

## **Öğretim Etkinlikleri Kuramı ve BDÖ Tasarımına Yeni Bir Dinamik Yaklaşım**

**Murat Paşa UYSAL<sup>1</sup>**

### **ÖZET**

Bilgisayar Destekli Öğretim (BDÖ) yazılımlarını tasarlamak ve geliştirmek yoğun zaman ve kaynak gerektiren süreçleri içermektedir. Yazılımların öğretimsel açıdan kalitesini belirleyen önemli unsurlardan birisi uygun öğretim tasarımı süreçlerinden geçirilmeleri, diğeri ise öğretim sırası veya sonrasında ihtiyaçlara süratle cevap verebilmeleridir. İyi bir BDÖ tasarım modeli, öğretim tasarımı ve yazılım geliştirme süreçleri açısından etkili ve verimli bir dinamik yaklaşımın benimsenmesini gerektirmektedir. Bu çalışmada, BDÖ'in doğasına uygun olduğu düşünülen Öğretim Etkinlikleri Kuramı (Instructional Transaction Theory) ile BDÖ yazılımı geliştirme süreçlerine Çevik Yazılım Geliştirme (Agile Software Development) doğrultusunda getirilen dinamik yaklaşım ele alınmıştır.

**ANAHTAR KELİMELER:** BDÖ Tasarımı, Öğretim Etkinlikleri Kuramı, Çevik Yazılım Geliştirme

## **Instructional Transaction Theory and a New Dynamic Approach to CAI Design**

### **ABSTRACT**

Designing and developing Computer Aided Instructional (CAI) softwares requires time-consuming and labor-intensive processes. One of the factors which determines the quality of these softwares is that they undergo appropriate design processes. The other factor is their capabilities of response to the instructional needs during or after instruction. A suitable CAI design model involves adoption of an efficient and dynamic approach to the both instructional design and software development processes. In this study, Both the design model Instructional Transaction Theory which is thought to be suited for the nature of CAI and the dynamic approach, brought to the CAI design in line with the Agile Software Development, are discussed.

**Keywords:** CAI Design, Instructional Transaction Theory, Agile Software Development

---

<sup>1</sup> Doktor, Kara Harp Okulu Savunma Bilimleri Enstitüsü Teknoloji Yönetimi Ana Bilim Dalı, Ankara / Türkiye e-posta:mpuysal@kho.edu.tr

## GİRİŞ

Öğretimin etkili olabilmesi için öğrenenlerin bilişsel, duyuşsal ve fizyolojik özelliklerinin öğretim ihtiyaçlarına göre dikkate alınması, öğretimin bu özelliklere göre uyarlanması gerekmektedir. Teknoloji ve bilgisayar bilimlerindeki hızlı gelişmeler Bilgisayar Destekli Öğretim (BDÖ) ortamlarının kullanımını yaygınlaştırmış, etkili bireysel ve öğrenci merkezli öğretim tasarımına imkan tanımıştır. Bu tür öğretim ortamlarının doğrudan bireye yönelik olması, zengin etkinlik ve çoklu ortam araçlarını sunması, öğretim niteliği ile uygulama biçiminin kolayca değiştirilebilmesi, uyarlanabilir öğretim tasarımı açısından BDÖ ortamlarını ideal bir çözüm olarak karşımıza çıkarmaktadır.

Jonassen'a (1988) göre uyarlanabilir öğretim tasarımı, dışsal (external) ve içsel (internal) olmak üzere iki yöntemle gerçekleştirilebilir. Dış etkenlere göre uyarlamada öğretim yazılımı, öğretim sırasında öğrenenin performansını, bilişsel stilini ve diğer karakteristik özelliklerini değerlendirir, öğretim modüllerini ve konuların sıralamasını öğrenene göre uyarlar. İçsel yöntemde ise öğretimi yapılacak bilginin içeriği ve yapılandırılması uyarlamada belirleyici unsur olup burada yine öğretim yazılımının kendisi, öğretimi ihtiyaçlara göre uyarlamaktadır. Öğretimi bireysel özelliklere veya öğretim durumlarına göre uyarlama ihtiyacı beraberinde öğretim yazılımlarına farklı görevler yüklemiştir. Geleneksel yazılımların aksine uyarlanabilir bir yazılım, öğretime başlanıldığı andan itibaren öğrenen hakkında bilgi toplamalı, öğretimi onun veya konunun ihtiyaçlarına göre dinamik biçimde yönlendirmeli, uzman bir öğretici gibi öğretim ortamını düzenleyip yönetebilmelidir.

BDÖ yazılımlarının etkililiğini arttıran ve zengin bir öğrenme ortamı sağlayan önemli faktörlerden birisi de değişik çoklu ortam araçları kullanılarak bilginin sunulmasıdır. Çoklu ortamlar, öğretimi yapılacak bilginin metin, ses, video, grafik ve canlandırmalar yardımı ile sunulmasını sağlayan çeşitli araçlardır. Çoklu ortam kaynaklarını içeren BDÖ yazılımlarını tasarlamak yoğun iş gücü, kaynak ve zaman gerektiren bir süreçtir. Öte yandan öğretim amaçlı geliştirilen yazılım projelerinin diğer yazılım projelerinden ayrıldığı temel unsur bunların doğrudan bireye yönelik olmalarıdır. Farklı alanlara, disiplinlere ve kurumsal çözümlere yönelik geliştirilen yazılım projelerinde, iş kurallarını tespit etmek ve bu kuralları bilgisayar ortamına aktarmak için gerekli olan süreçler, bilgisayar bilimleri ile yazılım mühendisliğindeki standartlarla belirlenmiştir. İnsanı var eden en önemli unsurlardan birisi olan eğitim ve öğretim alanında ise başta eğitim teknolojisi olmak üzere birçok akademik disiplin, bu alandaki ihtiyaçlara cevap verecek bilimsel ve sistematik yaklaşımları getirmiştir. Bu konulardan birisi de "Öğretim Tasarım Kuramlarıdır."

## BDÖ VE ÖĞRETİM TASARIM KURAMLARI

Bazı BDÖ yazılımlarına getirilen önemli eleştirilerden birisi maliyetlerinin yüksek olmasına karşın, öğretimsel açıdan yetersiz olduklarıdır. Bu yetersizliklerin bir kısmını eğitim teknolojisi kapsamında öğretim tasarımı ve materyal geliştirme konularında, diğerlerini ise yazılım mühendisliği ve proje yönetimi konularında gruplamak mümkündür. Öğretimsel açıdan yetersiz BDÖ yazılımlarının ortaya çıkmasındaki temel etkenlerden birisi de uygun “Öğretim Tasarım Kuramının” kullanılmamasıdır (Merrill, 1991c). BDÖ tasarımında kullanılması düşünülen tasarım kuramı, öğretim tasarımına olduğu kadar yazılımın geliştirilmesi ve güncellenmesiyle ilgili konulara da çözüm getirmelidir. Öğrenenin bireysel özelliklerini dikkate almalı, öğretim tasarımı açısından etkili, öğretim yazılımı geliştirme süreçleri açısından etkin ve verimli olmalıdır. Söz konusu öğretim tasarım kuramıyla geliştirilmesi düşünülen BDÖ yazılımları, fazla kaynak gerektirmeyen bir çabayla, değişen öğretim ihtiyaçlarına hızlı cevap verebilen bir yapıda tasarlanabilmelidir.

Öğretim tasarımıyla ilgili olarak Merrill (1991c), Gagne'nin öğrenme psikolojisi ve öğretim tasarım alanında yapmış olduğu çalışmaların daha sonraki araştırmalara önemli katkıları olduğunu belirtmektedir. Öge Gösterim Kuramı (Component Display Theory), Algoritmik ve Sezgisel Kuram (Landamatics Instructioanal Design Theory) gibi BDÖ'e uygun öğretim tasarım kuramlarını *Birinci Nesil Öğretim Tasarım Kuramları* olarak ifade ederken bu kuramlar hakkında genel olarak aşağıdaki eleştirileri getirmektedir:

**1.** Ders içeriklerinin analizini bütünlük bir yapı yerine parça olarak ele almaktadırlar:

Bu kuramlar, konuları bağımsız parçalar olarak ele aldıkları için içerikleri bütünlük bir yapıda bir bilgi tabanı oluşturamazlar. Dolayısıyla analiz sonucunda ortaya çıkan içerik yapısı, doğrudan öğretim yazılımında kullanılamamaktadır. Konu içeriklerinin ayrı olarak ele alınması, analiz edilen içerik tasarımının tam ve bütünlük biçimde öğretim yazılımına yansımaya da engel olmaktadır (Merrill, 1991c).

**2.** Bilginin edinilmesinde ve bu bilginin öğrenenlerde zihinsel model olarak oluşturulmasında yetersizlikleri vardır:

Merrill'e (1991c) göre birinci nesil kuramlar, karmaşık ve dinamik etkileşimi gerektiren öğretim konularının öğretiminde yetersizdirler. Karmaşık bilgi yapıları ve aralarındaki ilişkileri kavram ve kurallar boyutunda bağımsız olarak işlemelerinden dolayı, öğrenenlerin sunulan bilgileri bir bütün olarak algılamada zorlanmalarına neden olmaktadır. Bundan dolayı birinci nesil kuramların öğretilen konularla ilgili zihinsel modellerin oluşumunda etkili olmadıkları belirtilmektedir.

**3.** Birinci nesil kuramlarla tasarlanan öğretim sistemleri kapalı sistemlerdir:

Mevcut bilgi yapıları üzerine tasarlandıkları için yeni bilgi ve öğretim durumlarının oluşması durumunda, bu bilgileri öğretim sistemiyle

bütünleştirecek dinamik yapılara sahip değildir. Öğretime yönelik tanımlamalar, konu örnekleri, alıştırmalar ve değerlendirme gibi tasarım öğeleri, aynı zamanda bilginin sunumuyla ilgili konuları da belirlemektedir. Bu kuramlar, öğretim stratejilerini öğretim durumlarına göre dinamik biçimde düzenlenme ve uyarlanma konularında yetersizlerdir. Bundan dolayı öğretim tasarımcısı, sonradan veya öğretim esnasında gelişen durum ve ihtiyaçlara bağlı olarak tasarım öğelerini her seferinde yeniden gözden geçirmek, düzenlemek ve yeni konuları öğretim sistemiyle bütünleştirmek zorunda kalmaktadır.

4. Birinci nesil kuramlar, öğretim tasarımının her aşamasını bir bütün yapı yerine birbirinden bağımsız aşamalar olarak ele alırlar:

BDÖ ortamlarının tasarımında karşılaşılan önemli sorunlardan bir diğeri de bu ortamların tasarım aşamalarının bütünleşik hale getirilmesindedir (Merrill, 1991c). Birinci nesil kuramlarda genel olarak analiz, tasarım, geliştirme ve değerlendirme olarak dört tasarım aşaması bulunmaktadır. Her bir tasarım aşamasının çıktısı, diğer aşama için girdiyi oluşturmada, tasarım süreçleri tekrarlı bir biçimde biri diğerinin devamı ve uzantısı olacak şekilde ele alınmaktadır. Her aşamada farklı yöntem, araç ve bilgi gösterim yöntemi kullanılmaktadır. Ancak, tasarım ve öğretim gereksinimlerine bağlı olarak bir tasarım aşamasındaki değişikliğin, diğer aşamaya doğrudan nasıl yansıtılacağı açık olarak ifade edilmemiştir. Ayrıca bu kuramlarda, öğretim analizi ile öğretim etkinlikleri ve bilginin sunum biçimi arasında doğrudan tanımlanmış bir bağ veya yapısal ilişki bulunmamaktadır.

5. Birinci nesil kuramlar bilgisayar destekli öğretim ortamlarının tasarımı için etkili ve verimli yöntemler sunmamakta, çoğu zaman etkileşim yönünden yetersiz ürünler ortaya koymaktadırlar:

Öğretim etkinlikleri daha çok tek taraflı bir bilgi sunumu şeklinde gerçekleşmekte ve buna bağlı olarak tasarlanan etkileşimler yetersiz olmaktadır. Öğrenme sırasında ortaya konulan zihinsel çaba ile öğrenmenin gerçekleşmesi arasında önemli bir ilişki bulunmaktadır. Bundan dolayı öğretim tasarımında, öğretilen kavram ve kuramlar ile öğrenenin sarf edeceği zihinsel çaba arasında bir bağ kurulmalıdır. Öğretim tek taraflı bilgi sunumu biçiminde gerçekleştiği sürece, öğrenen kendi bilişsel yapılarını sorgulama gereğini hissetmeyecek, kalıcı olmayan, önceki öğrenmelerle ilişkilendirilmemiş ve yeni durumlarına transfer edilemeyen öğrenmeler gerçekleşecektir (Merrill, 1991c).

6. Bu kuramlar, BDÖ ortamlarının yaygınlaşmasından önce tasarlandıkları için bu tür ortamların tasarım ihtiyaçlarına bütünüyle cevap verememektedirler. Birinci nesil öğretim kuramları kullanılarak tasarlanan öğretim ortamlarına getirilen son eleştiri ise yoğun iş gücü ve zaman gerektirmeleridir. Öğretim tasarımcısı her dersin sunumunda gerekli olan tasarım elemanlarının neler olduğunu belirlemek, analiz etmek ve tasarlamak zorundadır. Bir saatlik dersin tasarımı ve bilgisayar ortamına aktarımı için uzun süreli bir çalışmaya ihtiyaç duyulmaktadır.

Bu konuyla ilgili olarak Anderson (1999), “Öğretim Etkinlikleri Kuramıyla” (Instructional Transaction Theory) (ÖEK) ilgili çalışmasında, günümüzde

kullanılan öğretim tasarımı ve yazarlık araçlarıyla ilgili eksiklikleri üç ana başlık altında toplamaktadır:

1. Üretici firmalar tarafından geliştirilen bazı yazarlık araçları, yine bu firmalarca belirlenen ve kendilerine özgü değişik standartlara bağımlıdır. Bu durum, hem öğretimsel hem de teknolojik açıdan tasarımcı ve uygulayıcılara çeşitli kısıtlamalar getirmektedir.

2. Söz konusu yazarlık araçları, öğretim tasarım kuramlarına karşı tarafsız olmak zorundadırlar. Ticari kaygıdan kaynaklanan bu durum, firmaların geliştirdikleri yazarlık araçlarında öğretim veya öğrenme yöntemlerine değil, daha çok bilgi sunum teknikleri ve çoklu ortam araçlarının geliştirilmeleri üzerine yoğunlaşmalarına neden olmuştur.

3. Bu yazarlık araçları, genellikle tek tip tasarım ve bilgi sunum yapılarına sahip olup öğretim amaçlı bilgi gösterimi ve analizinde yetersizlikleri bulunmaktadır.

Yukarıda anlatılan konular, yeniden kullanılabilir çoklu ortam araçlarına dayalı, değişen öğretim durum ve tasarım ihtiyaçlarına hızla cevap verebilen, sunulan bilgi ve kullanılan öğretim stratejilerini birbirinden ayıran ve öğretim etkinlikleri üzerinde yoğunlaşan “*Öğretim Tasarım Kuramına*” olan ihtiyacı ortaya koymaktadır.

### **ÖĞRETİM ETKİNLİKLERİ KURAMI (INSTRUCTIONAL TRANSACTION THEORY)**

Öğretim tasarımı, öğretim materyallerinin sistematik tasarımı ve geliştirilmesine yönelik süreç ve etkinliklerle ilgili kuralları içermektedir. Bu kurallardan birisi de tasarımda kullanılacak öğretim stratejilerinin belirlenmesidir. Gagne'nin belirttiği gibi öğretimin etkili olabilmesi için değişik öğretim amaçlarına yönelik farklı öğretim stratejilerinin kullanılması gerekmektedir (Akt. Merrill, 1996). Günümüz bilgisayar teknolojileri ve program geliştirme süreçlerini dikkate alan Öğretim Etkinlikleri Kuramı (Instructional Transaction Theory) (ÖEK), kendi tasarım yaklaşımını “bilgisayar programı” kavramına dayandırmaktadır (Merrill, 1996). Konu içerikleri ve sunulacak bilgiler, bir bilgisayar programının kullandığı veri gibi öğretim stratejilerden bağımsız olarak düşünülürken, öğretim etkinlikleri ve stratejileri birer algoritma yapısı kapsamında ele alınmaktadır. Bunun sonucunda öğretim stratejilerinden bağımsız olarak içerikler belirlenerek aynı içerik değişik stratejilerle farklı konuların öğretiminde kullanılabilir. ÖEK, temel olarak uzman öğretim sistemlerinin otomatik tasarımı ve geliştirilmesini hedeflemekte ve Tablo 1'deki gibi üç ana bileşenden oluşmaktadır. Bunlar:

1. Öğretilecek bilgi ve becerileri tanımlayan bileşen (Bilgiyi Tanımlayan Ana Bileşen),

2. Öğrenmenin gerçekleşebilmesi için ihtiyaç duyulan öğretim stratejilerini belirten bileşen (Stratejileri Tanımlayan Ana Bileşen),

3. Öğretilecek bilgiyi ve bunun için gerekli öğretim stratejilerini birbiriyle ilişkilendirerek öğretimi düzenleyen bileşendir (Öğretim Tasarımı Ana Bileşeni).

Tablo 1. ÖEK' Temel Bileşenleri

<b>Bilgiyi Tanımlayan Ana Bileşen</b>	<b>Stratejileri Tanımlayan Ana Bileşen</b>
<ul style="list-style-type: none"> <li>• Bilgi nesneleri</li> <li>• Bilgi nesne ilişkileri</li> </ul>	<ul style="list-style-type: none"> <li>• Etkinlik ortamı (transaction shell)</li> <li>• Öğretim durumu parametreleri</li> </ul>
<p><b>Öğretim Tasarımı Ana Bileşeni</b></p> <ul style="list-style-type: none"> <li>• Bilgi nesnelерinin seçimiyle ilgili kurallar,</li> <li>• Bilgi nesnelерini düzenleyen ve sıralayan kurallar,</li> <li>• Etkinlik grubunun (transaction) seçimiyle ilgili kurallar,</li> <li>• Etkinlik grubunu düzenleyen kurallar,</li> <li>• Etkinlik grubunu uygulayan öğretim algoritmaları,</li> <li>• Bireysel farklılıklara göre öğretimin uyarlanabilmesi için öğretim durum parametrelerini düzenleyen kurallar.</li> </ul>	

(Kaynak: Merrill, 1996)

### ÖEK'nda Bilgiyi Tanımlayan Ana Bileşen

ÖEK açısından bir bilgi, öğretilecek konu içeriğinin belirli kurallara göre tanımlı bir gösterimidir. Bu bilgi yapısı, değişik çoklu ortam araçlarıyla sunulurken aynı zamanda bu çoklu ortam kaynaklarının öğretilecek bilgidен bağımsız olarak ele alınması sağlanmaktadır. Bir Bilgi Nesnesi, bilgi tabanında kayıtlı olan değişik çoklu ortam araçları kullanılarak farklı görünümde öğrenene sunulabilmektedir. Bilginin bu görünümleri, metin, resim, grafik, video veya ses olarak bilgi nesnesinin canlandırılmış biçimlerini oluşturmaktadır. Öğretim tasarımı bu tür bir yapılanma, öğretimde kullanılan strateji ve bilgilerin, ihtiyaca göre öğretim sistemini etkilemeden değiştirilebilmesini sağlamaktadır (Merrill, 1991a). ÖEK, bilgiyi üç çeşit bilgi nesnesi bileşeniyle tanımlar. Bu bilgi bileşenleri, öğretimi yapılacak varlık, öğrenenin bu varlıkla yapmış olduğu etkinlikler ve bu etkinlikler sonucunda tetiklenen süreçlerdir.

### Bilgi Nesneleri

Bilişsel psikoloji, zihinsel modelleri iki bölümde incelemektedir. Bilgi yapıları ve şemalar birinci bölümü oluştururken bu bilgileri kullanan süreçler ise diğer bölümde yer almaktadır. Öğretim tasarımını ilgilendiren temel konulardan birisi de öğretim amacıyla öğretim içeriklerinin nasıl düzenleneceği ve

gösterileceğidir. Buradan hareket eden uzmanlar, bilginin dış dünyadaki gösterimi ve temsili ile aynı bilginin insan zihnindeki oluşumu ve yapısı arasında bir ilişki olduğunu düşünmektedirler. Öğretimi yapılacak içeriklerin sistematik biçimde analiz edilmesi iki süreci kolaylaştırmaktadır. Birincisi, bu tür bir bilgi analizinin öğretim tasarımı açısından bilginin dışsal temsilini ve gösterimine imkân tanınması, ikincisi ise öğrenen açısından bilginin zihinsel model şeklinde içsel olarak kendi zihninde oluşmasına katkıda bulunmasıdır (Merrill, 2000).

BDÖ tasarımında, öğretim hedefleri için değişik öğretim durumları ve bu durumlarla ilişkilendirilmiş bilişsel bilgi yapılarının kullanımı gerekmektedir (Merrill, 2000). Bilginin gösterimi ve yapılandırılma süreci farklı bilgi yapılarının oluşumunu da sağlamaktadır. Öğrenenin zihninde bulunan zihinsel modellerin bir parçası olan zihinsel süreçler bu bilgi yapılarını kullanırlar. Öğrenenler bilgilerini, belleklerinde oluşturdukları bilgi yapıları ile temsil etmektedirler (Akt. Merrill, 1992). Bu tür bilgi yapıları, öğrenenin konuyla ilgili bir problemi çözebilmesi için ihtiyaç duyduğu bilgi ve bileşenlerini içermektedir. Eğer bilgi bileşenleri ve aralarındaki ilişkiler uygun tanımlanamamış veya yetersizse, bilginin kullanımını gerektiren problemlerin çözümünde etkili olunamayacaktır. Zihinsel modeller, öğrenenin belleğindeki şemada bulunan bilgiler ile bu bilgileri kullanan süreçleri birleştirirler. Bir problemin çözümünde, sadece konuyla ilgili bilgiye ve ona ait bileşenlere sahip olmak yeterli değildir. Aynı zamanda bu bilgi bileşenlerini kullanan ve problemin çözüm adımlarını oluşturan algoritmaların da öğrenilmesi gerekmektedir (Merrill, 2000).

Bilginin yapılandırılması ve bilgi yapılarının kullanılmasının başta uzman öğretim sistemleri olmak üzere BDÖ tasarımına önemli katkıları bulunmaktadır. Bir bilgi yapısı konu ile ilgili kavramları ve bu kavramlar arasındaki ilişkileri temsil eder. Bilginin standart biçimde elektronik bir veritabanında gösterimi BDÖ için önemli bir tasarım aşamasını oluşturmaktadır (Merrill, 1992). Buna göre bir öğretim stratejisi, bir konunun öğretimi için gerekli olan bilgi kaynakları ve bileşenlerin kullanımını içeren bir çeşit algoritmadır.

“*Bilgi Nesnesi*”, öğretilecek konu veya içeriği tanımlama, yapılandırma ve konunun öğretimi için gerekli olan bilgiye ait bileşen ve bölümlerin neler olduğunu belirlemede kullanılan bir tür bilgi analiz ve gösterim yöntemidir (Merrill, 1998a). Bilgi nesnelere aynı zamanda içerikle ilgili kaynakları (metin, ses, video, grafik, vb.) düzenleme yöntemidir. Böylece, değişik içeriklere ve öğretim durumlarına göre bir öğretim algoritmasının kullanılabilmesi bilgi tabanı oluşturulmaktadır. Diğer bir ifadeyle bir bilgi nesnesi, içerikle ilgili çoklu ortam kaynaklarının, önceden tasarlanan farklı öğretim stratejileriyle kullanılabilmesini sağlayan bir bilgi tabanı (knowledge base) düzenleme yöntemidir. Bilgi nesnesini beş temel “bilgi bileşeni” oluşturmaktadır. Bunlar:

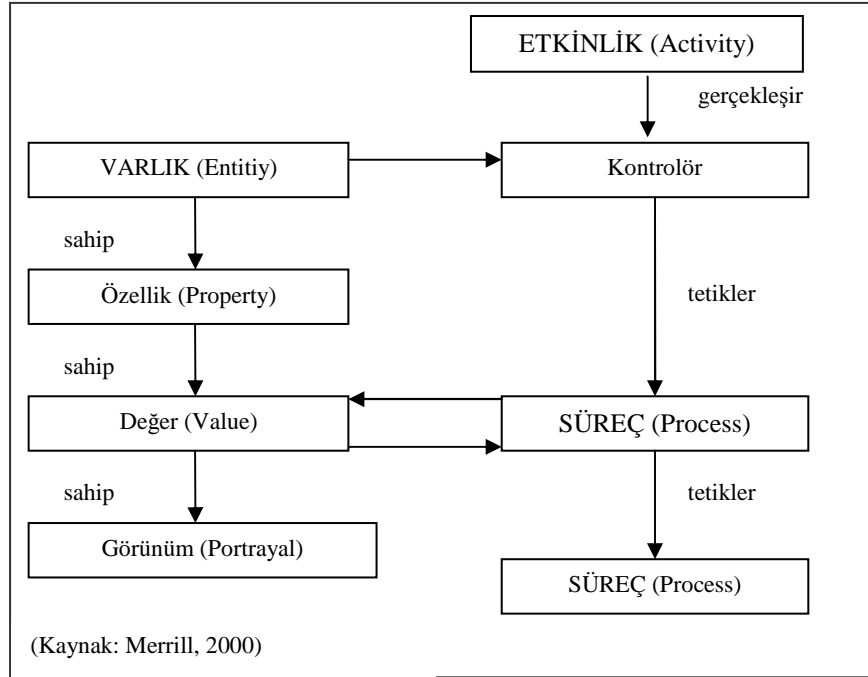
- 1.Varlık (Entity): Öğretilecek bir araç, yer, sembol, konu, kavram veya nesnedir.
- 2.Özellik (Property): Varlıkla veya onun bileşenine ait, onu tanımlayan, nitelleyen veya nicelik olarak belirleyen özelliklerdir.

3.Bölüm (Part): Varlığı oluşturan bölümleridir. Bütün varlıklar, daha küçük varlık veya bölümlere ayrılabilir. Bu işleme, varlığı tam olarak temsil edebilecek kavramsal seviyeye kadar devam edilebilir.

4.Etkinlik (Activity): Öğrenenin varlığa yönelik veya varlıkla beraber yapmış olduğu faaliyet ve etkinliklerdir. Bu etkinlikler de yine alt bölüm veya etkinliklere ayrılabilir. Her bir etkinlik bir sürecin başlamasını tetikleyerek varlığın özellik veya durumunda değişikliğe neden olur.

5.Süreç (Process): Öğrenenin yapmış olduğu bir etkinlik veya başka bir süreç tarafından tetiklenen, varlığa ait bileşenlerin veya özelliklerinin değerini değiştiren olaylardır. Süreçler, bir takım durumların veya şartların oluşması sonucunda işlemeye başlar ve değişik çoklu ortam kaynakları ile öğrenene bilgi sunarlar.

Bir bilgi nesnesi, bilgi bileşenleri için bir altyapı ve çalışma çerçevesini belirlemekte ve her bir bilgi bileşeni, bir öğretim amacına yönelik özel bir bilgiyi tanımlamaktadır. Bir varlığı öğrenmek, nesneye ait yapıyı, ona ait bölümleri ve fonksiyonlarını öğrenmek demektir. Öğrenenin nesne ile ilişkili bir etkinliği öğrenmesi, bu nesnenin kullanılmasına yönelik becerileri kazanmasını sağlar. Nesneye ait süreci öğrenmek ise öğrenenin yapmış olduğu etkinliklerin, bu nesne ve onunla ilgili bilgi bileşenleri üzerindeki etkilerini öğrenmesini, çeşitli tahmin ve yorum becerilerinin gelişmesini sağlayacaktır (Merril, 2000).



Şekil 1. Bilgi Nesnesi Bileşenleri Neden-Sonuç Ağ Yapısı



Bilgi nesnesi bileşenlerden olan varlık, etkinlik ve süreç arasındaki neden sonuç ilişkisi, PEA-Net (Process, Entity, Activity Network) olarak ifade edilen bir bilgi yapısında yukarıda Şekil 1'deki gibi gösterilmiştir. Bu gösterim yöntemi kullanılarak gerçekleştirilen süreç, nesne ve etkinlik bileşenleri arasındaki eşleştirme ve ilişkilendirme işlemi, bilginin bütünleşik olarak gösterimini temsil etmektedir. Sürecin harekete geçmesi için gerekli olan şartları, bilgi nesnesinin başka bileşenlerine ait özellikler belirlemekte, sürecin sona ermesi sonucunda, bu özelliklerden bir veya birkaçının değerinde veya görünümünde değişimler gözlenmektedir.

### **ÖEK'nda Stratejileri Tanımlayan Ana Bileşen**

ÖEK'nda öğretimle ilgili bütün etkinliklerin bulunduğu öğretim algoritması, "Etkinlik Ortamı" (Transaction Shell) olarak ifade edilmektedir. Bir etkinlik ortamı, gerekli olan bilgi nesnelerinin seçimi, nesnelerin düzenlenmesi ve sıralanması ile ilgili bütün kuralları içermektedir. Bu kurallar, bilgi nesnesinin, bilgi tabanında bulunan çoklu ortam kaynaklarından hangisi ile gösterileceğini de kapsamaktadır. Böylece bir etkinlik ortamı öğretim ihtiyaçları, değişik öğretim durumlarına göre bilgi nesnelerinin özellik ve değerlerinin neler olacağı ile ne tür çoklu ortam aracıyla (metin, grafik, video, ses vb.) sunulacağı konularını düzenlemiş olmaktadır. Öğretim stratejileri ise bu bilgi nesnelerini kullanan algoritmalar olarak tanımlanır (Merrill, 1991a). Böylece öğrenenin konuyla ilgili bilgi ve becerileri kazanabilmesi için gerçekleştirmesi gereken öğretim etkinliklerinin bütünü "Etkinlik Grubu" (Transaction) olarak gruplanmaktadır.

### **ÖEK'nda Etkinlik Gruplarının Sınıflaması**

Öğretilecek bilgiye ve öğretim hedeflerine yönelik değişik türde ana etkinlik grupları tanımlanmıştır. Bunlar, Bileşen Etkinlik Grupları (Component), Genelme Etkinlik Grupları (Abstraction), İlişkilendirme Etkinlik Grupları'dır (Association) (Merrill, 1991b).

*Bileşen Etkinlik Grubuna* ait alt etkinlik gruplarında, bir konunun öğretimi için gerekli bilgi bileşenleri, varlık veya ona ait bölüm ve parçaları, bu varlığa yönelik veya varlığı içeren etkinlik ve adımları, bu etkinlik sonucunda oluşan süreçler ve ilgili olaylar öğretilmektedir. Bileşen etkinlik grupları üçe ayrılırlar. Bunlar, bir varlığın tanımaya ve tanımlanabilmesine yönelik Tanımlama Etkinlik Grubu (Identify), bir etkinliğin uygulanmasına yönelik Uygulama Etkinlik Grubu (Execute), bir sürecin tahmin edilmesi ve yorumlanabilmesine yönelik Yorumlama Etkinlik Grubudur (Interpret). Bileşen etkinlik grupları, diğer sınıflardaki etkinlik grupları için ön öğrenmeleri teşkil ederler (Merrill, 1991b).

*Genelleme Etkinlik Grupları*, önceden öğrenilmiş ve bir sınıfa ait bilgi veya becerilerin, daha önce karşılaşılmamış başka bir örneğe veya konuya aktarılmasını sağlamaktadır. Aynı zamanda bilgi nesneleri arasındaki sınıf, alt ve

üst sınıf ile yine bu sınıflara ait konu örnekleri arasındaki ilişkiler gösterilmektedir. Bu sınıftaki etkinlik grupları, öğretimi yapılan konu ve varlıkların yardımıyla öğrenenin zihninde genel bir zihinsel model oluşturarak bilgiyi genellemektedirler. Değerlendirme (Judge) Etkinlik Grubu, Sınıflama (Classify) Etkinlik Grubu, Genelleme (Generalize) Etkinlik Grubu, Karar Verme (Decide) Etkinlik Grubu ve Aktarma (Transfer) Etkinlik Grubu adı altında beş çeşit alt etkinlik grubu bulunmaktadır (Merrill, 1991b).

*İlişkilendirme Etkinlik Grupları*, öğrenenin farklı konulardaki bilgi ve becerileri kullanarak ilişkili bilgi yapılarını bir bütün haline getirebilme bilgi ve becerisini kazandırmaktadır. Öğrenen önceden oluşturmuş olduğu zihinsel modelleri kullanarak yeni bir model oluştururlar. Böylece konu alanıyla ilgili yeni bir varlık, etkinlik ve süreci geliştirebilme imkanı sağlanmış olacaktır. Bilgi Akışı (Propagate) Etkinlik Grubu, Benzetim (Analogize) Etkinlik Grubu, Yerine Kullanma (Substitute) Etkinlik Grubu, Tasarım (Design) Etkinlik Grubu ve Keşif (Discover) Etkinlik Grubu olarak farklı alt etkinlik gruplarına ayrılmaktadır (Merrill, 1991b).

Bir etkinlik grubu, öğrenenle etkileşimin gerçekleşmesi için çeşitli öğretim görevlerini yerine getirmelidir. Bunun için öğretim durumlarını belirleyen öğretim parametrelerinin dikkate alınması gerekmektedir. Öğretim tasarımında kullanılan bütün etkinlik gruplarının dört temel işlevi vardır: Bunlar:

1. Bilginin Seçimi: Bilgi nesnesinin seçilmesi ve sunulmak üzere gerekli bileşenlerin belirlenmesi,
2. Bilginin Sıralanması: Öğrenene rehberlik ve öğretimin yönlendirilmesi amacıyla bilginin düzenlenmesi ve öğretimde sunulmak üzere sıralanması,
3. Etkileşimlerin Tasarımı ve Yönetilmesi: Etkinliklerde sunulacak bilginin öğretimini gerçekleştirmek amacıyla etkileşimlerin belirlenmesi ve öğretim sırasında yönetilmesi,
4. Etkileşimlerin Uygulamaya Konulması: Tasarlanan etkileşimlerin planlandığı biçimde uygulamaya konulmasıdır.

### **ÖEK’nda Öğretim Tasarımı Ana Bileşeni:**

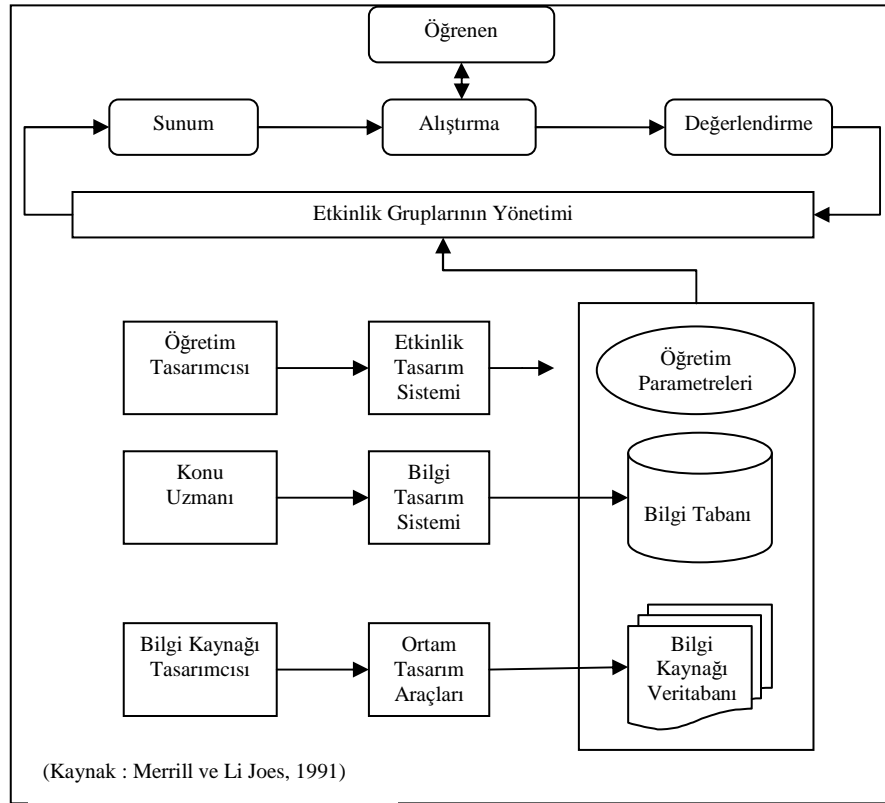
ÖEK’na dayalı BDÖ sistemlerinin tasarımında altı önemli tasarım adımının uygulaması gerekmektedir (Merrill, 1991a). Bunlar:

1. Öğretim için gerekli olan bilgi nesnelerinin seçimi,
2. Bilgi nesnelerinin konunun öğretimi için düzenlenmesi ve sıralanması,
3. Belirlenen bilgi nesnelerinin öğretimi için gerekli olan etkinlik gruplarının seçimi,
4. Etkinlik gruplarının düzenlenmesi ve sıralanması,
5. Öğretim sistemi ve öğrenen arasında gerekli olan etkileşimleri sağlayacak etkinlik gruplarının uygulanmaya konulması,

6. Öğretim sisteminin öğrenenin bireysel özelliklerini dikkate alacak biçimde etkinlik gruplarını öğretim durumlarına uyarlamasıdır.

### ÖEK’nda Öğretim Etkinlik Ortamı Tasarım Bileşenleri

Bir öğretim etkinlik grubu, daha önce belirtildiği gibi konuyla ilgili bilgi ve becerinin edinilmesi amacıyla öğrenen ve sistem arasında gerçekleştirilen bilgi alışverişleri, etkileşimler ve gerçekleştirilen etkinliklerdir. Dolayısıyla farklı bilgi, beceriler ve öğretim amaçları için değişik yapıda etkinlik gruplarının tasarlanması gerekmektedir. Öğretim etkinlik ortamı (Transaction Shell), etkinlik grubu için gerekli olan etkileşimleri, bilgisayar destekli öğretim ortam parametrelerini ve ilgili bilgi nesnelere tanımlayan ortamdır (Merrill, 1991b). Öğretim etkinlik ortamını oluşturan bileşenler Şekil 2’de gösterilmiştir.



Şekil 2. Öğretim Etkinlik Ortamı Tasarım Bileşenleri

1. Etkileşimler ve Etkileşimlerin Yönetimi: Bir bilginin öğretilmesi amacıyla etkinlik gruplarının gerçekleşmesini sağlar.
2. Öğretim Parametreleri: Öğrenenin bireysel özellikleri, öğretim ortamına ait değişkenler, öğretim görevine göre öğretimin uyarlanmasını sağlayan

öğretimle ilgili değer, özellik veya durumlardır. Bilginin sunum sırası, etkileşim biçimi, öğretim stratejileri, öğretim ortamı değişkenleri ve etkinlik gruplarının sırası öğretim parametrelerine örnek olarak gösterilebilir.

3. Bilgi Tabanı: Öğretimi yapılan bütün bilgilerin bilgi nesnesi biçiminde yapılandırıldığı veritabanıdır.

4. Kaynak Bilgi Tabanı: Öğretim sırasında sunulacak bilginin çeşitli çoklu ortam biçimleriyle canlandırıldığı, bilginin çeşitli görünümünü içeren bir veritabanıdır.

5. Bilgi Tasarım Sistemi: Öğretimi yapılacak bilginin yapılandırılmasını sağlar ve öğrenene rehberlik eder. Farklı konularla ilgili bilgileri veritabanında tutar ve bilgi nesnelerini ilişkilendirerek etkinliklerin tasarımında kullanılacak bilgileri içermektedir.

6. Bilgi Kaynağı Editörleri: Bilgi nesnesi formatında yapılandırılan bilgilerin çeşitli çoklu ortam araçlarıyla tasarımını sağlar. Tasarlanan bu kaynaklar etkinliklerin sunumunda kullanılır.

7. Etkinlik Grubu Tasarım Sistemi: Bir etkinlik grubu aslında öğretim amaçlı bir bilgisayar programıdır. Uygulamaya konulduğunda öğrenene etkileşimde bulunmak için hangi bilgileri içermesi gerektiğini bilir. Bu etkinlik grupları arasından öğretim için gerekli olan etkinliklerin seçimini ve öğretimde kullanılacak öğretim parametrelerinin çeşitli etkinlik ortamları için tasarımını sağlamaktadır.

### Öğretim Etkinlik Ortamının İşlevleri

Öğretim tasarımında kullanılan bütün etkinlik grupları (bileşen, genelleme, ilişkilendirme), konuya yönelik olarak öğretilecek bilginin seçimi, sıralanması, etkileşimlerin yönetimi ve etkileşimlerin uygulamaya konulması gibi öğretim görevleriyle ilgili temel işlevleri yerine getirmektedirler. Bu görevler aşağıda ana başlıklar altında özetlenmiştir.

#### 1. Bilginin Seçilmesi:

Bilginin seçilmesi, öğretimin uygulanması sırasında bilgi tabanında bulunan bilgilerden o anda hangisinin kullanılacağını belirlemesidir. Her bir öğretim modülü, değişik öğretim olaylarını, konulara ait bölümleri ve konunun uygulama adımları gibi bilgi bileşenlerini içermektedir. Bu bilgi bileşenleri, kaynak bilgi tabanında bulunan çeşitli çoklu ortam araçlarıyla öğrenene öğretim etkinlikleri içinde sunulurlar.

#### 2. Bilginin Sıralanması:

Bilginin hangi sırayla öğrenene sunulacağını belirlemesidir. Etkinliklerde sunulacak bilginin kapsamı geniş ise uygun büyüklükteki parçalara ayrılarak sunulur.

#### 3. Etkileşimlerin Yönetimi:

Etkileşimlerin yönetimi, seçilen ve sunum için sıralanan bilgi nesnelere ile ne tür etkileşimlerde bulunulacağını belirlemesidir. Bu işlem çeşitli öğretim stratejilerinin kullanımını gerektirmektedir. Tasarlanan etkinlik gruplarında etkileşim modu olarak tanımlanan, bilginin sunumu, alıştırmaya ve değerlendirme etkinliklerine yer verilir.

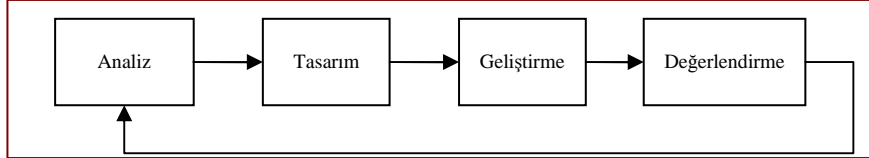
#### 4. Etkileşimlerin Uygulanması:

Burada her etkileşim modunun nasıl uygulanacağı belirlenir. Etkileşimlerde bilginin sunum biçimi, öğrenenin verebileceği tepkiler ve sunulacak geribildirimler kesinleştirilir. Öğrenenin özelliklerine göre öğretimin nasıl uyarlanacağına karar verilir.

### BDÖ TASARIMINA DİNAMİK BİR YAKLAŞIM

#### Doğrusal BDÖ Tasarımı

Şekil 3'te görüldüğü gibi literatürdeki BDÖ tasarımlarında genellikle doğrusal bir yaklaşım benimsenmekte, tasarım süreçleri *Analiz, Tasarım, Geliştirme* ve *Değerlendirme* olmak üzere dört aşamadan oluşmaktadır.



Şekil 3. Doğrusal BDÖ Tasarımı

Analiz aşamasında, problem, ihtiyaç ve amaçlarla ilgili çözümler yapılarak nelerin öğretilmesi gerektiği belirlenmektedir. Öğrenenlerin bireysel özellikleri ve hazır bulunuşluk seviyelerinin nasıl ölçüleceğiyle birlikte öğretim modüllerinin ana hatları kararlaştırılıp konular hiyerarşik biçimde yapılandırılmaktadır.

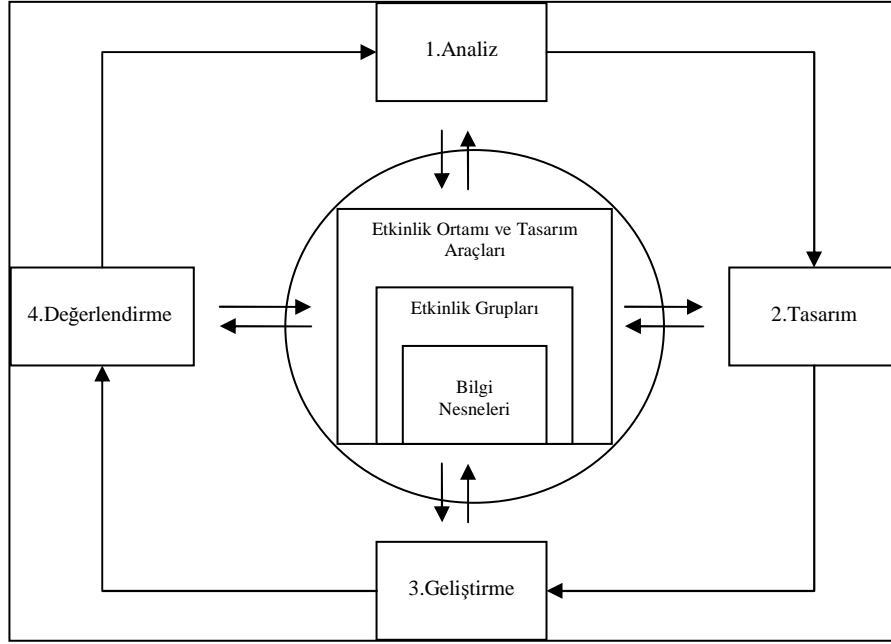
Tasarım ve geliştirme aşamasında öncelikle giriş etkinlikleri planlanmaktadır. Öğretim hedeflerinin öğrenene bildirilmesi ve eski ile yeni bilgilerin birbiriyle ilişkilendirilmesi önemlidir. İçeriğin sunumu planlanırken bilginin miktarı ve öğretim stratejileri belirlenir. Daha sonra, alıştırma etkinlikleri, geribildirim ve değerlendirmeyle ilgili konular ele alınmaktadır. Tasarım aşamasında BDÖ yazılımının geliştirilmesine yönelik ana çerçeve de tespit edilmektedir. Bu çerçeve doğrultusunda uygun yazarlık dillerinden birisi kullanılarak ürün gerçekleştirilmektedir. Geliştirilen BDÖ yazılımının uygulanmasına geçilmeden önce tercihen pilot bir uygulama ile uzman görüşleri alınmaktadır. Bu görüşler doğrultusunda tasarım süreçleri ve ürün değerlendirilmekte, gerekli düzenlemeler yapılmaktadır. BDÖ'ün uygulama sonrası elde edilen sonuçların ışığı altında, analiz, tasarım, geliştirme aşamaları gözden geçirilerek yeniden ele alınabilmektedir.

Doğrusal BDÖ yazılımları tasarım ve geliştirme modellerinde tasarım aşamaları birbirlerinden kesin çizgilerle ayrılmıştır. Bir aşamanın çıktısı diğer aşamanın girdisini oluşturmaktadır. Her aşamada kendine özgü araç ve yöntemler kullanılarak tasarım öğeleri kendi çerçevesinde ele alınmaktadır. Bu modellerde, geliştirme süreçlerine geçilmeden önce yazılımın öğretim tasarımı ve gerçekleşmesiyle ilgili temel konuların büyük ölçüde belirlenmesi gerekmektedir. Tasarlanan BDÖ yazılımı, bilgisayar programı ve yazılım projeleri geliştirme standartları doğrultusunda bilgisayar ortamına

aktarılmaktadır. Öğretim tasarımı yapan kişiler ile yazılım geliştiriciler çoğunlukla farklı kişiler olmakta, yazılımcının konuya yaklaşım biçimi ürünün öğretimsel kalitesini belirlemektedir. Ancak, birçok yazılım projesinde olduğu gibi BDÖ yazılımları da uygulamaya konulmadan önce tasarım ve geliştirmeyle ilgili önemli değişikliklere ihtiyaç duyabilmektedir. Sonuç olarak, uygulama sırası ve sonrasındaki öğretim ihtiyaçları ile ders ve yöntem değişikliğiyle ilgili gereksinimler, bazen yazılımın yeni baştan ele alınmasını ve tersine mühendisliğin (reengineering) uygulanmasını gerektirebilmektedir.

### BDÖ Tasarımına Dinamik Yaklaşım

Daha önce belirtildiği gibi BDÖ yazılımlarının gerçekleştirilmesi tasarım ve geliştirme aşamaları bakımından zahmetli süreçleri içermekte, yoğun zaman ve kaynak kullanımını gerektirmektedir. Şekil 4'te bu çalışmada önerilen dinamik BDÖ tasarım ve geliştirme modeli gösterilmiştir.



Şekil 4. Önerilen Dinamik BDÖ Tasarım ve Geliştirme Modeli

Önerilen BDÖ tasarım modeli, sistematik ve doğrusal BDÖ tasarım modelinin dört aşamasını alarak ÖEK'nin temelini oluşturan bilgi nesneleri, etkinlik grupları ve etkinlik tasarım ortamı kavramlarıyla bütünleştirmiştir. Tasarım modelinin merkezini ve çekirdeğini ÖEK'nin bileşenleri oluşturmaktadır (Bilgiyi ve stratejileri tanımlayan bileşenler). Tasarım aşamaları doğrusal yapıdan çıkarılarak döngüsel bir yapıya kavuşturulmuştur. Aşamalar birbirini izlemekte ve sürekli etkileşim halindedirler. Öğretim ihtiyaçlarına bağlı olarak herhangi bir

aşamadaki yapısal değişiklik, kendisinden sonra veya önceki aşamaya doğrudan yansıtılabilmektedir. Bu model ile BDÖ tasarım aşamalarına getirildiği düşünülen yenilikler şunlardır:

1. *Analiz Aşaması:*

a.Konu içerikleri bilgi nesnelere biçiminde bütünleşik olarak bilgi tabanında yapılandırılmıştır. Bu bilgi tabanında, bilgi nesnelere arasında hiyerarşik bir yapıdan farklı olarak çeşitli yapısal ve fonksiyonel ilişkiler tanımlanmaktadır. Böylece öğretilecek konuların, kavramsal olarak çok boyutlu bir yapıda analizine imkan tanınmaktadır.

b.Bilgi nesnelere yapılan içerik ve öğretim analizinin tasarım çıktıları, nesneye yönelik (object oriented) programlama tekniğine uygun bir format ortaya koymaktadır. Bunun sonucu olarak, öğretim analizi ile tasarım, geliştirme ve değerlendirme aşamaları arasında yapısal ve işlevsel bir bağ oluşturulmuş olmaktadır.

c.Bilgi nesnelere dayalı içerik analiziyle, ürün ve süreç değerlendirilmesi sonucunda ortaya çıkan düzenleme ihtiyaçlarına daha az kaynak kullanılarak cevap verilebilmektedir. Sonradan yapılan bu düzenlemeler doğrudan tasarım ve geliştirme aşamalarına da aynen yansıtılabilmektedir.

2. *Tasarım Aşaması:*

a.ÖEK'nda etkinlik grubu kavramı ve etkinlikleri detaylandırılması, çoklu ortam açısından farklı ve zengin öğretim etkinliklerinin tasarımına olanak sağlamaktadır.

b.Bilgi nesne tabanlı tasarımda bilginin kendisi ve sunum biçim birbirinden ayrılmıştır. Öğretim durumları, bireysel özellikler vb. öğretim ihtiyaçları bilginin sunumunu belirlemektedir. Bunun sonucu olarak dinamik ve gerçek zamanlı (real-time) ihtiyaçlara yönelik olarak bilginin görünümü farklı biçimlerde tasarlanabilmektedir.

3. *Geliştirme Aşaması:*

a.Bilgi nesnelere dayalı bir tasarım aynı zamanda nesneye yönelik programlama tekniği arasında doğal bir köprü oluşturmaktadır (Uysal, 2008).

b.Önerilen bu modeldeki BDÖ tasarımı, yazılımı geliştiren kişiler için aynı zamanda bir yol haritası görevini görmektedir. Analiz ve tasarım aşamasındaki bilgi nesnesi biçimindeki yapılar, yazılım mühendisliği kapsamında sınıf (class) ve nesne (object) olarak kodlanabilmektedir.

4. *Değerlendirme Aşaması:*

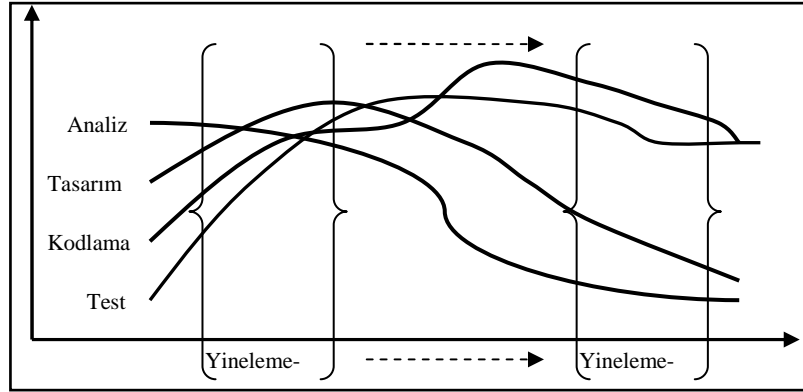
Bilgi nesnelere, etkinlik grupları ve ÖEK'nın temel yaklaşımını esas olan bu tasarım modeli, ürün ve süreç değerlendirme aşamalarına da önemli katkılarda bulunmaktadır. Değerlendirme sonucunda elde edilen dönütler, modüler ve çok boyutlu tasarım aşamalarına aynen yansıtılabilmekte, içerik yapısı, bilginin sunum sıralamaları ile öğretim stratejileri kolayca değiştirilebilmektedir.

## **BDÖ Yazılım Geliştirme Süreçlerine Dinamik Yaklaşım**

Literatürde yaygın olarak kullanılan değişik yazılım geliştirme standartları bulunmaktadır. Bunlardan birisi de IEEE/EIA-12207 standardının Şelale Modeline (Waterfall) uyarlanmış biçimidir. Bu modelde her bir aşamada detaylı

analiz ve tasarım adımları bulunmakta, proje genellikle kalabalık bir ekip tarafından yürütülmektedir. Bu standart, gerekli süreçleri ana hatlarıyla ele alarak genel çerçeveyi belirlemiş, ayrıntılı ve somut uygulama adımlarını ihtiyaç makamına bırakmıştır. Bu tür standartların en belirgin özelliği ise birçok sayfadan oluşan proje dokümanlarını gerektirmesidir. Ancak bilindiği gibi, uzun süreli ve kapsamlı yazılım projelerinde gereksinimleri başlangıçtan itibaren öngörebilmek her zaman için mümkün olmayabilmektedir. Test ve değerlendirme sonrasında ortaya çıkabilecek ihtiyaçların projeye uyarlanması çoğu zaman maliyeti arttırdığı gibi yazılımlarda köklü değişiklikleri de gerektirebilmektedir. Esnek, hızlı kod üretilebilen, ihtiyaçların tam ve zamanında karşılandığı ve ekip çalışmasına dayalı yazılım geliştirme yöntemlerine olan ihtiyaç bizi daha dinamik yöntemlere doğru yönlendirmektedir.

Çevik Yazılım Geliştirme (Agile Software Development) (ÇYG) yöntemleri, fazlasıyla kurallara bağlı, uygulama açısından zor, bürokratik ve sıkıcı yazılım geliştirme yöntemlerine tepki akımı olarak ortaya çıkmıştır. Bu akımın öncüleri 2001 yılında bir araya gelerek hızlı, basit ve müşteri odaklı yazılım geliştirmenin önemini vurgulayan Çevik Yazılım Yaklaşımı adlı manifestoyu yayınlamıştır. Son yıllarda yinelemeli yöntemlerle ele alınan bu yaklaşım, özellikle orta ölçekli yazılım geliştirme şirketleri tarafından yaygın olarak kullanılmaktadır. Scrum, Extreme Programming-XP, Agile Modeling, Agile Unified Process ve Feature Driven Development gibi metodolojiler belli başlı ÇYG yöntemleri arasında sayılmaktadır. ÇYG, yinelemeli yazılım geliştirme yöntemlerini esas alan, sık aralıklarla, parça ve artırımsal yazılım geliştirmeyi teşvik eden bir metodolojidir. Esnek ve kullanışlı yazılım geliştirmeyi hedefleyen bu yaklaşım, ayrıntılı modelleme ve tasarım teknikleri yerine, bunların ihtiyaç ve değişiklik isteklerine nasıl uygulanması gerektiğiyle ilgili çerçeveyi belirlemektedir. İhtiyaç ve değişiklik taleplerinin tam olarak öngörülemediği, uzun süreli ve detaylı planlamanın yapılamadığı, süreçlerle ilgili zaman kısıtlamasının olduğu ve ekip çalışmasının gerektiği durumlarda uygun bir yöntemdir.



Şekil 5. ÇYG'de Yinelemeli Yazılım Projesi Geliştirme



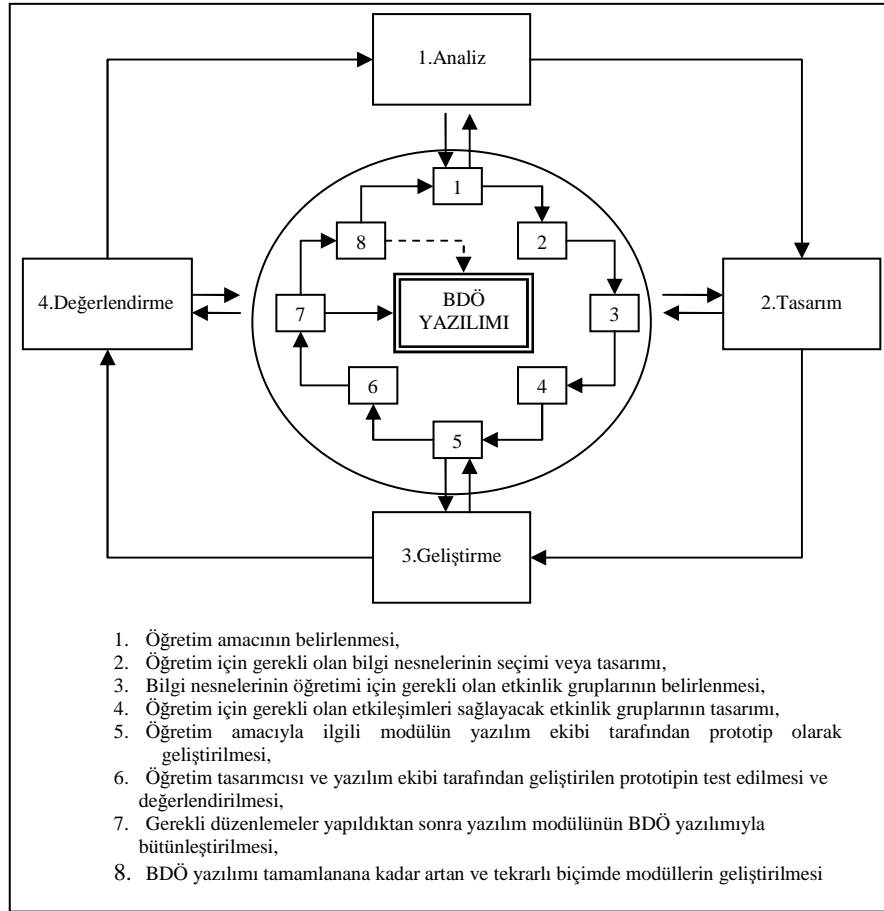
Bu çalışmada yazılım geliştirme süreçleri için benimsenen dinamik yaklaşımın temelini ÇYG yöntemi oluşturmaktadır. Genel olarak ÇYG yöntemlerinde uzun vadeli, detaylı planlama ve dokümantasyon etkinlikleri yerine sonraki süreçlerde çıkabilecek değişiklik ihtiyaçlarına cevap verebilen, hızlı programlama ve müşteri memnuniyetini ön plana çıkaran bir yaklaşım izlenmektedir. ÇYG modelinde artan ve tekrarlı bir yazılım geliştirme aşamalarından geçilmektedir. Uzun ve kapsamlı analiz ve planlama süreçleri yerini, karşılıklı işbirliği ve takım çalışmasına dayalı, hızlı ve kaliteli kod geliştirmeyi esas alan, üç veya dört haftalık kısa tekrarlara (*iteration*) bırakmıştır. Her bir tekrar, ihtiyaç ve fonksiyonel analiz sonucu ortaya konulan iş kuralı ve hedefleri kapsamakta, kendi içerisinde *analiz, tasarım, kodlama, test* ve *kabul* gibi yazılım geliştirme aşamalarını içermektedir (Şekil 5). Böylece projeye ilgili riskler en aza indirgenmekte, müşterinin onayı alınarak yeni değişiklik istekleri anında projeye yansıtılabilmektedir. ÇYG modeli; test edilmiş, müşteri onayı alınmış ve en önemlisi kısa sürede çalışan yazılım modüllerini ortaya koymak amacıyla yazılım geliştirme süreçlerini ele almaktadır.

ÇYG eksenli yazılım geliştirme, aynı zamanda çoklu ortam araçlarını içeren BDÖ yazılımlarının doğasına da uygun bir modeldir. Geliştirilen modül ve prototiplerle, analiz ve tasarım öğelerinin görselleştirilerek başlangıçtan itibaren yazılıma aktarılmasına imkan tanınmış olmaktadır. Bu çalışmada önerilen modelde, öğretim tasarımcısı ve yazılım ekibinin tasarım ve geliştirme aşamalarındaki karar alma süreçlerine birlikte katılımları, dinamik yaklaşımın çerçevesini çizmektedir. Öğretim analizi sonucunda öğretim amaçları belirlenmekte, her öğretim amacıyla ilgili yazılım modülleri ÇYG'deki gibi tekrarlı biçimde gerçekleştirilmektedir. Birinci adımda, öğretim amacının belirlenmektedir. İkinci adımda gerekli olan bilgi nesnelere bilgi tabanından seçilmekte veya yoksa yeniden tasarlanmaktadır. Üçüncü adımda bilgi nesnelere öğretimi için gerekli olan etkinlik grupları belirlenmektedir. Dördüncü adımda etkileşimleri sağlayacak etkinlik grupları tasarlanmaktadır. Beşinci adımda öğretim amacıyla ilgili modül yazılım ekibi tarafından prototip olarak geliştirilmektedir. Altıncı adımda öğretim tasarımcısı ve yazılım ekibi bir araya gelerek geliştirilen modülü değerlendirmektedir. Yedinci adımda, gerekli düzenlemeler yapıldıktan sonra geliştirilen modül BDÖ yazılımıyla bütünleştirilmektedir. Sekizinci adımda dögüsel ve tekrarlı sürece, BDÖ yazılımı tamamlanana kadar devam edilmektedir (Şekil 6).

## SONUÇ

Bilgisayar ve teknolojiye gelişmeler farklı BDÖ ortamlarının tasarımına imkân tanırken alandaki çalışmalar da bu doğrultuda gelişmektedir. BDÖ yazılımları gerçekleşmeleri zaman alan, yoğun kaynak kullanımı ve uzmanlığı gerektiren öğretim sistemleridir. Geliştirilen bütün öğretim sistemlerinde olduğu gibi BDÖ yazılımları da öğretim tasarımı bakımından kabul gören kuram veya tasarım yöntemine dayandırılmalıdır. Ancak, yazılımların öğretimsel açıdan kalitesini sağlayan en önemli etken, temel aldığı öğretim tasarım modelidir. Bir öğretim

yazılımı, yazılım mühendisliği ölçütleri açısından ne kadar kaliteli olursa olsun, sistematik ve dinamik öğretim tasarım aşamalarından geçirilmedikçe öğretimsel kalitesi hakkında akıllarda soru işareti bırakacaktır. Bu çalışmada, ÖEK'nın temel yapısı esas alınarak dinamik bir BDÖ yazılımı tasarım ve geliştirme modeli önerilmiştir. ÇYG modeli ve ÖEK'nın temel tasarım kavramları, yazılım projesi geliştirme kapsamında bütünleştirilmiştir. Yapılan araştırmalar ve teknolojiye gelişmeler, BDÖ alanında kaliteli, zengin ve öğretim ihtiyaçlarına dinamik biçimde cevap verebilen BDÖ yazılımlarına olan artan ihtiyacı ortaya koymaktadır. Uysal (2009a) ve Uysal (2009b) de belirttiği gibi bilgi nesnelere dayalı bir tasarım aynı zamanda yazılım mühendisliği disiplini ile öğretim tasarımı arasında doğrudan bir köprü oluşturmaktadır. Bu çalışmada önerilen tasarım modeli ile bilgi nesnelere kapsayan farklı nitelikteki çalışmaların kaliteli öğretim yazılımlarının geliştirilmesine yönelik önemli katkılar sağlayabileceği düşünülmektedir.



Şekil 6. BDÖ Yazılımı Tasarım ve Geliştirme Süreçleri

**KAYNAKLAR**

- Alessi, M.S. and Trollip S.R. (2001). Multimedia for Learning, Methods and Development *Massachusetts USA. Pearson Education Company*
- Anderson, A.T. (1999). Rethinking Authoring Tools : A Design for Standarts-Based Instructional Components. Utah State University, USA (Yayınlanmamış Doktora Tezi, UMI Numarası : 9926541)
- Boyle, T. (1997) .Design For Multimedia Learning . *Priontice Hall Europe*
- Gagne R.M. and Briggs L. and Walter W. (1992). Principles of Instructional Design Lawrence Erlbaum Associates, Publishers. New Jersey.
- Gagne, R.M. and Wagner, W. (1988). Designing Computer-Aided Instruction (ed. David H. Jonassen). Instructional Designs for Microcomputer Courseware. Lawrence Erlbaum Associates, Publishers. New Jersey USA.
- IEEE Std 1219 (1998) IEEE Standard for Software Maintenance.
- IEEE Std 1490 (2003) IEEE Guide, Adoption of PMI Standard, A Guide to the Project Management Body of Knowledge.
- IEEE Std 1074 (2006) IEEE Standard for Developing a Software Project Life Cycle Process.
- IEEE/EIA-12207 (2008) IEEE Standard for Software Life Cycle Processes-Implementation Considerations.
- IEEE Std 1016 (2009) IEEE Standard for Information Technology-Systems Design-Software Design Descriptions.
- Jonassen,H.D.(1988). Instructional Design for Microcomputer Courseware. *New Jersey Usa. Lea Inc. Publishers.*
- Koch, A.S.(2005). Agile Software Development: Evaluating the Methods for your Organization. *Artech House,Inc..*
- Landa, L. N. (1998) Landamatics Instructional Design Theory and Methodology for Teaching General Methods of Thinking. <http://eric.ed.gov/> adresinden 24 Mart 2008 tarihinde alınmıştır.
- Larman, C.(2003). Agile and Iterative Development: A Manager's Guide. *Addison Wesley.*
- Lawrence, L. Leff (2004) Landamatics in Teaching Computer Programming. *Journal of Computer Science Education* January 2004
- Merrill, M.D.(1987). A Lesson Based on the Component Display Theory. (ed. M.Charles). *Instructional Theories in Action.Reigeluth, New Jersey USA. Lawrence Erlbaum Associates, Publishers*
- Merrill, M.D.(1991)(A) Instructional Transaction Theory: Second Generation Instructional Design Resarch Project. [Http://Cito.Byuh.Edu/Merrill](http://Cito.Byuh.Edu/Merrill) adresinden 22 Şubat 2007 tarihinde alınmıştır.
- Merrill, M.D.(1991)(B). Instructional Transaction Theory: Classes of Transactions <http://cito.byuh.edu/merrill/text/papers.htm> adresinden 19 Ocak 2006 tarihinde alınmıştır.
- Merrill, M.D. and Li Z. And Jones M (1991)(C). Second Generation Instructional Design. <http://cito.byuh.edu/nerrill/text/papers.htm> adresinden 22 Şubat 2007 tarihinde alınmıştır.
- Merrill, M.D.(1992)(A). Instructional Transaction Theory: Knowledge Relationships Among Processes, Entities, and Activities <http://cito.byuh.edu/merrill/text/papers.htm> adresinden 19 Ocak 2006 tarihinde alınmıştır.
- Merrill, M.D. and Li Z. and Jones M.(1992)(B). Instructional Transaction Shells: Responsibilities, Methods, and Parameters. <http://cito.byuh.edu/merrill/text/papers.htm> adresinden 22 Ocak 2006 tarihinde alınmıştır.

- Merrill L, M.D. (1996). Instructional Transaction Theory: an Instructional Design Model Based on Knowledge Objects. <http://cito.byuh.edu/merril> adresinden 24 aralık 2006 tarihinde alınmıştır.
- Merrill, M.D.(1998). Knowledge Objects. [http://cito byuh.edu/merril](http://cito.byuh.edu/merril) adresinden 23 ağustos 2005 tarihinde alınmıştır.
- Merrill, M.D. (2000). Knowledge Objects and Mental-Models. <http://cito.byuh.edu/merrill/text/papers.htm> adresinden 16 Eylül 2005 tarihinde alınmıştır.
- Merrill, M.D. (2001). Components of Instruction: Toward a Theoretical Tool for Instructional Design. *Instructional Science* 2001, 29, S:291-310
- Paetsch, F. (2003). Requirements Engineering and Agile Software Development. <http://ase.cpsc.ucalgary.ca> adresinden 10 Ekim 2009 tarihinde alınmıştır.
- Pressman, S.Roger (1992). *Software Engineering. McGaw-Hill, Inc. New York USA*
- Reigeluth, M.Charles (1987). *Instructional Theories In Action. Lawrence Erlbaum Associates, Publishers. New Jersey USA*
- Reigeluth, M.Charles (1999). *Instructional Design Theories and Models. Volume II Lawrence Erlbaum Associates, Publishers. New Jersey USA.*
- Uysal, M.P.(2008). Öğretim Etkinlikleri Kuramına Göre Tasarlanan Öğretim Yazılımı ve Uyarlanabilir Alıştırma Yazılımının Akademik Başarıya Etkisi. Yayınlanmamış Doktora Tezi. Gazi Üniversitesi Eğitim Bilimleri Enstitüsü
- Uysal, M.P.(2009)(A). Öğretim Etkinlikleri Kuramı ile Yaşantısal Öğrenme Modelinin Bütünleştirildiği Zeki Öğretim Sistemi Tasarımına Yönelik Bir Model Önerisi. 3. Bilgisayar ve Öğretim Teknolojileri Sempozyumu. Trabzon
- Uysal, M.P.(2009)(B). Bilgi Nesnesi Tabanlı ve Öğrenme Stilllerine Uyarlanabilen Alıştırma Yazılımının Akademik Başarıya Etkisi. 3. Bilgisayar ve Öğretim Teknolojileri Sempozyumu. Trabzon
- Yalın, H. İ. (2004). Öğretim Teknolojileri ve Materyal Geliştirme. *Nobel Yayıncılık. Ankara*
- Zwart, W.Jaap (1992) *Instructional Transaction Theory Applied to Computer Simulations. ERIC No: ED352945*

### SUMMARY

Design and development of Computer Aided Instructional (CAI) softwares requires time-consuming and labor-intensive processes. Regardless of the software engineering method that is chosen, any CAI software must go through the basic three phases, such as project definition, development, and maintenance. In addition to that, software project development is discipline which integrates a number of methods, tools, and procedures for the development of computer software. Of all the factors, software development procedures and instructional development methods are the most determinant of the life cycle costs. What makes a CAI software similar to a human tutor is its powerful ability to adapt the presentations of course contents to meet the needs of a learner or a curriculum. Nature of the human behaviors and learning needs differentiate the CAI softwares from the other types of softwares. CAI softwares are also instructional systems and require a set of procedures for systematically designing and developing instructional materials. One of the factors which determines the quality of these softwares is that they undergo appropriate instructional design processes. The other factor is the their capabilities of response to the needs

during or after instruction. An Instructional Design Theory (IDT), prescribes appropriate instructional strategies to acquire knowledge. In general, four phases of instruction should occur for an effective and efficient learning. These are presenting knowledge, guiding the learner, practising, and assessing the learning. However contemporary IDTs and their applications require considerable effort, they do not prescribe efficient CAI software development methods. Merrill classifies the IDTs, Component Display Theory, Landamatics Instructional Design Theory, Information Processing Theory, Structural Analysis etc., as the First Generation Instructional Design Theories (FGIDTs). He claims that FGIDTs share a number of limitations: First, they are closed systems which are being not able to accommodate new knowledge as it is needed. Second, FGIDTs have independent instructional development phases which provide no means for integration. Third, their resulting instruction is often passive and lacks effective bidirectional interactions. Finally, FGIDTs predates the high-tech computer and technology-based delivery systems. Regarding the limitations of FGIDTs, Instructional Transaction Theory (ITT) attempted to automate the instructional design processes to accommodate the contemporary CAI software design. Descriptive theory of knowledge, descriptive theory of strategy, and the prescriptive theory of instructional design are the principal components of ITT. ITT adopted computer program assumption and it is an algorithmic instructional system. Knowledge is presented as data. Course contents, being the components of knowledge, are processed or displayed by the algorithms built into the instructional system. A knowledge object is a way to analyze and describe the subject matter contents which are included in an instructional system design. It provides necessary means for identifying knowledge components and organizing content resources (text, audio, video etc.) as a knowledge base. Entity, property, part, activity, and a process are the knowledge components of a knowledge object. A change in a property value of an entity as a result of a learner's act on a controller is an activity. The rules which change the property values of an associated entity or entities are defined as a process. PEAnets are predetermined relationships, which are formed by processes, entities, and activities interacting in particular ways. In ITT, all of the learning interactions, necessary to acquire a knowledge or skill, are defined as an Instructional Transaction. An instructional algorithm, which is promoting an instructional transaction, operates on a set of knowledge objects for a learner to acquire the instructional goal. Furthermore, Merrill suggested a number of classes of transactions which are suited for a particular kind of an instructional goal. In literature, CAI softwares are generally designed, implemented and evaluated in a structured and linear approach in terms of instructional design view. Analysis, design, development and evaluation phases are sharply marked out. Each phase provides the input to the next development phase. In this traditional design approach, the major drawback is that it is too rigid. The instructional and software refinements after or during implementation have to be processed in a linear fashion. In this study, ITT, which is thought to be suited for the nature of CAI, is initially discussed. Then, a design model is proposed, which adopted a dynamic approach for the both instructional and software development processes. Building blocks of ITT,

knowledge objects, transactions and transaction shells constituted the core of the proposed model. The instructional and the software design phases are integrated with each other. Also, in view of the Agile Software Development (ASD) paradigm, iterative and incremental prototyping is imported to the proposed model. ASD enables frequent adaptation to the needs and allows high-quality software development in a disciplined project management environment. It assumes working software as the primary measure of a progress while promoting adaptability, collaboration, and teamwork throughout the life cycle of the software project. Iterations are delivered within short time frames lasting from a couple of weeks to a month. Our model draws a conceptual framework for both instructional and software design issues on a common ground. Instructional design phases are combined and then transformed to a cyclic and iterative model rather than a linear one. Each iteration and its development steps comprise the kernel of the cyclic model. Instructional and software development tasks are broken into small increments, which are made up of eight steps using knowledge objects and ITT transactions for achieving an instructional goal. The 1st Step: Defining the instructional goal. The 2nd Step: Selecting and designing knowledge objects. The 3rd Step: Defining the instructional transactions required for knowledge objects. The 4th Step: Designing the instructional transactions. 5th Step: Development of the software module (iteration). 6th Step: Test and evaluation of the iteration by both instructional designer and the software development team. 7th Step: At the end of each iteration, the developed software module is integrated with the body of the incomplete system. 8th Step: Proceeding the iterative and incremental processes until the completion of the CAI software. With this proposed design and development model, main aim is to minimize overall risk and let the software project adapt to the changes quickly through the both instructional designer's and software developer's approval. Iterative design and development, coupled with the building blocks of the ITT, would be said to have important contributions to the designed and development of the CAI softwares.