

TBD 8. ULUSAL
BİLİŞİM
KURULTAYI

27 - 29 Eylül 1990
ANKARA

BİLİŞİM' 90
Bildiriler

TÜRKİYE BİLİŞİM DERNEĞİ

Türkiye Bilişim Derneği Yayınları

İÇİNDEKİLER

1. OTURUM

OTURUM KONUSU : **BİLGİSAYAR DESTEKLİ TASARIM**
OTURUM BAŞKANI : CENGİZ ULTAV

1. *Akıllı Bilgisayar Destekli Tasarım* 11
Y. Doç. Dr. Seyit GÜLER , Ege Üniv. Bilgisayar Müh. BİM.
2. *Bilgisayar Destekli Tasarım için Kuramsal Bir Çerçeve* 17
Varol AKMAN, Sibel ADALI, Bilkent Üniv. Bilgisayar ve Enf. Müh. Böl.
3. *EGS : Etkileşimli Grafik Sistemi* 25
F. BÜYÜKÖKTEN , O. IŞIKLI, U. HALICI, E. HALICI, HALSOFT A.G.
4. *Üretim İzlenemesi İçin Kural Dayanaklı Bir Karar Destek Sistemi* 33
Semra TUNALI, Dokuz E. Ü. Müh.Mim.Fak.En.Müh.Böl.
Emrah ORHUN, E.Ü. Müh.Fak.Bilgisayar Müh.Böl.

2. OTURUM

OTURUM KONUSU : **BİLGİSAYAR EĞİTİMİ**
OTURUM BAŞKANI : Doç. Dr. ÖNDER ÖZKAZANÇ
Anadolu Üniv. BİM Daire Başkanı.

1. *Programa Dilleri Eğitimi İçin Bir Akıllı BDE Yazılımı* 47
Bircan ŞİMŞEK, BİLTEK A.Ş.
H. Altay GÜVENİR , Bilkent Üniversitesi
2. *BDE' nin Başarıya Ulaşmasını Etkileyen Faktörler* 58
Metin KÖKSAL , Hülya YAVUZ , IBM BDE Araş. Geliştirme Merkezi
3. *Bilgisayar Destekli Öğretim İzlenmelerinin Geliştirilmesi* 66
Ahmet KAŞLI , Emrah ORHUN , Ege Üniv. Bilgisayar Müh. Böl.
4. *Hypertext ve Öğrenme* 75
Birgül EKELİ, Berkal BAYKAL , Emrah ORHUN , E.Ü. Bilgisayar Müh.Böl.

3. OTURUM

OTURUM KONUSU : **UYGULAMALAR**
OTURUM BAŞKANI : İ. SEMİH ARSLANOĞLU
Digital Equipment Türkiye Genel Müdürü

1. *Merkezi Bakım Ünitesi Uygulaması* 83
Hakan UYSAL, Çağatay ASLAN, Murat ILIMAN , ORTAŞ A.Ş.
Özer M. SAGER

Programlama Dilleri Eğitimi için bir Akıllı BDE Yazılımı

Bircan Şimşek
Biltek
Ankara

H. Altay Güvenir
Bilkent Üniversitesi
Ankara

ÖZET

Bu bildiride bilgisayar programlama dilleri eğitiminde kullanılmak üzere hazırlanmış olan bir akıllı BDE yazılımı (ProDil) anlatılmaktadır. Akıllı BDE yazılımlarının önemli özelliklerinden biri olan, konuya ve öğrencinin gereksinimlerine uygun olarak çok sayıda ve çeşitli örnek ve soru üretilmesi üzerinde durulmuştur. Örnek ve soru üretimine elverişli bir bilgi gösterim tekniği olarak geliştirilen örüntü kullanımı anlatılmaktadır.

Giriş

İçinde bulunduğumuz yüzyılın sonlarında bilgisayar olgusu, yaşamımızın hemen her noktasında ortaya çıkmakta ve yardımlarını sunmaktadır. Bilgisayarların yaşamımız üzerindeki etkisi son yıllarda daha da hızlı bir şekilde artmış ve 1990 yılı ülkemizde "Bilgi Yılı" olarak ilan edilmiştir. Bilgisayar en önemli etkilerinden birini de eğitim alanında gösterecektir. Eğitimde bilgisayarlardan yararlanılmasıyla ortaya çıkan Bilgisayar Destekli Eğitim (BDE), çeşitli açılardan kalitenin yükseltilmesinde önemli olanaklar sağlayacaktır.

Akıllı BDE Sistemleri

Klasik sistemde öğrenciye verilecek bilgi iki temel kaynaktan gelmektedir; bunlar kitap ve öğretmendir. Bilgisayar destekli eğitimde bunlara bir üçüncüsü, bilgisayar (donanım ve yazılım) eklenmiştir. Bu üç bilgi kaynağının özelliklerini şu şekilde özetleyebiliriz: Kitap içerdiği bilgiler sabit olan, buna karşılık her an erişilebilen bir kaynaktır. Kitabın içindeki bilgiler, konuyu anlatmak için verdiği örnekler değişmez. Ayrıca kitap size genellikle öğrettiği konuyu anlayıp anlayamadığınızı da söyleyemez. Buna karşılık, bir öğretmen anlattığı konuyu anlayıp anlamadığınızı çeşitli sorular yöneltmek belirleyebilir ve sizin gereksinimlerinize göre konuyu daha farklı bir şekilde ve yeni örneklerle tekrar aktarabilir. Özet olarak, eğer öğrenci konuyu anlayabilecek seviyede ise öğretmen kesinlikle bunu başaracaktır (ancak her öğrenciye bir öğretmen vermek günümüzde hiçbir ülkenin olanakları içinde değildir). Kitap ile öğretmen arasındaki bu fark öğretmenin, bir insan olduğundan dolayı bir zekaya sahip olması, buna karşılık, kitabın ise bu zekadan yoksun olmasından kaynaklanmaktadır. Bilgisayar ise bu ölçekte kitap ile öğretmen arasında yer almaktadır. Bilgisayar destekli eğitimin başarılı olabilmesi için, diğer bir deyişle kitaptan daha üstün olabilmesi için, bir öğretmene benzer şekilde zeki (ya da akıllı) olması zorunludur. Bu özelliklere sahip olan bilgisayar sistemleri, akıllı BDE sistemleri, ya da donanım sabit olduğuna göre *akıllı BDE yazılımları* olarak adlandırılacaktır.

Bir akıllı BDE yazılımı, kitaptan farklı olarak çeşitli özelliklere sahip olmalıdır. Konuyu genel çizgileriyle aktardıktan sonra bir öğretmen gibi çok sayıda ve farklı zamanlarda farklı örnekler verebilmelidir. Örnek sayısı ne kadar çok olursa (sıkıcı olmamak koşuluyla) öğrenci bu örneklerden okadar iyi genelleme yapabilecektir. Konunun öğrenci tarafından anlaşılıp anlaşılmadığı, konu ile ilgili olarak üretilen sorularla belirlenecektir. Burada iki nokta çok önemlidir. Birincisi, üretilen soruların her seferinde farklı olması gereğidir. Bir öğretmen her sınavda farklı sorular sormaya dikkat eder (bir kitaptaki sorular ise hep aynıdır). İkinci önemli nokta ise öğrencinin yanıtının doğru bir şekilde değerlendirilmesidir. Öğrencinin yanıtına verilecek sadece "doğru" ya da "yanlış" şeklin-

de ki bir karşılığın öğretici niteliği oldukça azdır. Bir öğretmen, öğrencinin yanıtı eğer yanlış ise, yanlışın hangi konuların yeterince anlaşılmasından kaynaklandığını belirleyerek bu konular üzerine tekrar eğilecektir. Bir akıllı BDE yazılımı da benzer şekilde öğrenci yanıtını da analiz edebilmelidir.

Belirli bir konuya ait farklı örnekler ve sorular üretilebilmesi ve sorulara verilen yanıtların analiz edilebilmesi için BDE yazılımının öğretilen konuyu anlaması, diğer bir deyişle, ne öğrettiğini bilmesi gerekmektedir. Bu da öğretilecek bilginin uygun bir biçimde kodlanmasıyla sağlanabilir. Çeşitli çıkarımlar yapılabilmesine elverişli bilgi gösterim tekniklerinin geliştirilmesi yapay anlayış (*Artificial Intelligence*) alanında yapılan çalışmaların önemli bir bölümünü oluşturmaktadır. Bu teknikler kabaca iki grupta toplanabilir.

1. Tanımsal (Declarative) yöntemler
2. Yordamsal (Procedural) yöntemler

Tanımsal yöntemlerde bilginin büyük bir kesimi, durumların statik topluluğu olarak gösterilir. Bu yöntemde ayrıca bilgiyi işleyecek genel yordamların küçük bir kümesi de bulunabilir. Tanımsal yöntemler, anlamsal ağlar (semantic nets), kavramsal bağımlılık (conceptual dependency), çerçeveler (frames), 'scripts', üretim kuralları (production rules) olarak gruplandırılabilir. Yordamsal yöntemlerde bilgi, onu kullanan yordamlar olarak gösterilir. Yordamsal gösterim yaklaşımları, yordamların çağırılma yöntemlerinin seçimine ve kontrol yapılarında kullanılan biçimlere göre sınıflandırılırlar. Her iki yöntemde uygulamanın türüne bağlı olarak birbirine göre üstünlükleri vardır. ProDil için geliştirilmiş olan bilgi gösterim tekniği, *örüntü*, tanımsal gösterim yöntemi olarak sınıflandırılabilir.

ProDil

ProDil programlama dilleri eğitimi için hazırlanmış bir BDE yazılımıdır. Gerçekleştirilen sistemde programlama mantığı kazandırmadan daha çok programlama dilleri öğretimi amaçlanmıştır. Öğretilecek dile ilişkin bilgi tabanının oluşturulması ve bu bilgiyi kullanarak eğitim veren modüller tamamen ayrılmıştır. Başka bir deyişle, sistemde öğretilecek dile özel kodlanmış bir kesim yoktur. Buradaki amaç, tek bir eğitim programının, çeşitli programlama dillerinin eğitiminde kullanılabilmesini sağlamaktır. Bu nedenle ProDil bir *ders hazırlama sistemi* (authoring system) dir. Ancak, bilgi gösterim yöntemi geliştirilmesinde yalnızca programlama dilleri eğitimi göz önünde tutulmuştur. ProDil, C programlama dili örnek alınarak bilgi tabanı oluşturulmuş ve denenmiştir. Bu bildiri- de verilen örnekler de C dili için oluşturulmuştur.

Genel yapı olarak ProDil, bir birinden bağımsız çalışan üç ayrı modülden (yazılımdan) oluşmaktadır. Bunlar;

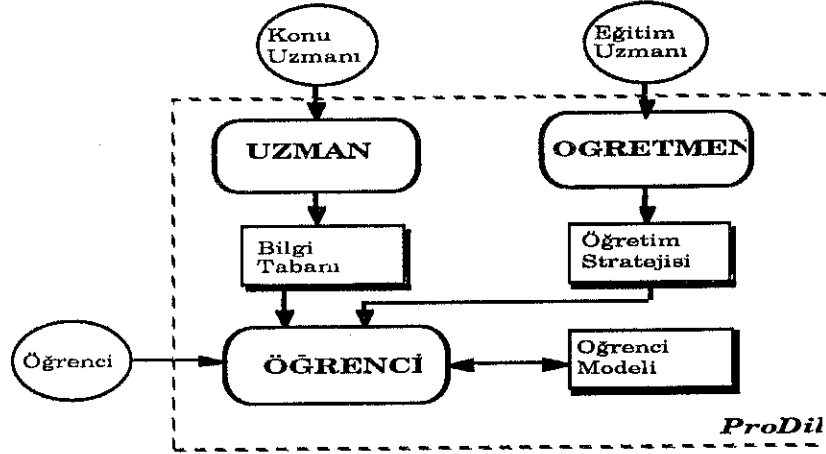
1. UZMAN modülü
2. ÖĞRETMEN modülü
3. ÖĞRENCİ modülüdür.

Bu modüller ve aralarındaki ilişkiler şekil 1 de gösterilmiştir.

UZMAN Modülü

Bu yazılım konunun uzmanı tarafından bilgi tabanının oluşturulmasında kullanılır. Bilgi tabanında, öğretilecek programlama dilinin yapısıyla ilgili bilgiler yer alır. Akıllı bir eğitim sisteminde, bu bilgiler kullanılarak, konunun genel olarak anlatımı, örnek ve soruların sistem tarafından üretilebilmesi ve sistemin kendi doğrularını öğrencinin yanıtları ile karşılaştırarak, hataların nerede olduğunun ve neden kaynaklandığının çözümlenmesi gerekmektedir. Bunu sağlayabilmek için gerçekleştirilen sistemde, programlama dillerinin özellikleri incelenmiş ve bunlara ilişkin anlamsal ve sözdizimsel bilgilerin düzenleniş biçimi olarak bir çeşit kalıp olan *örüntü* gösterim biçimi geliştirilmiştir. Bir *örüntü* genel olarak anlamsal bir işlevin gerekli sözdizim kurallarına uygun olarak gösterimi şeklinde tanımlanabilir. ProDil'de bir programlama dilinin öğretilmesinde yararlı olacak

çeşitli işlevler birer örüntü biçiminde ifade edilerek, farklı örnek ve soruların sistem tarafından üretilmesi ve öğrencinin yanıtının akıllı bir şekilde analiz edilebilmesi sağlanmıştır.



Şekil 1. ProDil'in genel yapısı.

UZMAN Yazılımı tarafından özel bir yapı editörü ile oluşturulması sağlanan ve bilgi tabanında yer aldığı öngörülen bilgiler şunlardır:

- Öğretilecek konu başlıkları ve alt başlıkları
- Her konuya ilişkin açıklama (metin) bilgileri
- Konulara ilişkin örnek ve soruların üretilmesini ve yanıtların değerlendirilmesini sağlayacak anlamlı sözdizim örüntü tanımları
- Her anlamlı sözdizim örüntü tanımına ilişkin:
 - soru ve örnek cümleleri
 - değişken ve değişmezlerin uyması gereken sınırlayıcıların tanım bilgileri
 - ifade ettikleri anlam
 - altgrup örüntülerinin bütün içindeki önemini ifade eden puan değeri
 - ilgili olduğu konu bağlantıları
- Konuya ilişkin özel sorular ve ilgili bilgiler
- Anlamsal olarak eşdeğer ifadeler
- Ekranda görünmeyen (white space) karakterleri ve açıklama bilgilerini işaret eden karakterlerin tanım bilgisi

UZMAN Yazılımı, konu başlıklarının belirlenmesinden sonra her konuya ilişkin açıklama, anlamlı sözdizim örüntüleri ve özel örnek ve soruların tanımlanmasını sağlar.

Açıklama bilgileri konu hakkındaki çeşitli tanımları, kavramları ifade eder. Bu bilgi öğretmenin, bir konuyu anlatırken örnek ve soruları vermeden önce yaptığı açıklayıcı bilgilere karşı düşer. Aşağıda bu bilgilerin oluşturulduğu bir ekran görünümü verilmiştir.

```

----- for deyimi -----
for deyimi, C dilinin döngü denetim yapılarından biridir.
Sözdizimi:      for (expr1 ; expr2 ; expr3)
                  statement
şeklindedir. 'expr1' ve 'expr3' birer aktarma deyimidir. 'expr2' is koşul
deyimidir ve bu koşul sağlandığı sürece döngü içindeki deyimler
(statement) uygulanır.

```

Anlamli sözdizim örüntü tanımları ve ilgili bilgilerin oluşturulması: Anlamli sözdizim örüntüleri, sistem tarafından üretilecek soru ve örneklerin oluşturulmasında kullanılmaktadır. Aynı örüntü tanımları hem örnek hem de soru üretiminde kullanılabilir. Her örüntü tanımları anlatılmak istenen deyim ya da deyimlerin içerdikleri değişkenler, değişmezler anahtar sözcükler, işaretler ve bunlar arasındaki ilişkileri içerir. Ayrıca bir deyim altdeyimler içeriyor ise bunların altgrup örüntü tanımları olarak tanımlanması, sistemin bir bütünü, parçalarını, anlamlarını ve bunlar hakkındaki diğer bilgileri tanımlamasına olanak sağlayacaktır. Bu nedenle tanımlar ne derece iyi ve parçalarına ayrılmış şekilde oluşturulursa, sistem öğrencinin yanıtlarını o ölçüde ayrıntılı analiz edebilir.

Bir konuya ilişkin birden fazla anlamli sözdizim örüntü tanımları yapılabilir. Bu tanımların her biri konunun çeşitli yönlerini öğrenciye aktarmada birer örnek ve soru stili oluşturacaktır. Anlamli sözdizim örüntülerinin tanımlanmasında bir takım özel simgeler kullanılarak çeşitli kavramların ifade edilmesi sağlanır. Bunlar, @Sn, @Rn, @Cn simgeleridir. @Sn, @Rn, @Cn, konu öğretilirken örnek ve soruların üretilmesi aşamasında, bunlara ilişkin sınırlayıcılar dikkate alınarak belirlenecektir. Bu da her seferinde farklı örnek ve soruların üretilmesini sağlayacaktır. @S, anlamli sözdizim örüntü parçalarını ya da bütünü ifade eder. n ise bu ifadenin numarasını gösterir. Örneğin, '@S0' sıfırıncı yani en üst seviye, '@S1' birinci anlamli sözdizim örüntüsü anlamındadır. @R, anlamli sözdizim örüntülerinde kullanılan değişkenleri simgeler. n değişken numarasıdır. @C, değişmezleri ifade etmek amacı ile kullanılır. n değişmez numarasını belirtir.

Örnek olarak, C programlama dilinin döngü komutu olan **for** deyimine ilişkin belirli bir amaca yönelik örüntü tanımlarını aşağıdaki şekilde yapılabilir. Amacımız, belirli bir değerden başlayarak belirli bir değere kadar sayan döngü ifadesini tanımlamak olsun. **for** deyimini sözdizim olarak;

for (expr1 ; expr2 ; expr3)

şeklinde. Burada *expr1*, anlamsal olarak döngü değişkenine ilkdeğer aktarma işleminin yapıldığı kesimdir. *expr2*, bir koşul deyimini içerir ve koşul sağlandığı sürece döngü içinde yer alan komutlar uygulanır. *expr3*, ise koşul deyimini etkileyecek bir ifade olup, genellikle döngü değişkeninin değeri artırılır ya da eksiltir. Bu durumda örüntü ifadesini;

1. @S0 : for (@S1 ; @S2 ; @S3)
@S1 : @R0 = @C0
@S2 : @R0 < @C1
@S3 : @R0++

şeklinde tanımlayabiliriz. Burada @R0 döngü değişkenini, @C0 ve @C1 sınır değerleri ifade etmektedir. @S1, @S2, ve @S3 altgrup (parça) örüntü tanımlarını oluşturmaktadır. Örüntü parçalarında geçen @Rn ve @Cn değişken ve değişmez simgeleri sistem tarafından, belirtilen sınırlayıcılara uygun bir şekilde rastgele olarak üretilir. Yukarıdaki örüntü tanımında @S0 **for** deyiminin genel yapısını göstermekte, @S1 ilk değer aktarma kesimini, @S3 döngü değişkeninin artırılması kesimine karşı düşmektedir. Bu şekilde sisteme **for** deyiminin üç parçadan oluştuğu, bunların nasıl olması gerektiği ve neyi ifade ettikleri tanımlanmış olur.

Örüntü tanımları oluşturulduktan sonra, her bir parçanın ifade ettiği anlamlar, parçanın bütün içindeki önemini ifade eden puan değeri, örnek ve soru cümleleri, değişken ve değişmezlerin sağlanması gereken sınırlayıcılar ve ilişkili olduğu diğer konu bağlantıları oluşturulur.

Her örüntünün neyi ifade ettiği bilgisi, öğrencinin bu kesimi ilgilendiren bir hata yaptığında açıklayıcı bilgi sunmak amacı ile öngörülmüştür. Yukarıdaki örnek sözdizim örüntülerine ilişkin anlam ifadeleri aşağıdaki gibi verilebilir;

- @S0 : **for** deyiminin genel yapısı 'for (expr1 ; expr2 ; expr3)' şeklindedir.
@S1 : İlkdeğer aktarma deyimini olarak 'değişken = değişmez' şeklinde ifade bekleniyor.
@S2 : Koşul ifadesi olarak 'değişken < değişmez' şeklinde ifade bekleniyor.
@S3 : Döngü değişkenini 1 artırmak için 'değişken++' şeklinde ifade bekleniyor.

Puan değeri örüntü parçasının önemini gösterir. Bir örüntünün alt parçalarına puan verilmesinin nedeni, bu parçaların öğrenci tarafından yanlış ya da doğru yapıldığında konuya ilişkin notunu ilgili

puan kadar artırmak ya da eksiltmek, dolayısıyla öğrenciyi daha detaylı değerlendirebilmekdir. Yukarıdaki örneğe ilişkin puan değerleri aşağıdaki gibi verilebilir.

```
@S0 : 5
@S1 : 2
@S2 : 2
@S3 : 1
```

Soru cümleleri öğrenciyeye sorulacak soruların metnini içerir. Bu cümleler içinde ,@Rn ,@Cn, @Sn gibi ilgili örüntü tanımında yer alan simgeler geçebilir. Yukardaki örneğe ilişkin bir soru cümlesi şu şekilde olabilir;

@C0 değerinden @C1 değerine kadar olan sayıları yazdıran aşağıdaki programı tamamlayınız.

```
main () {
    int @R0;
    @S0
    printf("%4d",@R0);
}
```

Bu cümlede ,@R0 simgesine karşılık değişken adı ve @C0 ,@C1 simgelerine karşılık düşen değişmez değerleri belirtilen sınırlayıcılara uygun olarak sistem tarafından rastgele bir şekilde üretilen olacaktır. Öğrenciyeye soru sorulması durumunda, @S0 öğrenciden beklenen yanıtın uyması gereken anlamlı sözdizim örüntü tanımını ve yanıtın konumunu göstermektedir. Bir örüntü tanımına ilişkin birden fazla soru cümlesi olabilir.

Benzer şekilde, aynı örüntü tanımına ilişkin örnek cümlesi , aşağıdaki gibi olabilir;

@C0 değerinden @C1 değerine kadar olan sayıları 'for' döngüsü kullanarak yazdıran bir program örneği aşağıda görülmektedir;

```
main () {
    int @R0; /* dongu degiskeni */
    @S0
    printf("%4d",@R0);
}
```

Yukarıda verilen örnek örüntü için sınırlayıcı tanımları aşağıdaki gibi yapılabilir:

```
int @C0
int @C1
< (@C0,@C1)
> (@C0,0)
<= (@C1,100)
>= (@C1,10)
AD (@R0,say,count,indis,index)
```

Buna göre sözü geçen örüntü için bir örnek veya soru üretilirken, @C1 ve @C2'nin birer tamsayı, @C0'ın @C1'den küçük, @C0'ın 0'dan büyük ve @C1'in ise 10 ile 100 arasında bir değer almasına dikkat edilecektir. @R0 değişkeni ise 'say', 'count', 'indis' ya da 'index' olarak seçilebilir.

Sınırlayıcı belirtilmeyen değişmez simgeleri için 'default' değer aralığı kullanılır. Ad listesi olmayan değişkenler için tek harften oluşan bir değişken adı yine 'default' aralıktan seçilir.

Anlamli sözdizim örüntü parçası, başka bir konu ile ilintili ise, ilgili konu bağlantısı kurulur. Örneğin, **for** deyiminde *expr1* bir aktarma deyimidir ve aktarma deyimleri konusu ile bağlantısının kurulması, konunun öğrenciye iyi aktarılması açısından önemlidir. Yukardaki örnekteki sözdizim örüntü parçaları şu konular ile ilişkilendirilebilir:

- @S1 : aktarma deyimleri
- @S2 : koşul deyimleri
- @S3 : artrma deyimleri

Örneğin, öğrencinin sürekli olarak @S2 örüntüsü ile ilgili hatalar yaptığı belirlenirse, **for** deyimi anlatımına ara verilerek koşul deyimleri tekrar öğrenciye aktarılır.

ÖĞRETMEN Modülü

Bir BDE yazılımını uygulayacak olan öğretmenin, yazılımın öğrencilere karşı olan davranışlarını her öğrenci grubuna göre en uygun şekilde belirleyebilmesi önemlidir. Bunun için uygulayıcı, ÖĞRETMEN modülünü kullanarak, öğrencinin ne şekilde değerlendirileceği ve bu değerlendirmenin sonucuna göre üretilecek örnek ve soru sayısının, soruların zorluk derecesinin belirlenmesinde kullanılacak parametrelerin oluşturulmasını sağlar. Öğretmen tarafından oluşturulan bu parametreler ve öğretimi sağlayan ÖĞRENCİ modülünün bunları kullanarak öğretim akışını ne şekilde belirlediği aşağıda açıklanmaktadır:

P1: Öğrenciye ilk kez anlatılan bir konu hakkında, toplam olarak verilecek örnek ve soru sayısının belirlenmesinde kullanılır. Bu değer bilgi tabanında örnek ve soruların üretilmesinde kullanılan toplam örüntü tanımı sayısının belirli bir oranını ifade eder. Örneğin, **if** deyimi konusuna ilişkin bilgi tabanında 12 tane örüntü tanımı varsa ve P1 değeri 4 ise her öğrenciye 3 (=12/4) tane örnek ve 3 tane soru üretilecektir.

P2: Öğrencinin daha önce anlatılmış bir konuda, eksik olduğu anlaşılırsa, bu konunun tekrarı sırasında öğrenciye verilecek örnek ve soru sayısını belirler.

P3: Bilgi tabanında yeralan soruların zorluk derecelerinin yeralma konumlarına göre artan sırada olduğu varsayılır. Bu parametre, soruların zorluk derecelerine göre kaç grupta toplanacağını gösterir. Örneğin, **if** deyimi konusuna ilişkin bilgi tabanında 12 tane örüntü tanımı varsa ve P3 değeri 3 ise, örüntüler zorluk derecelerine göre artan sırada 4'erli gruplara ayrılır. Öğrenciye konuya ilişkin örnek ve sorular üretilirken, bu parametre kullanılarak belirlenen, zorluk dereceleri farklı gruplardan basitten zora doğru ve her gruptan eşit sayıda olacak şekilde belirlenir. Ayrıca konu hakkındaki durumu "kötü" olarak nitelendirilen bir öğrenciye örnek ve soru üretimine en basit örüntü grubundan başlanır.

P4, P5, P6: Öğrencinin başarı durumunu gösteren her konuya ilişkin puan değerleri öğrenci modelinde yer almaktadır. Herhangi bir konuda öğrencinin topladığı puan değerinin hangi düzeyi ifade ettiği bu parametreler kullanılarak hesaplanır. P4, P5, P6 değerleri "kötü", "orta", "iyi" ve "çokiyi" değerlendirme sınırlarını belirler.

P7: Öğrencinin hatalarına ilişkin eksiltilecek puan değerinin belirlenmesinde kullanılmaktadır. Bilgi tabanında, soruların üretildiği örüntü tanımları ve alt parçalarının puan değerleri yer almaktadır. Ancak, öğrenci temel olarak bir örüntü tanımını doğru olarak oluşturduğu ve değişken adını ya da değişmez değerlerini yanlış yazdığı durumlarda, eksiltilecek puan değeri, bilgi tabanında değişken ve değişmezler ayrı örüntü şeklinde tanımlanmamış ise belirsizdir. Bu durumlarda P7 parametresinin, öğrencinin hata yaptığı örüntü tanımı parçasına ilişkin bilgi tabanındaki puan değerine bölümü oluşturularak eksiltilecek puan değeri saptanır. Örneğin, öğrencinin **for** deyimine ilişkin bir soruyu yanıtlarken sadece ilk değer aktarma kesiminde döngü değişken adını yanlış yazdığını varsayalım. Bu durumda, ilk değer aktarma kesimine ilişkin puan değeri 4 ve P7 değeri 3 ise öğrencinin **for** konusuna ve aktarma deyimleri konusuna ilişkin puan değeri 4/3 kadar eksiltilecektir.

Uygulayıcı öğretmen bu parametre değerlerini değiştirerek, kullanıcı öğrenci grubuna en uygun öğretim stratejisini belirleyebilir. Örneğin başka dillerde programlama deneyimi iyi olan bir öğrenci grubu için P1 değeri düşük tutularak öğrencilerin daha hızlı bir şekilde ilerlemelerine olanak sağlanmış olur.

ÖĞRENCİ Modülü

Bu yazılım, UZMAN yazılım tarafından oluşturulan bilgi tabanını kullanarak, öğrenciye burada yer alan konuların öğretimini sağlar. Öğretim stratejisinin belirlenmesinde bilgi tabanının yanı sıra, öğretmen tarafından oluşturulan değerlendirme parametrelerini ve öğrenci modelinde yer alan durum bilgilerini kullanır. Bu modul temel olarak aşağıdaki aşamaları içermektedir:

- Öğrencinin tanınması
- Öğrenciye verilecek konunun ve düzeyinin belirlenmesi
- Konu hakkında açıklama bilgilerinin gösterilmesi
- Verilen konu hakkında örneklerin ve soruların seçimi
- Örnek ve sorularda yer alan değişken ve değişmezlerin üretilmesi
- Belirlenen örnek ve soruların öğrenciye sunulması
- Öğrenci yanıtlarının değerlendirilmesi ve öğretim akışının belirlenmesi
- Öğrencinin istekleri doğrultusunda yönlendirmelere izin verilmesi

Öğrencinin Tanınması: Öğrenci, yazılım ile daha önce çalışmış ise (yazılım tarafından tanınıyor ise) o ana kadar olan çalışmaları ve bunlar hakkındaki durumunu ifade eden bilgiler ilgili kütükten okunur. Öğrenciyi modelleyen bu bilgiler, öğrencinin yazılım tarafından tanınmasını ve ona göre öğretim stratejilerinin belirlenmesine yardımcı olur.

Öğrenciye verilecek konunun ve düzeyinin belirlenmesi: Öğrenciye hangi konunun hangi düzeyde verileceğinin belirlenmesi aşağıdaki kriterlere göre yapılır:

- Genel olarak, bilgi tabanında konuların yerleş sırasına göre öğretilecek konuların sırası belirlenir. Ancak bazı özel durumlarda anlatılmakta olan konu akışı kesilerek sıra dışı başka konuların anlatımı yapılabilir.
- Bir konunun öğrenciye anlatılması, sırasıyla konuya ilişkin açıklama bilgilerinin gösterilmesi, örnek ve soruların üretilmesi, daha sonra da özel soruların sorulması şeklindedir.
- Belirli bir konunun genel olarak anlaşılması için gerekli olan örnek, soru ve özel soru sayısı bilgi tabanında yer alan konuya ilişkin toplam örnek ve soru sayısının belirli bir oranı kadardır. Bu oran öğretmen tarafından oluşturulan P1 parametresidir.
- Genel olarak, konunun anlaşılması için gerekli bilgi verildikten sonra, öğrencinin konuyu yeterince anlamadığı belirlenirse öngörülen düzeye gelebilmesi için örnek ve soruların üretilmesine devam edilir. Böylelikle öğrencinin düzeyine göre bir öğretim sağlanmış olur. Öğrencinin konu hakkındaki yeterliliğine karar verilmesi, sorulan soru sayısı, bu sorularda öğrencinin aldığı toplam puan değeri, bu soruların doğru yapılması durumunda alınabilecek toplam puan değeri ve öğretmenin yeterlilik için öngördüğü parametreler (P4, P5, P6) kullanılarak belirlenir. Değerlendirme sonucunda, öğrencinin belirli bir konuda yetersiz olduğu (durum yargısı "kötü") ortaya çıkarsa anlatılan konu akışına ara verilerek yetersiz olduğu ilişkili konu sorulmaksızın tekrar anlatılır.
- Bir konuya ilişkin öğrencinin düzeyine göre bilgi verme işlemi tamamlandıktan sonra, sıradaki konu ele alınır. Ancak, eğer öğrenci bu konuyu daha önce görmüş ve "orta" veya üzerinde bir not almışsa bir sonraki konuya geçilir.

Konu hakkında açıklama bilgilerinin gösterilmesi: Bir konunun anlatımına başlandığında öncelikle bilgi tabanında yer alan konuya ilişkin açıklama bilgileri öğrenciye aktarılır.

Örnek ve Soruların Seçimi: Verilen konu hakkında örnek ve soruların seçimi, bilgi tabanında yeralan toplam örnek ve soru sayısına göre zorluk düzeyleri farklı gruplardan seçilir. Bilgi tabanındaki soruların zorluğunun yeralma sırasına göre artan derecede olduğu varsayılır. Her gruptan P3/P1 tane örnek ya da soru rastgele olarak belirlenir. Grupların sırası ise kolaydan zora doğrudur. Rastgele seçilen soruların ya da örneklerin numaraları saklanarak oturum sırasında, aynı soru ve örneklerin üretilmesi engellenir.

Değişken ve Değişmezlerin Üretimi: Anlamli sözdizim örüntü tanımlarına ilişkin örnek ve soru cümlelerinde geçen @Rn değişken simgeleri için değişken adı, @Cn değişmez simgeleri için ise belirtilen sınırlayıcılara uygun değer üretilmesi sağlanır.

Değişken adları üretilirken, öncelikle değişken simgesine ilişkin sınırlayıcı olup olmadığına bakılır. Eğer değişken için bir ad listesi var ise, bu liste içinden rastgele bir ad seçilir. Herhangi bir sınırlayıcı belirtilmeyen değişken simgeler için ise tek harften oluşan bir değişken adı belirli bir küçük harf aralığında rastgele olarak seçilir.

Değişmez simge değerlerinin üretimi için de öncelikle bilgi tabanında yeralan ilgili sınırlayıcılar incelenir. Değişmez için belirtilen sınırlayıcılar var ise bunlar, değişmezin türü, alabileceği alt ve üst sınır değerleri şeklinde bilgilere dönüştürülür. Bundan sonraki aşamada uygun türde (tamsayı, kayan noktalı, karakter) ve belirlenen aralıkta bir değer rastgele olarak oluşturulur. Bu yöntem ile uygun değer tek bir seçim ile elde edilir. Eğer değişmez için belirlenen değer aralığı değişmez simge içeriyor ise öncelikle ilgili değişmez simgenin değerinin oluşturulması için gerekli işlemler yapılır. Örneğin sınırlayıcılar;

```
>(@C0,1)
<(@C0,@C1)
>(@C1,20)
```

şeklinde ise @C0 için değer aralığı (1, @C0) olacaktır. Dolayısıyla @C0 değerinin oluşturulabilmesi için öncelikle @C1 değerinin belirlenmesi gerekecektir.

Örnek ve soruların öğrenciye sunulması: Belirlenen anlamli sözdizim örüntüsüne ilişkin örnek ve soru cümlelerinden rastgele olarak bir tanesi seçilir ve ekranda görüntülenir. Bu sırada örnek ve soru cümlesi içinde geçen değişken ve değişmez simgelerin yerine üretilmiş olan değişken adları ve değerler kullanılır. Verilen bir örnek ise cümle içinde geçen @Sn simgeleri yerine, ilgili anlamli sözdizim örüntü tanımı özyineli bir şekilde çözümlenerek oluşturulan ifade konur. Bir soru cümlesi söz konusu ise, @Sn simgesinin geçtiği konum öğrenciden yanıt almak üzere işaretlenir ve ilgili satır boş bırakılır. Yukarıdaki for deyimine ilişkin örnek örüntü cümlesi kullanılarak aşağıdaki görünümde bir örnek üretilir:

```
=====| C programlama dili eğitimi |=====
Konu Kontrol Akışı
Altkonu Döngüler-For
=|Örnek ve Sorular |=====
OR.10 değerinden 50 değerine kadar olan sayıları 'for' döngüsü kullanarak yazdıran
bir program örneği aşağıda görülmektedir;

    main () {
        int say; /* dongu degiskeni */
        for (say = 10 ; say < 50 ; say++)
            printf("%4d",say);
    }

=====
F1-yardım F2-sonraki adım F4-metni tekrar görme F3-F5-başka konuya geçiş Esc-çıkış
Konu bilgileri tanıtımı                               Ins ON
```

Öğrenci Yanıtlarının Değerlendirilmesi: Bilgi tabanında yer alan örüntü tanımlarıyla öğrenci yanıtı karşılaştırılarak analiz edilir. Bunun sonucuna göre öğrenciye gerekli açıklamalar yapılır ve konu hakkındaki yeterliliğini ifade eden puan değerleri belirlenir. Yanıtlar, bilgi tabanında yer alan tanımlar ne derece detaylı ise o ölçüde iyi analiz edilebilir. Bilgi tabanındaki bu tanımların örüntü şeklinde gösterilmiş olması, öğrenci hata yaptığı durumlarda, hatanın yeri ve niteliği ayrıca doğrusunun ne şekilde olması gerektiğinin anlaşılabilmesine ilgili açıklamaların yapılabilmesine olanak sağlamaktadır.

Öğrencinin yanıtı, bilgi tabanında uyması gereken anlamlı sözdizim örüntü tanımı ile örüntü tanıma teknikleri kullanılarak karşılaştırılır ve değerlendirilir. Bu amaçla, öncelikle örüntü tanımında yer alan değişmez ifadeler (@Sn, @Cn ve @Rn dışındaki ifadeler) yanıt içinde aranır. Bu ifadeler olması gereken yerlerde bulunursa, örüntü tanımında yer alan (eğer var ise) altgrup örüntü tanımları yerine, öğrencinin yanıtında karşılık gelen ifadeler ayrıştırılır. Daha sonra bu ifadeler uyması gereken altgrup örüntü tanımları ile, benzer şekilde özyineli olarak karşılaştırılır. En sonunda değişken ve değişmez değerleri yerine öğrencinin yazdığı ifadeler düzeyinde ayrıştırma yapılarak bunlar karşılaştırılır. Değerlendirme sırasında görülen ilk hatada durulur ve gereken açıklama ile hatanın türüne göre öğrencinin konu hakkındaki bilgisini ifade eden puan değeri eksiltir.

Yukarıdaki **for** deyimine ilişkin örüntü bilgilerini (örüntü tanımı, ifade ettikleri anlamlar, puan değerleri, ilişkili olduğu konular, sınırlayıcılar ve soru cümlesi) örnek olarak alalım. Bilgi tabanında yer alan bu örüntüye ilişkin soru cümlesinde kullanılan değişken ve değişmezlere ait sınırlayıcılar dikkate alınarak aşağıda görüldüğü gibi öğrenciye sorulur:

5 değerinden 25 değerine kadar olan sayıları yazdıran aşağıdaki programı tamamlayınız.

```
main () {
  int  indis;
  printf("%4d", indis);
}
```

Bu soruya ilişkin öğrencinin yanıtı değerlendirilirken öncelikle;

```
for ( * ; * ; * ) ifadesi aranacaktır.
      ↓   ↓   ↓
      @S1 @S2 @S3
```

Burada, * simgesi sıfır yada daha fazla karaktere karşılık gelir. Eğer bu ifade görünümü elde edilemez ise @S0'a ilişkin anlam ifadesi

for deyiminin genel yapısı 'for (expr1 ; expr2 ; expr3)' şeklindedir

öğrenciye açıklama olarak verilecektir. Aynı zamanda, @S0'a ilişkin puan değeri (5) kadar öğrencinin **for** konusu ile ilgili puanı eksiltilecektir. İfade görünümü doğru olarak elde edilirse, yanıtın '*' simgelerine karşı gelen kesimleri benzer şekilde ilişkilendirildikleri anlamlı sözdizim örüntü parçaları ile özyineli olarak karşılaştırılırlar. Örneğin, öğrencinin bu soruya verdiği yanıt :

```
for (indis = 1 ; indis < 10 ; indis++)
```

ise

```
indis = 1    @S1 ile,
indis < 10   @S2 ile,
indis++     @S3 ile ilişkilendirilir.
```

Benzer şekilde,

@R0 = @C0
* = *
indis = 1

@S1: @R0 = @C0 çözümlenirken öncelikle '=' ifadesi aranacak, bulunamaz ise @S1 ile ilgili açıklayıcı ifade öğrenciye mesaj olarak verilecek ve hem **for** konusuna ilişkin notu ve hem de aktarma deyimlerine ilişkin notu @S1'in puan değeri kadar düşürülecektir. '=' ifadesinin bulunması durumunda, @R0 ve @C0 ile eşleştirilen kesimler karşılaştırılır. Eğer öğrencinin yanıtındaki değişken adı sistem tarafından beklenen değişken adından farklı ise bu değişken adının geçerli olup olmadığına bakılarak, @S1 ile ilgili olan anlam ifadesi açıklama olarak verildikten sonra "beklenen değişken adı bulunamadı" şeklinde bir mesaj öğrenciye verilir. Bu durumda, öğrenci @S1 anlamlı sözdizim örüntü parçasının değişken adına ilişkin kesiminde hata yapmıştır ve puanı, @S1 de belirtilen puanın belirli bir oranı kadar eksilir. Bu oran öğretmen tarafından P7 değeri olarak belirlenir. Bir ilkdeğer aktarma deyimi beklenen kesimde, '=' ifadesinin yazılmaması ile bu ifadenin yazılması fakat değişken adının hatalı yazılması aynı derecede önem taşıyan hatalar olmadığından bunların hem öğrenciye yanlısını açıklarken hem de puan değerlendirmesi yaparken ayrıştırılması çok önemlidir. Değişmez değerlerin karşılaştırılması da değişkenlerde olduğu gibidir.

Bazı durumlarda, öğrencinin yanıtı doğru olduğu halde yanlış olarak değerlendirilebilir. Bu, iki nedenden kaynaklanabilir:

- Öğrenci yanıtı karşılaştırıldığı anlamlı sözdizim örüntü tanımına eşdeğerdir.
- Değişmez ifadeler (anlamlı sözdizim örüntü tanımlarında '@' ile başlayan simgeler dışındakiler) temel alınarak ayrılan örüntü parçaları aynı değişmez ifadenin ardarda geçtiği bazı durumlarda problem oluşturabilir.

Birinci problem, örneğin;

(a * b) = (b * a)
(a = a + b) = (a += b) (C dilinde)
(a = a + 1) = (a++) = (a += 1) ... (C dilinde)

gibi eşdeğer ifadelerden kaynaklanır. Bu duruma çözüm olarak öğrencinin yanıtı ilgili anlamlı sözdizim örüntüsüne uymadığı zaman, bilgi tabanında yeralan eşdeğerlik bilgileri taranarak eşdeğer bir ifade olup olmadığına bakılır. Eşdeğer bir ifade bulunursa bu ifadeye uygun olarak örüntü tanımı değiştirilerek öğrencinin yanıtı buna göre tekrar analiz edilir.

İkinci problem ise

@S0 : @S1 + @R0
@S1 : @R1 + @R2

gibi örüntü tanımlarında ardarda aynı değişmez kullanılmasından kaynaklanır. @S1 örüntüsünü @S0 içinde yerine koyarsak '+' simgesi ardarda tekrarlanacak ve bu örüntü ile karşılaştırılacak bir ifadede @S1 doğru olarak ayrıştırılamayacaktır. Örneğin, örüntü tanımında geçen @R0 = a, @R1 = b, @R2 = c değişken adları sistem tarafından üretilmiş olsun. Öğrencinin bu konudaki yanıtı ;

a + b + c

ise sistem tarafından, 'a' @S1 örüntüsü ile 'b + c' ifadesi de @R0 ile eşleştirilerek karşılaştırılacak ve yanıt doğru olduğu halde yanlış sonuca varılacaktır. Bunun için bütün değişmez ifadelerin tüm eşleşmeleri tek tek sınanmaktadır.

Öğrencinin İstekleri Doğrultusunda Yapılan Yönlendirmeler: Öğrencinin istekleri doğrultusunda yapılan yönlendirmeler iki türlü olmaktadır. Birincisi, öğrenci herhangi bir anda (bir soru

sorulduğunda, açıklamalar gösterilirken, örnek verilirken, vb.) başka bir konuya ilişkin bilgi isteyebilir. Bu amaçla öğrenciye, istediği konunun anlatımı yapıldıktan sonra, kalınan konuya geri dönülür. Öğrencinin isteği doğrultusunda yapılan ikinci tür yönlendirmede ise geri dönüş yoktur. Benzer şekilde, seçtiği konudan itibaren ileriye doğru devam etmesi sağlanır.

Değerlendirme ve Sonuçlar

Bu bildiride, bilgisayar programlama dilleri eğitiminde kullanılmak amacıyla gerçekleştirilen bir akıllı eğitim sistemi olan ProDil anlatılmıştır. Programlama dillerine ait deyimler, anlamsal ve sözdizimsel tanımlar, bunlara ilişkin açıklamalar, örnek ve soruların üretilmesinde ve öğrencinin yanıtlarının değerlendirilmesinde kullanılacak bilgiler bilgi tabanında toplanmıştır. Örnek ve soruların üretilmesinde ve öğrencinin yanıtlarının değerlendirilmesinde kullanılan tanımlar örüntü şeklinde yer almaktadır.

Gerçekleştirilen anlayışlı sistem, bilgi tabanında yer alan bu tanımları kullanarak öğrenciye çeşitli örnek ve sorular üretebilmektedir. Öğrencinin yanıtlarının değerlendirilmesinde, hatasının nerede olduğunu örüntü tanıma teknikleri kullanarak anlaşılabilir. Benzer şekilde, doğrunun ne olması gerektiği yine bu tanımları kullanarak üretilmekte ve öğrenciye gerekli açıklamalar yapılarak hatasına göre konu hakkındaki bilgisi ölçülebilmektedir. Ayrıca öğrencinin belirli bir konuya ilişkin verilen soruda yaptığı hataların, ilişkili olan başka bir konudan kaynaklanıp kaynaklanmadığı analiz edilebilmekte ve bu durumda konu akışı, öğrenciden soruya ilişkin tekrar yanıt almak üzere kesilerek, öncelikle ilgili konudaki eksikliğini tamamlamak amacı ile gerekli açıklamalar yapılmakta, örnek ve soruların üretilmesi sağlanmaktadır. Böylece öğrencinin gereksinimleri doğrultusunda bir eğitim verilebilmektedir. Sistem aynı zamanda, öğrencinin istekleri doğrultusunda bir konunun anlatımına ya da belirli bir konudan itibaren konu akışını belirlemesine olanak sağlamaktadır.

Prodil, C programlama dili ve MASM kullanılarak IBM PC uyumlu bilgisayar üzerinde gerçekleştirilmiştir. Bilgi tabanına C diline ilişkin bilgiler girilerek deneme ve sınamalar yapılmış, başarılı olduğu gözlenmiştir.

KAYNAKLAR

Anderson, J. R. ve B. J. Reiser, "The Lisp Tutor" Byte 10, 4, 1985.

Davenport, D. ve H. A. Güvenir, "Knowledge Base Concepts for ITS" Proceedings of ISCIS-3, İzmir, 1988.

Orhun, E. ve E. Demirörs, "Issues in the Design of an Intelligent Tutoring System for Computer Programming" Proceedings of ISCIS-2, İstanbul, 1987.

Soloway, E. ve W. L. Johnson, "PROUST", Byte 10, 4, 1985.

Şimşek, B. "Programlama Dilleri Öğretimi için Anlayışlı bir Eğitim Sisteminin Tasarımı ve Gerçekleştirilmesi" Yüksek Mühendislik Tezi, Hacettepe Üniversitesi, BBM, Ankara, 1990.

Wolf, B. ve D. D. McDonald, "Building a Computer Tutor: Design Issues" IEEE Computer Magazine, 1984.