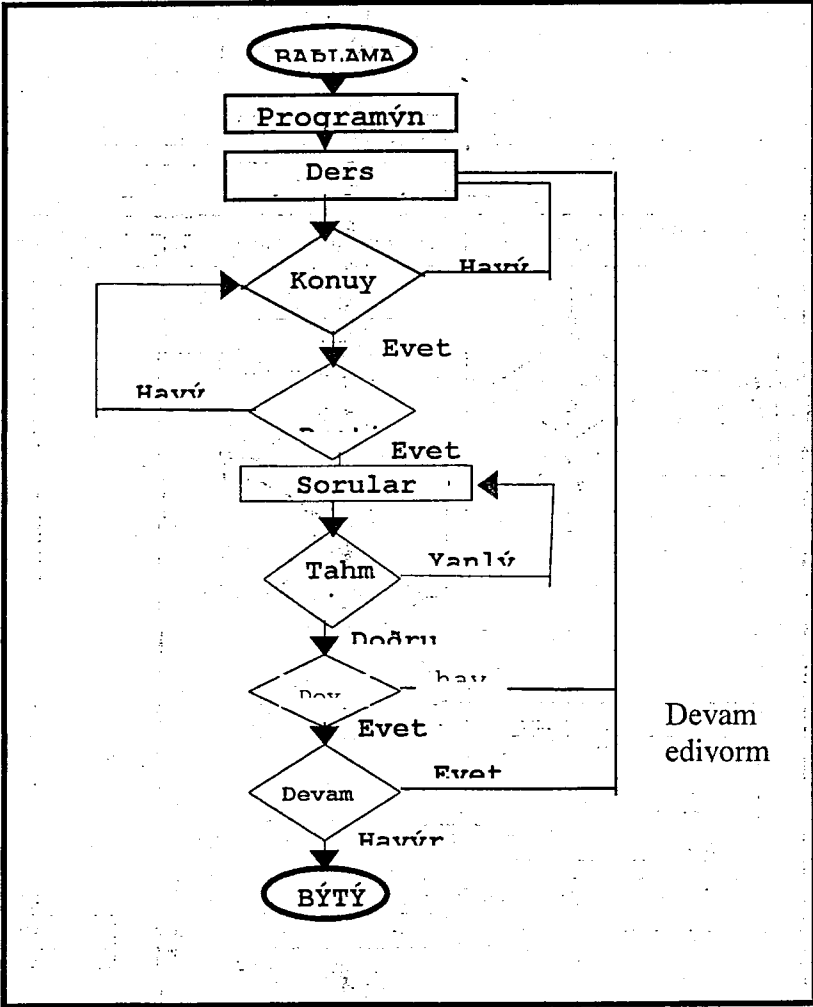
Şekil 2 BÖP Tasarımı ve geliştirmede ilk düzey için akış şeması süreci

Bu hataların oluşmasındaki en büyük eksiklik, akış şemasının dersle ne yapılacağını iyi belirtmemiş olmasıdır. Yani programı yapan kişiye yanlış veya hatalı bir akış şeması verilir ise, bu boş yere uğraşmaktır. Burada genellikle iki türlü hatadan söz edebiliriz. Birisi yazıyla ilişkin (typographical) hatalar, diğeri ise mantıksal hatalardır.

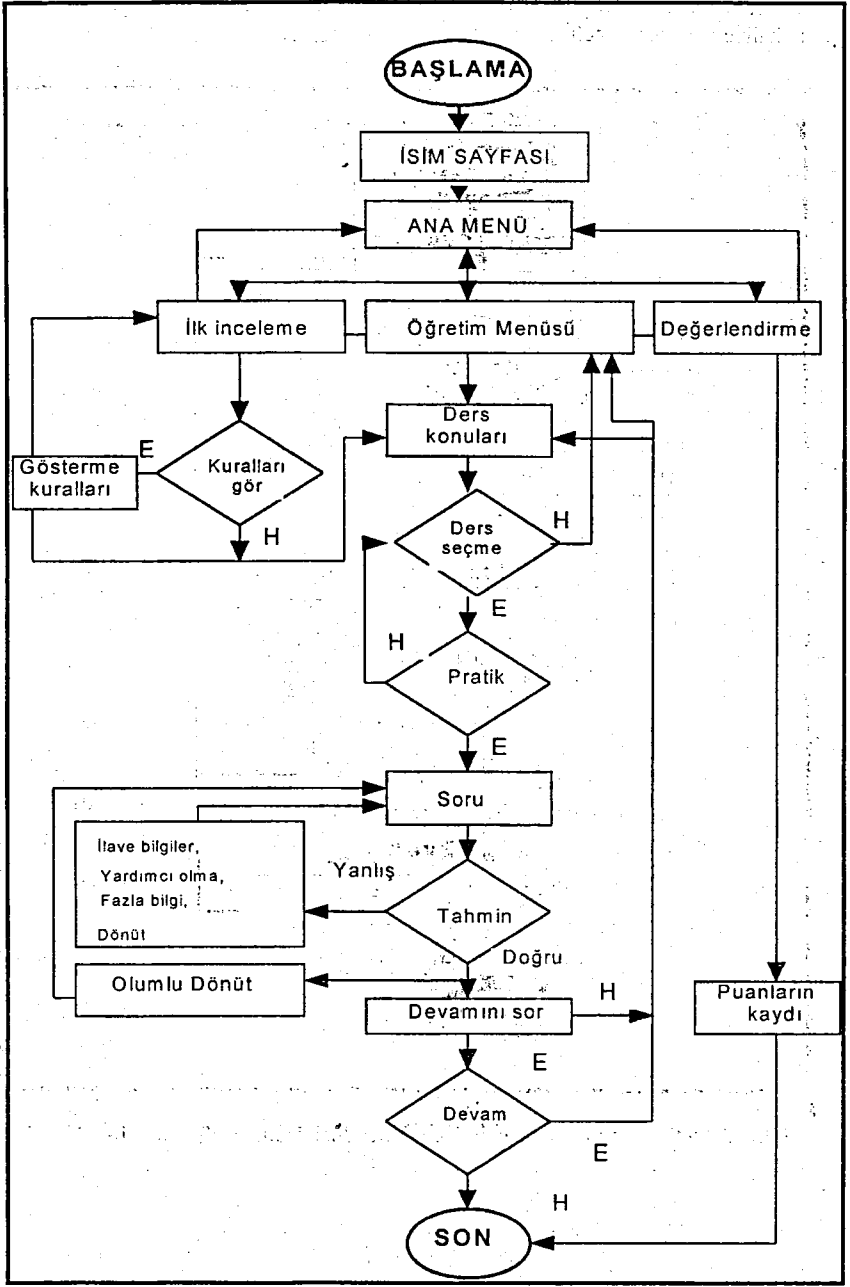
BÖP'nin akış şemasının (dry-running) bir kişinin bilgisayar gibi çalışarak, tüm öğretim sıralamasını izleyerek çalışması programda test edilmeyen noktayı ortaya koyar. Bunun için (dry-run) bütün öğretim işlevlerini tekrar tekrar kırmızı mürekkeple göstererek defalarca izlemek gerekir. Bu süreç sonucunda her sorudaki yanıtlamanın ve dönütlerin doğru yanlış olması ile ilgilenilmez. Sadece verilen doğru dönütlerle, bizim gelecekteki olası işlemleri uygun olarak gözden geçirmemize yol açabilir.

Diğer bir hata şekli ise, akış şemasının mantıksal işlemleri içermesidir. Buradaki hataların nedeni öğrencilere verilen yönergelerin karmaşık olarak anlaşılması veya hiç bir şey vermemesidir. Bu gün çok iyi akış şemaları için programlar üretilmiştir. Bunlardan bazıları, "Macflow" ve "EasyFlow" isimli programlar olup güzel örneklerdir. Akış şemaları nitelikleri öğretim programı tasarımı (ÖPT) için önemli bir bölümdür. Dersin grafiklerle ilgili kısmını yansıtır. Olaylar belirten (oyun tahtası veya öykü panosu) materyal ile kullanıldığı zaman dersin içeriğini, dolayısıyla programın konusunu ve işleniş sırasını çok iyi açıklayabilir. Bu

durum dersin ve tüm programın adaptasyonu yani uyarlanması ile etkileşim gücü bulan dersler haline dönüşür ve gelişir.



Şekil 3 BÖP Tasarımı ve geliřtirmede ikinci düzey için akıř řeması (İpek, 2001)



Şekil 4 BÖP Tasarımı ve geliştirmede üçüncü düzey için akış şeması (İpek, 1995)

#### Akış şeması niçin önemlidir?

Akış şemasının etkili ve görsel olarak ÖT süreci içinde kullanımı, BÖP tasarımı ve geliştirilmesi için çok yararlı olmasının yanında, asıl görevi olan derslerin öğretim tasarımının tüm uygulanabilecek ayrıntıları ile yansıtılmasıdır. Bunlar hedeflerin analizi yani gereksinim

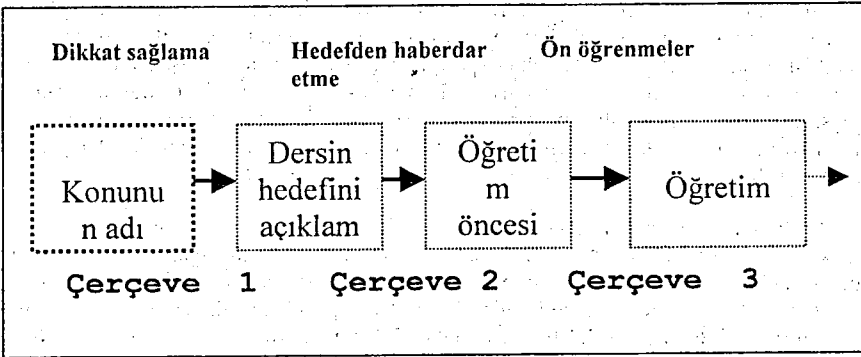


analizi sonunda tüm problem analizinin yapılması, yani problem nedir, Bu problemi nasıl çözeriz' gibi sorulara yanıt arar. İlave olarak iş ve görev analizlerinin yapılarak işin ve konunun ne olduğu, öğrencilerin neyi öğrenmiş olacaklarını açıklamaya yönelik öğretim analizi yapılarak öğretim tasarımı geliştirilmelidir. Bu süreci uygulama ve değerlendirme basamakları izleyebilir (Seels ve Glasgow, 1990, 1998). Tüm bunların ötesinde akış şemaları, öğretim programlarındaki hataların giderilmesi için harcanan süreyi kısaltmaya yardımcı olur.

Akış şemalarının genel olarak 3 ayrı düzeyde ve farklı amaçlarla geliştirildiğine değindikten sonra, akış şemalarının taşıdıkları nitelikler bakımından çeşitlerine değinmek gerekir. Bunlar, öğretim tasarımı için bir olay ve durumu görsel duruma getirebilmek için oldukça yararlıdır. Bu nedenle, öğretim programı tasarımı içinde bir örnek olay-senaryo, storyboard, ve script (öykü panosu) birlikte ele alınarak, tasarım sürecinde kullanılabilir. Bu planlama araçları akış şemasında gösterilerek, program hatalarının en aza indirilmesi sağlanmış olabilecektir.

### Akış Şemalarından Bazı Örnekler: Doğrusal ve dallara ayrılma yaklaşımları

BÖP tasarımında tarihsel bir görüşle bu süreç yaklaşarsak, davranışsal hedeflerin gerçekleştirilmesine dayalı programlı öğretim (PÖ) dediğimiz "Programmed Instruction" yaklaşımına ulaşırız. Bu yaklaşım Skinner ile özdeşleşmiş ve anlam kazanmıştır. Buradaki yaklaşıma uygun olarak, doğrusal programa şöyle bir örnek sunabiliriz.



Şekil 5 Doğrusal programlama ve öğretim süreci (İpek, 2001)

Buradaki şekil 5 yalın olarak bir doğrusal programın akışını gösterir. Şekil 5 de görüldüğü üzere işlemler sırası doğrusaldır. Bu süreçteki çerçeveler içinde sorular ve yanıtlar yer alır. Sonra diğer çerçeveye (konuya) girilir ve öğretime devam edilir.

Program tasarımı yaklaşımlarından biri doğrusal yaklaşım olup, tek bir öğretim sırasının bütün öğrenciler tarafından izlenmesi ve gösterilmesi esasına dayalıdır. Bu konulara ilişkin etkileşim ve adaptasyon sistemleri gelecek paragraflarda konu edilecektir. Kabul etmemiz gereken önemli bir konu, ÖT (Instructional Design) sürecinin sadece bilgisayar programı tasarımını gerçekleştirmek için daha kritik ve önemli olduğunu varsaymaktır.

Diğer bir program yaklaşımı kısımların, paragrafın, basit cümlelerin, çoktan seçmeli seçeneklerin sunulması, gereken yardımın verilmesinin ya da yetersiz bilgilerin meydana gelmesi durumunda ek bilgilerin sunulması (remediation) durumudur. Bu sistem karmaşık olmakla beraber bazı yararlı yanları vardır. Aynı zamanda BÖP tasarımı ve uygulamalarında oluşan hata ve tuzaklardan bahsetmek gerekir. Tasarımcılar bu hatalardan oluşturacakları hata

önleme mekanizması ile nasıl uzak durabilirler? Bu soru oldukça önemli olmaya devam etmektedir. Burada izlenebilecek yol ve teknik, oluşturulacak çoktan seçmeli seçeneklerin her madde için yanlış yanıtlar içereceği varsayımından hareketle, aynı zamanda yeni ek bilgiler sağlamak, "remediation" yöntemi ile programdaki hataları önleme olanağı sağlanır (İpek, 2001).

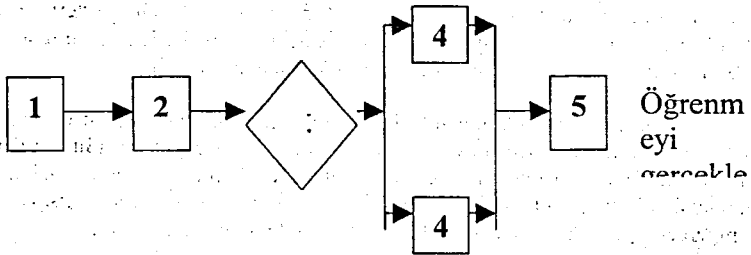
Dallara ya da kollara ayırarak (branching) akış şemasının tasarımı aynı isimle anılan programlama dilinden kaynaklanır. BÖP tasarımı ve oluşturulmasında dal-kollara ayırma sistemine dayanan yaklaşım çok etkili ve değişkenlik sağlama bakımından etkileşimi ve adaptasyonu kolaylaştırmaktadır. Bu konuda BÖP ve dersinin adaptasyonunu sağlamada izlenecek yollardan birisi de dallara-kollara ayırma (branching) yöntemidir. Bu yöntem hem yazılım tasarımı hem de öğretim tasarımında kullanılır (Jonassen, 1988).

Bu yöntemde ağaç yapısı (tree structure) ve bir probleme dayalı (case) olarak sayısal kararlara tek bir seçim yolu ile ulaşarak olası seçenekleri karşılaştırmaktır. Ağaç şeması ve yapısı ile oluşacak kararların sırasını temsil eder. Öğrencilerin bilişsel öğrenme biçimleri bağımlı-bağımsız ise (dependent-independent), bu gibi öğrenme durumuna karşıt olabilecek durumlar için de uygulanabilmelidir.

### Dallanma Programı Teknikleri

Dal-kollara (branching) ayırma yaklaşımı ile öğretim programı geliştirmede, başka tekniklerden de söz edebiliriz. Bu teknikler Price (1991) tarafından doğrusal olarak dallara-kollara ayırma (linear branching), tek olarak sağlanan ek yardım (single remediation), çok miktarda sağlanan bilgiler (multiple remediation) ve kullanıcının yönettiği (user-directed), gözden geçirme (review) dallara ve kollara ayırma teknikleri ile tasarım olarak tanımlanmaktadır. Çünkü bu teknikler, program içinde bir yerden başka bir yere hareketi işaret eden ve gösteren birer tekniktir. Bunlar ek olarak, yeniden durumu açıklama, yerleştirme ve yardım (restatement-remediation), yeniden dönüşüm (recycle branch), yeni durum (restatement branch), ayrıntıları ayırma (elaborating branch) gibi ilaveler yapılabilir (Berry, 1991).

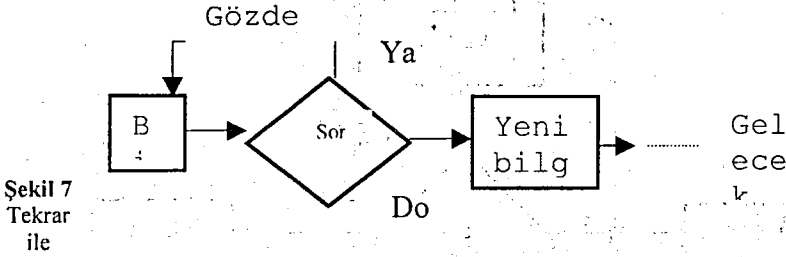
Ayrıca bu tanımlamalara ilave olarak, tekrar ile doğrusal dallanma, ön test ve atlama, tek olarak yardım sağlama ve dallanma, çoklu yardım sağlama dallanması, öğrenci kontrollü dallanma, ek olarak verilen bilgilerin birleştirilmesi ile oluşan dallanma gibi örnekler özellikle hizmet içi ve hizmet öncesi eğitimi sağlama süreçlerinde etkin olarak yer alabilirler (Kemp ve Cochern, 1994). Bu örneklerden bazıları şekil 6, 7, 8, 9 ve 10 da gösterilmektedir. Bu öğretim programı tasarım yolları öğretmen, tasarımcı ve program yazılımcıları tarafından amaca uygun olanı seçilerek öğretim tasarımı ve yazılımı sürecinde etkinlikle kullanılabilir (İpek, 2001).



Şekil 6 Genel dallanma yöntemi

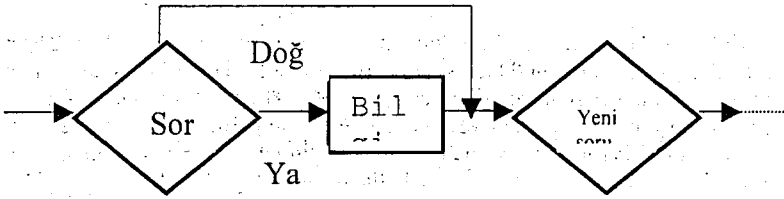
Şekil 6 daki genel dallanma örneğinde görüldüğü üzere bu süreç içinde 1 ve 2 nolu ünite ya da konular sırayla öğretilir. Sonra 3 nolu aşamada sorular sorulur. Bunu izleyen iki yol

üzerinde yeniden yanıtların eksikliği ve doğruluğuna bağlı olarak yeni bilgiler sunulur. Bu aşamadan sonra 5 nolu basamakta öğrenme gerçekleştirilir.



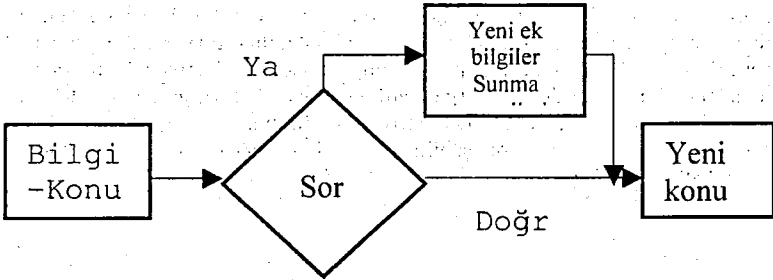
doğrusal dallanma yöntemi (gözden geçirme)

Öğretim sürecinin gözden geçirilmesine yardımcı olan tekrarlar ile dallanma yöntemi işleyişinde şu sıra izlenir. Konular verilir ve sunulur. Hemen sonrasında konunun öğrenilip öğrenilmediğini test etmek için soru verilir. Eğer yanıt doğru değilse ilk basamağa gidilerek konu tekrar gözden geçirilir. Daha sonra tekrar verilen soruya doğru yanıt alındıktan sonra yeni bir konuya geçilir. Sonra bu süreç diğer konular için devam ederek öğrenme gerçekleştirilir (bakınız şekil 7).



ile geçiştirme (atlama)

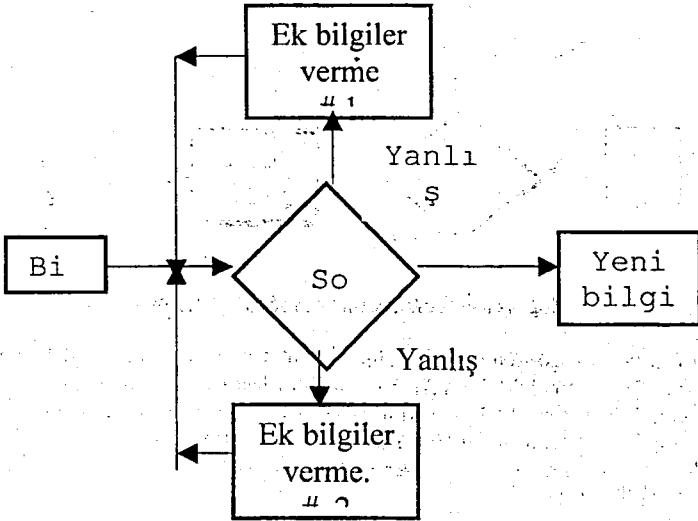
Öğretim sürecinde yeni verilecek konuya ilişkin bilgileri test etmek amacıyla bir ön test verilir. Buradan alınan yanıtlar doğru ve konu yeterli düzeyde biliniyor ise ilgili konu atlanarak diğer konularla ilgili test verilir. Eğer alınan sonuç yetersiz ve yanıt yanlış ise ilgili konu öğretilir. Sonra test verilir. Böylece öğretim süreci devam eder (bak şekil 8).



başına ek bilgiler sunarak dallanma

Şekil 9 da gösterilen dallanma sürecinde, önce bilgi ve konular verilir. Daha sonra bir test sorusu sunulur. Elde edilen yanıt yanlış ise öğrenci için yardımcı ek bilgiler verilerek

konu öğretilir. Sonra yeni konuya geçilir. İlk konudan sonra verilen yanıt doğru ise ek bilgiler sunulmadan ikinci konu ya da ünite ile devam edilir.



Şekil 10 Fazla miktarlarda yeni bilgiler vererek dallanma (çoklu)

Çoklu dallanma tekniği ile bilgilerin öğretilmesi süreci şekil 10 da gösterilmiştir. Bu süreç bilgi ya da konu verildikten sonra soru sorulmayı gerektirir. Buradan alınan her yanlış yanıt için ek bilgiler (remediation) her yanlış yanıt için sunulur. Yanıtlar sonrasında doğru yanıt ulaşıldıktan sonra yeni konuya geçilebilir. Böylece öğretim süreci sürer gider. Burada verilen akış şemaları ve dallanma teknikleri öğretimde etkileşimi sağlayıcı ve artırıcı unsurlardır. Her çeşit tasarım sürecinde kullanılabilir.

#### Sonuçlar

Sonuç olarak özetlersek, burada akış şemalarının tasarımında önemli olan bu sistem ve tekniklerin oluşturduğu problemlerdir. Bunun için aynı bölüm ve kısımların tekrarından sakınmak gerekir. Diğer bir nokta ise, bitmeyen sürekli bir dönüşüm (loop) durumu önlenmelidir. Öğrencinin soruyu doğru yanıtlamadığı durumda izlenecek yol ve ne olabileceği yanında, sorudaki ilk seçenekçe doğru yanıtlanması durumunda, ikinci soruda yanlış yanıt vermesi durumunda gene ne olabileceği şeklinde olmalıdır. Bu tür öğretim durumlarında tasarımcı, öğrencinin öğretim programından problem örnekleri saptayarak, bunların iyi test edilmesinin ve mevcut yolların kontrol edilmesinin sonucunda bu tür problemlerden kaçınma olasılığını bir hayli yükseltebilir. Amacımıza en uygun akış şemaları (çizenekleri) ve öğretim programı tasarımı teknikleri kullanılarak, etkili bir öğrenme ve öğretim sürecinin gerçekleştirilmesine yardımcı olmak kolaylaşır. Bu tekniklerin kullanılması öğretmenlerin ve tasarımcıların katkılarıyla sınıflarımızda anlamlı ve yaratıcı bir öğretim materyali geliştirme işlevine dönüşebilir.

## Akış şemasının hazırlanması, tasarımı ve çizimi

### Planlama ve hazırlama

Önce basit bir akış şeması oluşturulur ve işe girilir.

Sonra önemli kararları içeren akış şemasına ilave edilerek akış şemasının düzeyi ikinci seviyede hazırlanır

Simülasyon veya karmaşık programlar için üçüncü düzeyde akış şeması kullanılır. Karmaşık akış şemalarında çok fazla ayrıntıya girmeden alt adımlar ve basamaklar ile dersin tasarımının tamamlanması gerekir.

Akış şemasındaki bağlantıları birbirleri ile karmaşıklığa meydan vermeksizin ortaya koymak gerekir.

Bu aşamadan sonra öğretim tasarımı için gereken şemaların ve şekillerin uyumu gerekir.

Akış şemasının istenilen ve yapılacak tüm işlemleri içerecek biçimde tasarlandığından ve gösterdiğiinden emin olunması önem taşır.

Akış şeması tasarım planı için BÖ uzmanları ve tasarımcıların görüşünün alınması çok önemlidir.

### Akış şemasının çizilerek gösterimi

Akış şeması boyunca tüm olası yolların planlanıp tasarımının yapılması.

Hangi yolların karar noktalarına gereksinimi olduğuna karar verilmesi.

Var olan değişkenlerin saptanması ve bir yerde tutulması gerekir.

Akış şemasının mantıksal ve istenilen sonuçlara götüreceğinin farkında olunması.

Bütün basamakların öğretim sistem ve tasarım modelleri teori ve uygulamasına uygunluğunun kontrol edilmesi gereklidir.

Program sürecinde bazı hata ve eksikler kontrol edilir ve düzeltilir.

Son olarak tasarım grubu üyelerinin ve uzmanların görüşlerine başvurulur.

Programın akış şeması planlandığı gibi ise, uygun bir program yardımıyla çizimi en son teknikle gerçekleştirilir.

## Kaynakça

- Alessi, S. M., & Trollip, S. R. (1991). *Computer-based instruction: Methods and development* (2<sup>nd</sup> ed.). Englewood Cliffs, New Jersey: Prentice Hall, Inc.
- Berry, L. H. (1991, Spring term). Instructional Design and Technology, 3324 Theory, Design, and Research in Computer-Based Instruction Course Notes. School of Education, University of Pittsburgh. Pittsburgh, PA. USA
- Dills, C. R., & Ramiszowski, A. J. (1997). *Instructional development paradigms*, Englewood Cliffs, New Jersey: Educational Technology Publications
- Gentry, C. G. (1994). *Introduction to instructional development: Process ve technique*. Belmont, CA: Wadworth Publishing Company.
- Gibbons, A. S., & Fairweather, P. G. (1998). *Computer-based instruction: Design and development*. Englewood Cliffs, New Jersey: Educational Technology Publications.
- İpek, İ. (1995). The Effects of window presentation type and field dependence on learning from a CBI geology tutorial, *Dissertations Abstracts International*, (University Microfilms No. DAO 72699 Ann Arbor, MI )
- İpek, İ. (2001). *Bilgisayarla öğretim: Tasarım, geliştirme ve yöntemler*. Ankara: Tıp ve Teknik Kitabevi Ltd. Şti. Yayınları.
- Jonassen, D. H. (1985). Interactive lesson design: A taxonomy. *Educational Technology*, 25(6), 7-17.
- Jonassen, D. H. (Ed). (1988). *Instructional design for microcomputer courseware*.

- Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.
- Jonassen, D. H. (Ed.) (1996). *Handbook of research for educational communications and technology*. New York, NY: Macmillan.
- Luppa, N. V. (1984). *Practical guide to interactive video design*. White Plains, NY: Knowledge Industry Publications, Inc.
- Kemp, J. E., & Cochern, G. W. (1994). *Planning for effective technical training: A guide for instruction and trainers*. Engle wood Cliffs, New Jersey: Educational Technology Publications.
- Merrill, M. D. (1994). *Instructional design theory*. (D. G. Twitchell, Ed.). Englewood Cliffs, New Jersey: Educational Technology Publications.
- Price, R. V. (1991). *Computer-aided instruction: A guide for authors*. Belmont, CA: Wadsworth, Inc.
- Seels, B., & Glasgow, Z. (1990). *Exercises in instructional design*. Columbus, Ohio: Merrill Publishing Company.
- Seels, B., & Glasgow, Z. (1998). *Making instructional design decisions*, (2<sup>nd</sup> ed.) Upper Saddle River, New Jersey: Prentice Hall, Inc.