

ZAMAN ÇİZELGELEME PROBLEMLERİ İÇİN STANDART BİR VERİ FORMATI ÇALIŞMASI: TTML

Ender Özcan¹

e-mail: ozcan@ics.yeditepe.edu.tr

Yeditepe Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, 34850, Kayışdağı/İstanbul

Anahtar Kelimeler: Zaman Çizelgeleme, XML, Standart Veri Formatı, Web Standartları, Kısıtlı Eniyileme

ÖZET

Zaman çizelgeleme problemleri kısıtlı eniyileme problemleridir. Çözüm için geliştirilen modeller ve problem temsilinde kullanılan veri formatları birbirinden oldukça farklıdır. Bu sebeple literatürde bulunan sonuçları değerlendirmek, karşılaştırma ve yaratılan veriyi paylaşmak zordur. Zaman çizelgeleme konusuyla ilgilenen araştırmacıların, kullanıcıların ve endüstri çalışanlarının XML'e dayalı yeni bir standartı destekleyecekleri atılmış olan adımlardan anlaşılmaktadır. Bu makalede, TTML adı verilen, zaman çizelgeleme problemleri için tasarlanan XML'e dayalı yeni bir standart ve esas alınan modeller tanımlanmaktadır.

I. GİRİŞ

Zaman çizelgeleme problemleri özellikle devamlı yada ayrık zaman dilimlerinin ve diğer ek kaynakların değişik kısıtlar göz önünde bulundurularak, kullanıcılara *kabul edilebilir* bir şekilde atanmasını gerektiren zor problemlerdir [9]. Schmidt, bildirisinde zaman çizelgelemeyle ilgili ilk çalışmalardan bahseder [19]. En basit haliyle zaman çizelgeleme problemi harita boyama problemine indirgenebilir [13, 20]. Bir çok araştırmacı değişik metodlarla, değişik veriler kullanarak, birbirinden farklı kısıtlarla ellerindeki problemleri çözmeye çalışmışlardır [1, 4-10, 14-18]. Bu kadar araştırma yapılmasına rağmen, kullanılan modellerin farklılığı ve problem tanımı için genel bir standardın olmamasından dolayı, eldeki çözümlerin kıyaslanabilmesi, standart örnek deney setinin oluşturulabilmesi çok zordur.

Zaman çizelgeleme problemleri için ortak bir veri formatı oluşturma fikri ICPATA'95 konferansında Andrew Cumming tarafından ortaya atılmıştır. Bu konuda yapılan çalışmalar Kingston'ın geliştirdiği ve Internet üzerinden ücretsiz elde edilebilecek SSTL adı verilen bir yorumlayıcı ile son bulmuştur [3, 12]. Bu yorumlayıcı yeterince duyurulmadığından yada olası kullanım zorlukları gibi sebeplerden ilgili araştırmacılar tarafından beklenen ilgiyi görmemiştir.

Causmaecker ve ark. [2] zaman çizelgeleme araştırmacılarının gelişen Internet teknolojilerinden de faydalanabilmesi için çok katmanlı bir mimariye sahip olan *Anlambilime Dayalı Web* (ADW) çalışmalarını başlatmıştır [12]. ADW'i gerçekleştirebilmek için tasarlanan katmanlardan birisi de, belgelerin sözdizimini kotaran XML (eXtensible Markup Language) teknolojisidir. XML kullanarak zaman çizelgeleme problemleri için standart bir veri formatı oluşturulabilir. Bu fikirden yola çıkarak, konudaki ilk çalışmalardan biri olan TTML (TimeTabling Markup Language) adı verilen XML tabanlı zaman çizelgeleme anlamlandırma dili tasarlanmaya başlanmıştır [15].

İlerideki bölümlerde XML teknolojisi, TTML tasarlanırken kullanılan modeller ile bazı önemli TTML öğeleri anlatılmaktadır.

II. XML

XML, HTML (HyperText Markup Language) gibi anlamlandırma dilleri yaratabilmek ve bu dillerin dilbilgisi kurallarını tanımlayabilmek için kullanılan bir dildir. HTML insan-makina etkileşimini esas alır. Tarayıcılar imleri yorumlayarak, bu imlerin içerisindeki yazıların (bilgilerin) nasıl görüntüleyeceğine karar verir. XML ile kullanıcılar kendi imlerini yaratabilir ve kendi belgelerinin yapısını belirleyebilirler. XML makinalar arası etkileşimde veriyi temsil etmenin en verimli yollarından biridir. Hatta aynı tipteki XML belgelerin bütünsel olarak birbirine bağlanmış bir veritabanı olduğu düşünülebilir. Bir belgenin içeriği ile sunumu XML sayesinde birbirinden ayrılabilir. Böylece hem insanların hem de makinaların imlerin içeriğini anlamaları mümkün hale gelir. XML yapılandırılmış verinin web üzerinden bir noktadan bir noktaya kayıpsız aktarılabilmesini kolaylaştıran bir unsurdur. Bir XML formatını bir diğerine çevirmek kolaydır (XSLT, ayrıştırıcılar).

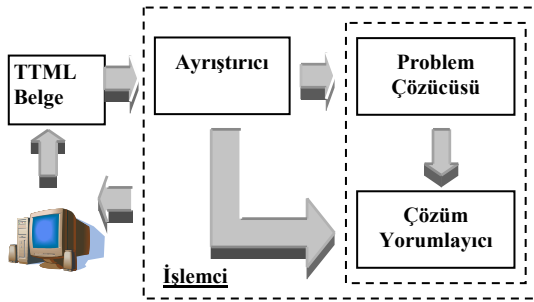
XML belgeler ikiye ayrılırlar: iyi yapılanmış belgeler ve geçerli belgeler. İyi yapılanmış belgeler XML'in dilbilgisi kurallarına uymalıdır. Geçerli bir belgenin ise iyi yapılanmış belgeye ek olarak uygun belge türü bildirimine ihtiyacı vardır. Bildirim, o belgede kullanılan imlerin hangileriyle ve ne şekilde iç içe geçeceği, her imin öznelikleri ve özneliklerin alabileceği değerler gibi dilbilgisi kurallarının tanımını içerir. Geçerli belgeler bildirilen bütün dilbilgisi kurallarına uyan belgelerdir. Bir XML belge için dilbilgisi kurallarını tanımlamanın iki yolu vardır: *DTD* (Document Type Definition) ve *Schema*.

Zaman çizelgeleme problemleri için standart bir veri formatı genel, tam ve dönüştürülebilir olmalıdır. TTML, ilgili araştırmacılar tarafından saptanmış gereklilikler göz önünde bulundurularak tasarlanmaktadır [3, 11]. Bütün TTML imleri zaman çizelgeleme elemanlarının açık ve net olarak tanımlanmasına bağlıdır. Bu makalede TTML için DTD yada Schema tanımlanmayacak, TTML öğeleri oluşturulurken DTD yada Schemaya yönelik detaylar sunulacaktır. XML belgeyi inceleyerek DTD yada Schema oluşturabilen XML programlar mevcuttur. Değinilen Internet teknolojileriyle ilgili detaylı bilgilere W3C sitesinden ulaşılabilir [21].

III. ZAMAN ÇİZELGELEME PROBLEMLERİNİN MODELENMESİ

TTML, Şekil 1'deki model esas alınarak tasarlanmıştır. Kullanıcılar problemine çözüm arayan ve araştırmacı olmak üzere iki türdür. Her iki kullanıcı için de ortak senaryo,

ellerindeki bir probleme çözüm aramaktır. Bu durumda kullanıcılar oluşturacakları TTML belge ile İnternet yada kendi makinaları üzerinde çalışan, problemlerine uygun bir TTML işlemcisini kullanarak çözüm elde edebileceklerdir. TTML belgesi, işlemci tarafından alınarak, yine aynı TTML belgesinde belirtilen çıktıyı yada çıktıları sağlayacaktır. İşlemci TTML ayrıştırıcısı aracılığıyla gelen veriyi kendi içerisinde kullandığı veri yapılarına dönüştürecektir. Bulunan sonuç *Çözüm Yorumlayıcı* (ÇY) tarafından kullanıcının TTML belgesindeki isteklerine göre kendisine gönderilecektir. Örneğin, bir zaman çizelgeleme probleminde derslerin buluşma zamanları çözüm olarak kullanıcıya sunulabilir. Kullanıcı, eğitmenlerin yada belli bir sınıftaki öğrencilerin ders programlarının görüntüsünü isteyebilir. ÇY kullanıcının isteğine göre çıktı üretebilmelidir.

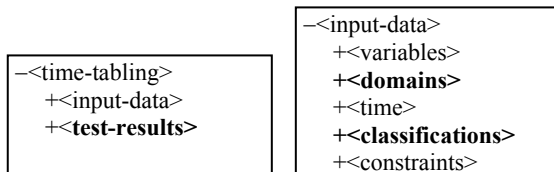


Şekil 1. TTML tasarlanırken esas alınan model

Zaman çizelgeleme problemleri dört ana kümeyle tanımlanabilir; (D, T, S, K) . D kümesi değişkenler kümesidir (*variables*). T kümesi değişkenlerin tanım kümelerini ifade eder (*domains*). Zaman çizelgeleme problemlerinde T kümesi *en az* bir eleman içerir, o da zaman kümesidir (*time*). S kümesi, D kümesi üzerinde tanımlanmış sınıflandırmalardır (*classifications*). K kümesi de V kümesindeki her bir değişkenin alabileceği değerler için tanımlanmış kısıtlardır (*constraints*). S kümesi bu kısıtların tanım kolaylığı sağlar. Öncelikli hedef TTML kullanarak verilerin paylaşılmasını sağlamaktır. Bu sebeple TTML belgeleri ilgili zaman çizelgeleme problemini tanımlayabilecek (D, T, S, K) kümelerini içermelidir. TTML dilinin benzersiz kök ögesi olarak *time-tabling* seçilmiştir. Araştırmacılar TTML'in daha fazla bilgi içermesini bekleyebilir. Yukarıda anlatılanlara ek olarak diğer bir hedefimiz sonuçların, dolayısıyla değişik yaklaşımların karşılaştırılabilmesidir. Zorunlu olmasa da bir TTML belgesi yapılan testleri ve elde edilen sonuçları içerebilmelidir. Daha fazla detay istenirse sonucu değerlendirmede kullanılan fonksiyon da belgeye eklenebilir.

Bütün anlatılanların ışığında ana ve ilk seviye alt öğeleri ile girdinin alt öğeleri

Şekil 2'de gösterildiği gibidir. TTML belgeleri, isteğe bağlı olarak test sonuçlarını içerir. Ana öge olan *time-tabling* için belgenin yazarı, belge üzerinde yapılan son değişiklik tarihi, problem tipi ve benzeri bilgileri içeren öznitelikler tanımlanabilir.



Şekil 2. Bir TTML belgenin ana, ilk seviye ve girdi öğesinin alt öğeleri.

IV. TTML ALT ÖGELERİNİN MODELLENMESİ

Bir TTML belgesi girdi olarak en azından zaman, değişkenler ve kısıtları içermelidir. En genel haliyle zaman çizelgesi arka arkaya sıralanmış zaman aralıklarının bütünü halinde tanımlanabilir. Çizelgelerde kullanılan zaman, ayrık, sürekli yada her ikisinin karışımı olabilir. Karışık zaman tipi, zaman çizelgeleme problemlerinde görülmesine de desteklenebilir. Ayrıca zaman tamamen yada kısmen periyodik olarak ilerleyebilir, veya tanımlanan zaman çizelgesinde tekrarlar hiç olmayabilir. TTML tasarlanırken zaman çizelgesi tablo olarak düşünülmüştür. Çizelgelerde zaman, sütun öncelikli ilerler. TTML işlemcisi kısıtları değerlendirirken ayrık zaman dilimlerinin büyüklüklerine, yada sürekli zamanın hassasiyet derecesine (saniye, salise, vs.) ihtiyaç duyabilir. Bu sebeple her sütun ve satırın ifade ettiği zaman aralığı tanımlanabilmelidir. Ayrık zaman dilimleri birbirine eşit olmayabilir. Ayrıca, kullanıcı çıktılarda her sütun, yada ayrık zaman dilimleri için uygun gördüğü bir etiket tanımlamak isteyebilir. Zaman çizelgesi *time* imiyle tanımlanır.

Değişkenler kümesi, zaman çizelgesinden yada ek olarak başka kümelerden atama yapılması istenen elemanları içerir. Örneğin, üniversite ders programı hazırlama problemlerinde ilgili dönemde açılan bütün dersler şubeleriyle değişkenler kümesini oluşturur. Bu kümedeki her elemana en az bir zaman dilimi atanmalıdır. Buna ek olarak her ders şubesine bir de sınıf ataması beklenebilir. Herhangi bir kümedeki eleman çeşitli özelliklere sahip olabilir. Önceki örnekten yola çıkarak, her ders şubesinin buluşma sayısı önemlidir, ama tek başına yetersizdir. Bunun yanısıra, ders şubelerinin her bir buluşmasının uzunluğu da bilinmelidir. Bu bilgi zaman çizelgesinin türüne göre ayrık yada sürekli olabilir. S kümeleri ya değişkenler kümesinin sınıflandırılabilir alt kümeleri yada değişkenler kümesinden bağımsız kendi içlerinde birbirlerine bir şekilde bağlı olan kümelerdir. Örneğin her bir eğitmenin verdiği ders şubeleri, eğitmenler grubuna ait kümeler olup, her biri kümenin alt kümesidir. Benzer şekilde bir bölümde açılan herhangi bir döneme ait ders şubelerinin kümesi de değişkenler kümesinin alt kümeleridir.

Dersi alan öğrencilerin sayısı gibi bütün derslikler ile onlara ait bir özellik olan kapasiteleri değişkenler kümesinden bağımsız bir küme oluşturur. Kısıtlar bu kümelere dayalı oluşturulabilir. Örneğin, ders şubesine atanan sınıfın kapasitesinin, o dersi alan öğrenci sayısından büyük ya da eşit olması beklenebilir. Dolayısıyla, bu özellikler de ifade edilebilmelidir. Her kümeye ve gruba benzersiz bir isim, her küme elemanına uygun benzersiz bir numara atanabilir. Kısıtlar tanımlanırken bu bilgiler çok yararlı olacaktır.

Kümeler düşünüldüğünde herhangi bir sınıflandırma içerisindeki değişkenler kümesi elemanlarının aynı saate atanmaması istenebilir (*no-overlap*). Örneğin, bir eğitmen sorumlu olduğu dersleri aynı anda veremez, veya bir öğrencinin dönem dersleri aynı saate atanamaz. Bazı değişkenler kümesi elemanlarının atamaları önceden biliniyor olabilir (*require*). Örneğin; Türkiye dersleri bütün şubeler için Pazartesi 10:00 ile 12:00 arasında veriliyor olabilir. Bazı grup elemanlarının ise belirli saatlere atanmaması istenebilir (*exclude*). Örneğin; bir fakültede A bölümü için 12:00-13:00, B bölümü için 13:00-14:00 arası öğle tatili olarak belirlenmiş olup, dolayısıyla o bölümlere ait dersler bu saatlere atanamaz. Benzer şekilde bir grup içerisindeki değişkenler kümesi elemanlarının zaman çizelgesinde belirli

aralıklara atanması (*require*), yada atanmaması gerekebilir (*exclude*). Örneğin; yarızamanlı bir eğitmen vereceği derslerinin sadece salı ve cuma günlerinde olmasını, veya özellikle salı hariç herhangi bir günde olmasını tercih edebilir. Diğer bir tür kısıt ise kesişmeyen olayların bir takım bağılı değerlendirilmeler ışığında zaman çizelgesinde yerini almalarıdır (*total*). Örneğin bir eğitmenin ders verdiği gün toplam olarak 5 saati geçmemesi, yada 2 saatten az ders vermemesi, yada ortalama günde 2 saat ders vermesi istenebilir. Yine kısıt olarak, atamalar sırasında bir sıra gözetilmesi istenebilir (*ordering*). Örneğin bir dersin labaratuvar saatlerinin dersten sonra olması tercih edilebilir.

```

-<constraints>
  -<hard>
    +<no-overlap>
    +<exclude>
    +<require>
    +<ordering>
    +<total>
    +<cf>
  +<soft>

```

Şekil 3. TTML’de kullanılan kısıtları ifade eden ögenin alt öğeleri

Kümeler ve özellikleri üzerinde tanımlanabilecek diğer tipteki kısıtlar ise *cf* ögesinin altında bir araya toplanabilir. TTML’de kullanılan kısıtları ifade eden ögenin alt öğeleri Şekil 3’dedir. Her bir alt öğe içerisinde tanımlanan kısıt, o kısıtın hangi elemanlar için geçerli olduğu bilgisini içerir. Bu elemanlar değişkenler kümesinden belirlenmiş bir takım elemanlar yada bir kaç gruba ait elemanlar olabilir.

```

-<test-results>
  -<result>
    +<eval>
    +<test-output>

```

Şekil 4. TTML’de test sonuçları ögesinin alt öğeleri

Araştırmacılar, yarattıkları veri üzerinde deneyler yaparak, elde ettikleri sonucu diğer araştırmacılarla paylaşmak isteyebilirler. TTML belgenin sonuna bu deney sonuçları ilâştirilebilir. Ek bilgi olarak deney sonucunu değerlendirmek için kullanılan fonksiyon da isteğe bağlı olarak TTML belgeye eklenebilir. Bu fonksiyonun ve *cf* alt ögesindeki diğer fonksiyonların tanımlanması için MathML’den faydalanılabilir. MathML matematik terimlerinin tanımlanabilmesini sağlayan bir XML standardıdır. Şekil 4, TTML’de test sonuçlarını ifade eden ögelerin alt öğelerini göstermektedir.

V. SONUÇLAR

Bu makalede zaman çizelgeleme problemlerinin ifadesi için gerekli olan modeller ve XML’e dayalı yeni bir standart olan TTML tanıtılmıştır. TTML XML’in sağladığı üstünlük ve güce sahiptir. TTML kullanarak gerçek zaman çizelgeleme problemleri tanımlanabilir. Zaman çizelgeleme araştırmacıları bu modele kıyaslamalarda kullanılmak üzere deney sonuçlarını da ekleyebilirler. TTML MathML’den yararlanılarak güçlendirilebilir. Zaman çizelgeleme problemlerinin standart olarak ifade edilebilmesi için belirlenmiş gerekliliklerin tümü hatta daha fazlasının sağlanması düşünülmektedir [3]. TTML’i destekleyen programlar İnternet üzerine taşınabilir, hatta bir web servisi haline getirilebilir. İlerideki çalışmalar TTML öğelerini detaylandırmaya ve kullanılan modelden yola çıkarak sentetik

veri üretmek değişik zorluklarda zaman çizelgeleme problemleri yaratmaya yönelik olacaktır.

KAYNAKÇA

1. D. Abramson, H. Dang and M. Krisnamoorthy, *Simulated Annealing Cooling Schedules for the School Timetabling Problem*, Asia-Pacific Journal of Operational Research, 16 (1999) 1-22.
2. P. D. Causmaecker, P. Demeester, Y. Lu and G. Vanden, *Using Web Standards for Timetabling*, PATAT’02, pp.238-258, 2002.
3. E. K. Burke, P. A. Pepper and J. H. Kingston, *A Standard Data Format for Timetabling Instances*, Springer Lecture Notes in Computer Science, 1408:213-222 (1997).
4. E. K. Burke, J.P. Newall, R.F. Weare, *A Memetic Algorithm for University Exam Timetabling*, Lecture Notes in Computer Science, 1153:241-250, Springer, 1996.
5. D.Corne, P. Ross, H.L. Fang, *Evolutionary Timetabling: Practice, Prospects and Work in Progress*, Proc. of the UK Planning and Scheduling SIG Workshop, 1994.
6. JP Cladeira, AC Rosa, *School Timetabling using Genetic Search*, PATAT’97, pp. 115-122, 1997.
7. F.P.M. Dignum, W.P.M. Nuijten, L.M.A. Janssen, *Solving a Time Tabling Problem by Constraint Satisfaction*, Tech. report, Eindhoven University of Technology, 1995
8. W. Erben, J. Keppler, *A Genetic Algorithm Solving a Weekly Course-Timetabling Problem*, Proc. of the First Int. Conf. on the Practice and Theory of Automated Timetabling (ICPTAT), pp. 21-32, Napier University, Edinburgh, 1995.
9. S. Even, A. Itai, and A. Shamir, *On the Complexity of Timetable and Multicommodity Flow Problems*, SIAM J. Comput., 5(4):691-703, December 1976.
10. A. Hertz, *Finding a feasible course schedule using a tabu search*, Discrete Applied Mathematics, 35 (1992), 255-270.
11. J.H. Kingston, *Modeling timetabling problems with STTL*, Springer Lecture Notes in Computer Science, 2079:309, 2001.
12. T. B.-Lee, J. Hendler and O. Lassila, “The Semantic Web”, Scientific American, May 2001.
13. F.T. Leighton, *A graph coloring algorithm for large scheduling problems*, Journal of Research of the National Bureau of Standards, 84:489-506, 1979.
14. A. Monfroglio, *Timetabling Through a Deductive Database: A Case Study*, Data & Knowledge Engineering, 3:1-27, 1988.
15. E. Ozcan, *Towards an XML based standard for Timetabling Problems: TTML*, MISTA’03, accepted.
16. E. Ozcan, A. Alkan, *Timetabling using a Steady State Genetic Algorithm*, PATAT’02, pp. 104-107, 2002.
17. E. Ozcan, A. Alkan, *Çok Nüfuslu Kararlı Hal Genetik Algoritması Kullanarak Otomatik Çizelgeleme*, 19. Bilişim Kurultayı Bildiriler Kitabı, s. 149-154, 2002.
18. A. Schaerf, *Tabu Search Techniques for Large High-School Timetabling Problems*, Proc. of the Fourteenth National Conference on AI, pp. 363-368, August, 1996.
19. G. Schmidt, and T. Strohlein, *Time table construction-an annotated bibliography*, The Computer Journal, 23(4):307-316, 1979
20. D. De Werra, *An introduction to timetabling*, European Journal of Operations Research, 19:151-162, 1985.
21. World Wide Web Consortium web site, <http://www.w3c.org>, June 2003.