

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**AYRIK OLAYLI SİSTEMLERİN
PETRİ AĞI İLE MODELLENMESİ**

YÜKSEK LİSANS TEZİ
Kontrol ve Bilg. Müh. Kadir ÇELİKMIH

Anabilim Dalı : KONTROL VE BİLGİSAYAR MÜHENDİSLİĞİ

Programı : KONTROL VE BİLGİSAYAR MÜHENDİSLİĞİ

KASIM 2002

**AYRIK OLAYLI SİSTEMLERİN
PETRİ AĞI İLE MODELLENMESİ**

YÜKSEK LİSANS TEZİ
Kontrol ve Bilg. Müh. Kadir Çelikmih
(504981173)

Tezin Enstitüye Verildiği Tarih : 02 Ekim 2002
Tezin Savunulduğu Tarih : 08 Kasım 2002

Tez Danışmanı : Doç.Dr. Salman KURTULAN
Diğer Jüri Üyeleri Prof.Dr. Leyla GÖREN
Y.Doç.Dr. Neslihan Serap ŞENGÖR

KASIM 2002

ÖNSÖZ

Amacı, Petri ağı teorisi ile ayrık olaylı sistemlerin modellenmesi olan bu çalışmada, yardımlarını esirgemeyen danışmanım Doç. Dr. Salman Kurtulan'a ve iş akışları konusunda yardımcı olan SAP İnsan Kaynakları danışmanları Burçin Oryan ve Alkin Aksoy'a teşekkür ederim.

Kasım-2002

Kadir Çelikkıh

İÇİNDEKİLER

KISALTMALAR	v
ŞEKİL LİSTESİ	vi
SEMBOL LİSTESİ	viii
ÖZET	ix
SUMMARY	x
1. GİRİŞ	1
1.1. Giriş ve Çalışmanın Amacı	1
2. PETRİ AĞLARININ TEMELLERİ	4
2.1. Petri Ağları Notasyonları ve Tanımları	4
2.2. Petri Ağları İşaretleri ve Durum Uzayı	6
2.3. Petri Ağları Dinamikleri	8
2.4. Durum Denklemleri	12
3. PETRİ AĞLARI İLE MODELLEME	15
3.1. İmalat Makinasının Petri Ağı ile Modellenmesi	15
3.2. Üretici Tüketici Probleminin Modellenmesi	16
3.3. Program Modellenmesi	20
3.4. Kuyruk Sistemleri İçin Petri Ağı Modeli	24
3.5. Yemek Yiyen Filozoflar Problemi (Dining Philosophers Problem)	26
4. PETRİ AĞLARININ ANALİZLERİ	28
4.1. Güvenlik (Safeness) Özelliği	28
4.2. Sınırlılık (Boundness) Özelliği	28
4.3. Canlılık (Liveness) Özelliği	29
4.4. Sakınım (Conservation) Özelliği	30
4.5. Ulaşılabilirlik Ağacı	32
5. İŞ AKIŞLARININ PETRİ AĞI İLE MODELLENMESİ	37
5.1. İş Akışının Temel Kavramları	37
5.2. İş Akışlarının Petri Ağı ile Modellenme Nedenleri	39
5.3. İş Akışı Kavramlarının Petri Ağları ile Belirtimi	43
5.4. Petri Ağı ile Modellenen İş Akışlarının Analizleri	48
5.4.1 Sağlamlık (Soundness) Özelliği	48
5.4.1.1 Serbest Seçimli İş Akışı Ağları	50
5.4.1.2 İyi Yapılı İş Akışı Ağları	51

6. EĞİTİM TALEBİ İŞ AKIŞININ MODELLENMESİ VE ANALİZİ	53
6.1. Eğitim Talebi İş Akışının Petri Ağı ile Modellenmesi	53
6.2. Eğitim Talebi İş Akışı Ağıının Analizi	70
7. SONUÇ	71
KAYNAKLAR	72
ÖZGEÇMİŞ	74

KISALTMALAR

DES : Discrete Event Systems
WFMS : Workflow Management System
WFMC : Workflow Management Coalition

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1 : Basit bir otomat.....	1
Şekil 2.1 : Tanımlara ilişkin örnek petri ağı grafi.....	6
Şekil 2.2 : İşaretli petri ağı grafına örnek $x_1 = [1, 0]$	7
Şekil 2.3 : İşaretli petri ağı grafına örnek $x_1 = [2, 1]$	8
Şekil 2.4 : $x_0 = [2, 0, 0, 1]$ durumlu petri ağı.....	10
Şekil 2.5 : $x_1 = [1, 1, 1, 1]$ durumlu petri ağı.....	11
Şekil 2.6 : $x_2 = [1, 1, 0, 2]$ durumlu petri ağı.....	11
Şekil 2.7 : $x_2^1 = [0, 1, 0, 0]$ durumlu petri ağı.....	12
Şekil 3.1 : Bir imalat makinesinin petri ağı ile modeli.....	16
Şekil 3.2 : Üretici tüketici problemi (1)	17
Şekil 3.3 : Üretici tüketici problemi (2)	18
Şekil 3.4 : Üretici tüketici problemi (3)	19
Şekil 3.5 : Üretici tüketici problemi (4)	20
Şekil 3.6 : Akış diyagramı petri ağı dönüşümü	21
Şekil 3.7 : Örnek 3.1'deki programın akış diyagramı.....	23
Şekil 3.8 : Akış diyagramının petri ağı ile temsili.....	24
Şekil 3.9 : Basit bir kuyruk sisteminin petri ağı ile modeli.....	25
Şekil 3.10 : Yemek yiyen filozoflar probleminin petri ağı ile çözümü.....	26
Şekil 4.1 : Canlılık özelliğinin değişik seviyelerini gösteren petri ağı.....	30
Şekil 4.2 : Koruyucu olmayan bir petri ağı.....	31
Şekil 4.3 : Koruyucu olan bir petri ağı.....	32
Şekil 4.4 : Örnek 4.1 için ulaşılabilirlik ağacı.....	33
Şekil 4.5 : Örnek 4.2 için ulaşılabilirlik ağacı.....	34
Şekil 4.6 : Örnek 4.3 için örtüşebilirlik ağacı yapısı.....	36
Şekil 5.1 : İş akışı kavramlarının petri ağında ifadesi.....	40
Şekil 5.2 : Bir iş akışı prosedürünün olay tabanlı tanımı.....	41
Şekil 5.3 : Bir iş akışı prosedürünün durum tabanlı tanımı.....	41
Şekil 5.4 : Şikayet değerlendirilmesi iş akışı ağı.....	45
Şekil 5.5 : İş akışı ağında ardışıl yol.....	46
Şekil 5.6 : İş akışı ağında paralel yol.....	47
Şekil 5.7 : İş akışı ağında şarta bağlı yol.....	47
Şekil 5.8 : İş akışı ağında iterasyonlu yol.....	48
Şekil 5.9 : Sağlam olmayan iş akışı ağı.....	49
Şekil 5.10 : Serbest seçimli olmayan iş akışı ağı.....	50
Şekil 5.11 : İş akışı ağında iyi ve kötü yapılar.....	51
Şekil 5.12 : İyi yapıli iş akışı ağı.....	51
Şekil 6.1 : Eğitim talebi iş akışı akış diyagramı.....	55
Şekil 6.2 : Eğitim talebi iş akışının petri ağı ile modeli.....	56
Şekil 6.3 : Eğitim talebi iş akışının başlangıç durumu.....	60
Şekil 6.4 : Personelin eğitim talep ettiği durum.....	61
Şekil 6.5 : Talebin eğitim bölümüne gönderildiği durum.....	62

Şekil 6.6	: Personelin niteliklerinin değerlendirildiği durum.....	63
Şekil 6.7	: Personelin niteliklerinin uygun bulunduğu durum.....	64
Şekil 6.8	: Dışarda uygun eğitimin olup olmadığına bakılacak durum.....	65
Şekil 6.9	: Bütçenin kontrol edilme aşamasına geline durum.....	66
Şekil 6.10	: Genel müdürün değerlendirme yapacağı durum.....	67
Şekil 6.11	: Personele bilgi gönderilecek durum.....	68
Şekil 6.12	: Bitiş durumu (personele bilgi gönderilmiştir).....	69

SEMBOL LİSTESİ

P	: sonlu yer kümesi
T	: sonlu geçiş kümesi
A	: yerlerden geçişlere ve geçişlerden yerlere çizilebilen yaylar kümesi
w	: yayların ağırlık fonksiyonu
I(t_j)	: t _j geçişine yapılan giriş
O(t_j)	: t _j geçişinden yapılan çıkış
I(p_i)	: p _i yerine yapılan giriş
O(p_i)	: p _i yerinden yapılan çıkış
x	: Petri ağındaki işaretlemeye ilişkin satır vektörü

ÖZET

Bu tez çalışmasının amacı ayrık olaylı sistemlerin Petri ağı ile modellenmesine ilişkin kavramların tanıtılması ve analizine ilişkin yöntemlerin incelenmesidir. Ayrık olaylı sistemler (DES) olayların, düzensiz zaman aralıklarında meydana gelmesi ile zaman içinde gelişen ancak dinamik yapısındaki gelişimleri olaylara bağlı olan dinamik sistemlerdir. Ayrık olaylı sistem uygulamalarına günlük hayatta çokça rastlanır. Örnek olarak trafik kontrol sistemleri, esnek imalat sistemleri, bilgisayar haberleşme sistemleri, üretim bandları verilebilir. Tezde ilk önce Petri ağının temel kavramları verilmiştir. Bu temel kavramların anlaşılması için Petri ağları ile modellemeye örnek olarak bazı ayrık olaylı sistemlerin modelleri sunulmuştur. Daha sonra Petri ağlarının analizleri için geliştirilen çeşitli teknikler sunulmuştur. DES'e bir örnek olarak ayrık olaylar içeren iş akışı yönetim sistemlerinin kavramları tanıtılmıştır. Daha sonra iş akışları kavramlarının Petri ağları ile ifadeleri gösterilmiştir. İş akışı kavramları Petri ağı ile modellenmiş ve Petri ağındaki değişkenler ile ilişkilendirilmiştir. İş akışı yönetim sistemlerindeki iş mantığı Petri ağı ile grafiksel bir şekilde temsil edilebilir. İş akışı kavramları Petri ağı kavramları ile çok iyi bir şekilde tanımlanabilmektedir. Bu tanımlar WFMC (workflow management coalition) tarafından geliştirilmiştir. İş akışlarının Petri ağı ile modellenmesinde kullanılan ana yapılar gösterilmiştir. Petri ağlarına bazı şartları sağladıkları takdirde iş akışı ağı denir. Bulunması gereken bu şartlar sunulmuştur. Bu şekilde iş akışı ağına özel bir Petri ağı diyebiliriz. Çeşitli tipteki iş akışları için analiz teknikleri gösterilmiştir. Daha sonra bir şirkette kullanılan eğitim talebi iş akışının görevleri açıklanmıştır. Bu eğitim talebi iş akışının akış diyagramı çıkartılmıştır. Bu akış diyagramından eğitim talebinin Petri ağı temsili elde edilmiştir. Eğitim talebi iş akışının görevleri Petri ağı üzerinde gösterilmiştir. Daha sonra eğitim talebi iş akışının örnek bir çevrimi Petri ağı modeli üzerinde gösterilmiştir. Son olarak bu modelin analizi yapılmıştır.

MODELLING OF DISCRETE EVENT SYSTEMS BY PETRI NET

SUMMARY

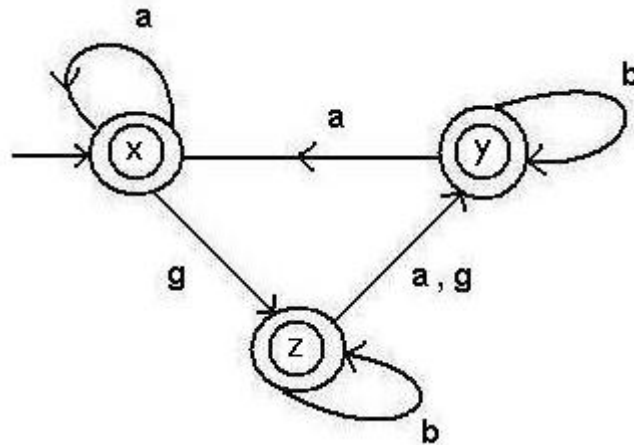
The objective of this study is introducing the Petri net concepts related to modelling and examining the methods related to analysing of the discrete event systems by Petri nets. Discrete event systems (DES) are dynamic systems which evolve in time by the occurrence of events at possibly irregular time intervals but evolution in dynamic structure of system is related to events. DES abound in real-world applications. Examples include traffic control systems, flexible manufacturing systems, computer-communications systems, production lines. Firstly we give the basic concepts of Petri nets. We present several examples of discrete event systems which have been modeled by Petri nets for understanding of these basic concepts. Then we present several techniques which have been developed for the analysis of Petri nets. As a sample for DES we introduce the concepts of workflow management system which has discrete events. Then we specify the description of workflows by Petri net. We illustrate the mapping of workflow concepts onto Petri nets. Workflow concepts are modelled by Petri net and illustrated by variables of Petri net. Work logic of workflow management systems is represented grafically by Petri net. Petri net concepts can define workflow concepts very well. These definitions are developed by WFMC(workflow management coalition). We present main structures which is used for workflow modeling by Petri net. Petri net is a workflow net when some requirements are satisfied. We present these requirements. So we can say that workflow net is a special Petri net. We show various types of workflow net analysis techniques. Then we explain the tasks of a training request workflow which is used in a company. We get the flowchart of training request workflow. We derive a Petri net representation from the flowchart of training request workflow. So we show tasks of training request workflow on the Petri nets. Then we give an example life cycle of training request workflow on the Petri net model. Finally we analyse this model.

1. GİRİŞ

1.1 Giriş ve Çalışmanın Amacı

Ayrık olaylı sistemler (DES), olayların düzensiz zaman aralıklarında meydana gelmesi ile zaman içinde gelişen ancak dinamik yapısındaki gelişimleri olaylara bağlı olan dinamik sistemlerdir. Ayrık olaylar meydana geldiğinde sistemi bir durumdan diğer bir duruma geçirirler. Petri ağları grafiksel ve matematiksel bir gösterim olarak ayrık olaylı sistemlerin modellenmesinde kullanılmaktadır. Ayrık olaylı sistemlerin modellenmesine alternatif bir yaklaşım Petri ağları tarafından sağlanmıştır.

Petri ağları teorisi ilk defa C.A. Petri tarafından 1960'ların başında geliştirilmiştir. Otomatlarla ilişkilendirilen Petri ağlarının açıkça ayrık olaylı sistemlerin (DES) geçiş fonksiyonlarını temsil ettiği görülmüştür. Bir (DES)'in davranışını tasvir etmenin formal bir yolu olarak bir dil düşünülebilir. Bir otomat, iyi tanımlanmış kuralları üzerinden bir dili ifade etme kapasitesine sahip bir araçtır. Her otomat modeli bir Petri ağı ile temsil edilebilir. Fakat her Petri ağı bir otomat ile gösterilemez. Şekil 1.1'de basit bir otomat verilmiştir.



Şekil 1.1 Basit bir otomat

Burada; $X = \{ x, y, z \}$ durum kümesini
 $E = \{ a, b, g \}$ olay kümesini, yaylar da durum geçişini
gösterir.

$$f: X \times E \rightarrow X \quad (1.1)$$

Burada geçiş fonksiyonuna $f(a, b) = c$ dersek; a durumundan, b olayı olduğu zaman, c durumuna geçiş yapılır. Şekil 1.1 için aşağıdaki ifadeleri yazabiliriz:

$$\begin{aligned} f(x, a) &= x & f(x, g) &= z \\ f(y, a) &= x & f(y, b) &= y \\ f(z, b) &= z & f(z, a) &= f(z, g) = y \end{aligned}$$

Görüldüğü gibi x 'de iken a olursa durum değişikliği olmuyor. A ve g 'nin etkisi z 'de iken aynı oluyor. Ayrıca $f(x, b)$ ve $f(y, g)$ tanımlanmamıştır.

Otomatlarda olduğu gibi, Petri ağları olayları kesin kurallara göre değerlendiren bir aygıttır. Bu ifade uygun olarak Petri ağları ile tanımlanabilir. Petri ağları ile trafik kontrol sistemleri, lojik devreler, işletim sistemleri, kaynak paylaşımı gibi konular modellenenebilir. Haberleşme protokollerinin analizinde ve modellenmesinde kullanılabilir. Petri ağları ile yazılım konuları modellenenebilir. İşletim sistemindeki kaynak paylaşımı, kilitlenme ve işlemlerin düzenlenmesi modellenenebilir. Ayrıca programların akış diyagramlarını da modelleyebilirler.

Ayrık olaylı sistemleri Petri ağı ile modellemenin 3 avantajı vardır :

- Petri ağlarının grafiksel yapısı nedeni ile ayrık olaylı sistemin anlaşılması daha kolaydır.
- Petri ağlarının kesin matematiksel tanımları vardır. Sistemin durumları ve davranışları açıkça, kesin olarak ifade edilebilir.
- Ayrık olaylı sistemlerin Petri ağları ile kurulmasını cazip kılan bir başka neden de sistemi incelemek için bir çok analiz tekniğinin geliştirilmiş olmasıdır.

Bu çalışmanın amacı ayrık olaylı sistemlerin Petri ağları ile nasıl modellendiğini incelemek ve ayrık olaylar içeren örnek bir iş akışını modellemektir. Bu amaçla 2. bölümde Petri ağı kavramları tanıtılmıştır. 3. bölümde çeşitli örneklerle bu

kavramların modeller üzerinde anlaşılması sağlanmıştır. 4. bölümde Petri ağlarının özellikleri ve analiz teknikleri incelenmiştir. 5. bölümde ayrık olaylar içeren iş akışlarının temel kavramları verilmiştir. İş akışlarının Petri ağları ile modellenme nedenleri belirtilmiştir. İş akışı kavramlarının Petri ağı ile belirtileri gösterilmiştir. Petri ağı ile modellenen iş akışlarının analizleri incelenmiştir. 6. bölümde eğitim talebi iş akışının Petri ağı ile modellenmesi ve analizi yapılmıştır.

2. PETRİ AĞLARININ TEMELLERİ

2.1 Petri Ağları Notasyonları ve Tanımları

Petri Ağları tanımı iki adımda yapılabilir. İlk önce Petri ağları grafi (Petri ağları yapısı) bir otomatın durum geçiş diyagramına benzer şekilde tanımlanır. Daha sonra bu grafa başlangıç durumu , işaretli durumlar, geçiş fonksiyonları ekleyerek bir Petri ağları modeli , ilgili dinamikleri ile beraber verilir[1].

Petri ağı grafi yerler ve geçişler olmak üzere iki tip düğüme sahiptir. Geçiş, yerler ve bunlar arasındaki kesin ilişkiler Petri ağları grafinin temel parçalarıdır. Geçişleri ve yerleri birbirine bağlayan yapıya da yay denir. Yaylar aynı tip düğümleri birbirine bağlamazlar. Yani ya yer düğümünü geçiş düğümüne ya da geçiş düğümünü yer düğümüne bağlarlar.

Petri ağındaki geçişler ayrık olaylı sistemdeki olayları temsil ederler. Bir geçişin olması için incelenen sisteme ilişkin, bazı şartların sağlanması gerekir. Bu şartlara ilişkin bilgiler tanımlı yerlerde tutulur. Yerler bir geçişe giriş olarak gösterilebilir. Bunlar bu geçişin sağlanabilmesi için gereken şartlar ile birlikte ilişkilendirilir. Yerler bir geçişin çıkışı olarak da gösterilebilir. Bunlarda geçiş sonucu oluşan şartlarla ilişkilendirilir[1].

Tanım : Petri ağı yapısı ağırlıklı, iki bölgeli (bipartite) bir (P, T, A, w) grafi olarak tanımlanabilir[1].

$P = \{ p_1, p_2, p_3, \dots, p_n \}$ sonlu yerler kümesi (graftaki düğüm tiplerinden biri)

$T = \{ t_1, t_2, t_3, \dots, t_n \}$ sonlu geçiş kümesi (graftaki diğer düğüm tipi)

$A : A \subseteq (P \times T) \cup (T \times P)$ olmak üzere grafta yerlerden geçişlere ve geçişlerden yerlere çizilebilen yaylar kümesi.

$w : A \rightarrow \{ 1, 2, 3, \dots \}$ yayların ağırlık fonksiyonu.

Doğal olarak ilgilendiğimiz Petri ağının (P, T, A, w) yalıtılmış yerleri ve geçişleri olmadığını varsayıyoruz.

Yaylar (p_i, t_j) ya da (t_j, p_i) olarak gösterilir. Yayları ilişkilendireceğimiz ağırlıkları da pozitif tam sayı olmalıdır. Bu şekilde sınırlı sayıda yerler geçişler ve yaylar bir ayrık olaylı sistemi modellemeye yeterlidir [1,2].

Petri ağı grafının bir otomattan daha detaylı olduğunu söylemiştik. Bunun ilk nedeni otomatın düğümleri tek bir kümeden (x) seçilen düğümlerdir. Petri ağlarında ise düğümler P kümesinden seçilen yerler olabilir ya da T kümesinden seçilen geçişler olabilir. İkinci nedeni durum geçiş diyagramında her olay sonucunda durum geçişini gösteren tek bir yay vardır. Petri ağı grafında ise iki düğümü birleştiren birden fazla yay olabilir ve her yaya yay sayısını belirten bir ağırlık numarası verilebilir. Bu nedenle bu yapıya çoklu graf da denilebilir[1].

Petri ağları grafında bir yerden t_j geçişine yapılan giriş $I(t_j)$ olarak, t_j geçişinden bir yere çıkışı da $O(t_j)$ olarak gösterebiliriz. Yani ;

$$I(t_j) = \{p_i \in P : (p_i, t_j) \in A\} \quad (2.1)$$

$$O(t_j) = \{p_i \in P : (t_j, p_i) \in A\} \quad (2.2)$$

Aynı şekilde bir geçişten p_i yerine yapılan girişleri $I(p_i)$, çıkışları da $O(p_i)$ olarak gösterebiliriz. Petri ağı grafını çizerken iki tip düğümü (geçiş ve yer) ayırt etmek için yerleri daire ile geçişleride dik çizgi ile göstermek uygun olur. Yerleri ve geçişleri birbirine bağlayan yaylar kümesi de A ile tanımlanır. P_i yerinden t_j geçişine çizilen bir yay $P_i \in I(t_j)$ olarak gösterilir. Eğer p_i yerinden t_j geçişine çizilen k adet yay varsa ya da k ağırlığında tek yay varsa bu da $w(p_i, t_j) = k$ olarak gösterilir. Aynı şekilde t_j geçişinden p_i yerine k yay bağlanırsa $P_i \in O(t_j)$ ve $w(t_j, p_i) = k$ ile ifade edilir. Petri ağı grafında birden fazla yayla bağlı olan

düğümüleri tek yay çizip üzerine ağırlığını yazarak göstermek uygun olur. Eğer ağırlığı belirtilmemişse 1 olduğu kabul edilir. Buradan şunu yazabiliriz [1] :

$$p_i \notin I(t_j) \text{ ise } w(p_i, t_j) = 0 \text{ 'dır.} \quad (2.3)$$

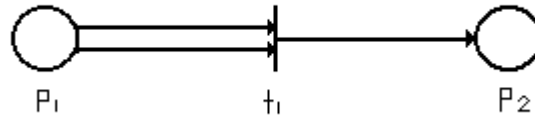
$$p_i \notin O(t_j) \text{ ise } w(t_j, p_i) = 0 \text{ 'dır.} \quad (2.4)$$

Örnek 2.1 : Bu tanımlardan aşağıdaki ifadelerle verilen grafi şu şekilde çizebiliriz [1] :

$$P = \{ p_1, p_2 \} \quad T = \{ t_1 \}$$

$$A = \{ (p_1, t_1), (t_1, p_2) \}$$

$$w(p_1, t_1) = 2 \quad w(t_1, p_2) = 1$$



Şekil 2.1 Tanımlara ilişkin örnek Petri ağı grafi

2.2 Petri Ağları İşaretleri ve Durum Uzayı

Petri ağı grafindaki geçişlerin ayrık olaylı sistemdeki olayları temsil etmesi için Petri ağındaki yerleri bu olayların meydana gelmesindeki şartlar olarak tanımlayabiliriz. Bu çerçevede şartların sağlanıp sağlanmadığını kontrol için bir mekanizmaya ihtiyaç vardır. Bu mekanizma yerlere jetonlar (token) atanarak sağlanabilir. Yerlere konulan jetonlar o yerin gerektirdiği şartlar olarak tanımlanır. Petri ağı grafinda yerlere atanan jetonlar işaretleme (marking) olarak tanımlanır[1,2]. Bu özellikleri de göz önüne

almak için Petri ağı tanımında ek büyüklüklere gerek olduğundan tanım aşağıdaki gibi düzenlenmiştir.

Tanım : Bir Petri ağları grafında (P , T , A , w) işaretli bir Petri ağını N ile gösterirsek

$N = (P , T , A , w , E , L , X_0 , X_m)$ olur. Burada ;

E : geçişleri etiketlemek için olay kümesi

$L : T \rightarrow E$ geçiş etiketleme fonksiyonu

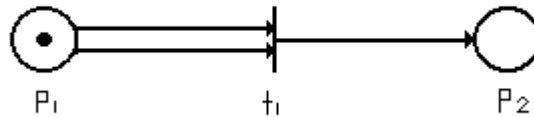
$X_0 \in N^n$ ağın ilk durum kümesi (her yerdeki jetonların ilk sayısı)

$$X : P \rightarrow N = \{ 0, 1, 2, \dots \} \quad (2.5)$$

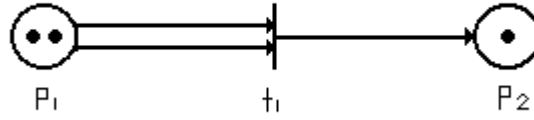
$$X = [x(p_1) , x(p_2) , x(p_3) , \dots , x(p_n)] \quad (2.6)$$

$X_m \subseteq N^n$ ağın işaretlenmiş durum kümesidir.

Petri ağındaki yer sayısı n olmak üzere işaretleme x satır vektörü olarak tanımlanır. Bu vektörün i. elemanı p_i yerindeki jeton sayısını gösterir. Petri ağı grafında bir jeton ilgili yerin üzerinde siyah bir nokta olarak gösterilir.



Şekil 2.2 İşaretli Petri ağı grafına örnek $x_1 = [1 , 0]$



Şekil 2.3 İşaretli Petri ağı grafına örnek $x_1 = [2 , 1]$

Jetonların Petri ağı içindeki dağılımı petri ağının o anki durumunu tanımlar. Yerlere atanan jeton sayısı sıradan pozitif tamsayılardır. N yere sahip bir Petri ağının durum uzayı pozitif tam sayı elemanlı n boyutlu vektörler ile ifade edilebilir. $X = N^n$

Buraya kadar ki tanımlamalar Petri ağlarının durum geçiş mekanizmasını tanımlamaya yeterli değildir. Ayrık olaylı sistemi Petri ağları ile modelleyebilmek için bir notasyona daha ihtiyaç vardır. Petri ağları yapısındaki durum geçiş mekanizması için ‘olanaklı geçişler’ kavramı da kullanılmalıdır[1,2].

Tanım : Olanaklı geçiş

Petri ağında bir geçişin $t_j \in T$ mümkün olabilmesi için ;

$$p_i \in I(t_j) \text{ için } X(p_i) \geq w(p_i, t_j) \quad (2.7)$$

şartı sağlanmalıdır. Yani t_j geçişinin yapılabilmesi için p_i ‘deki jeton sayısı en az p_i ’yi t_j ’ye bağlayan yay sayısı kadar olmalıdır. (t_j ‘ye girişi olan bütün p_i ’ler için) Önceden belirtildiği gibi bir geçişin olabilmesi için gerekli şartlar yerler ile ilişkilendirilmiştir. Bütün şartlar sağlanınca bu geçiş mümkün olur. Jetonlar bu şartların sağlanması sürecinde kullanılan karar verme mekanizmalarıdır. Bir Petri ağındaki olanaklı geçişler o otomasyonun durumundaki aktif olay kümesine eşittir [1,2].

2.3 Petri Ağları Dinamikleri

Petri ağında durum geçiş mekanizması jetonların hareket ettirilmesiyle sağlanır. Bu şekilde Petri ağının durumu değişir. Bir geçiş mümkün olduğunda buna canlanabilir

ya da oluşabilir denir. (Petri Ağları literatüründe (firing) canlanabilir terimi kullanılır.) Petri ağının durum geçiş fonksiyonu oluşabilecek olanaklı geçişlerin durumuna göre tanımlanır[1].

Tanım : Durum geçiş fonksiyonu

(P , T , A , w) Petri ağının $t_j \in T$ geçişi için durum geçiş fonksiyonu $f : N^n \times T$ aşağıdaki koşul altında

$$p_i \in I(t_j) \text{ için } x(p_i) \geq w(p_i, t_j) \quad (2.9)$$

tanımlanabilir.

Eğer $f(x, t_j)$ tanımlanırsa $x^1 = f(x, t_j)$ şu şekilde olur : $i = 1, 2, \dots, n$ olmak üzere

$$x^1(p_i) = x(p_i) - w(p_i, t_j) + w(t_j, p_i) \quad (2.10)$$

(2.10)'dan anlaşılabacağı gibi durum geçiş fonksiyonu sadece olanaklı geçişler için tanımlanabilir. Bu da otomasyonda ki mümkün olay ifadesine eşittir. (2.10)'dan görüleceği gibi bir sonraki durum ilgili geçişin giriş ve çıkış yerlerine ve bu yerlerden geçişe bağlanan yayların ağırlığına bağlıdır. Eğer p_i, t_j geçişine giriş yeri ise p_i 'yi t_j 'ye bağlayan yayın ağırlığı kadar p_i jeton kaybeder. Eğer p_i, t_j 'nin çıkış yeri ise t_j 'yi p_i 'ye bağlayan yayın ağırlığı kadar p_i jeton kazanır. Petri ağında jeton sayısı geçişler sonucu korunmayabilir[1,2].

(2.10)'den görüleceği gibi aşağıdaki eşitsizlikler mümkündür :

$$\sum_{p_i \in P} w(t_j, p_i) > \sum_{p_i \in P} w(p_i, t_j) \quad (2.11)$$

$$\sum_{p_i \in P} w(t_j, p_i) < \sum_{p_i \in P} w(p_i, t_j) \quad (2.12)$$

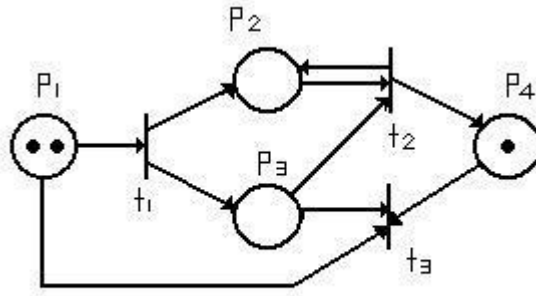
Bu eşitsizlikler bir yerden bir geçişe çizilen yayların sayısının, aynı geçişten aynı yere çizilen yayların sayısından büyük ya da küçük olabileceğini söylüyor.

$x^1 = f(x, t_j)$, x 'den daha çok ya da daha az jetona sahip olabilir. Genel de birkaç geçiş sonunda $x = [0, 0, \dots, 0]$ durumu da mümkündür ya da çok fazla geçiş sonunda belli yerlerdeki jeton sayısı çok fazla artabilir.

Örnek 2.2 : Bir Petri ağının durumunun geçişler sonucu değişmesini bir örnekle gösterelim[1].

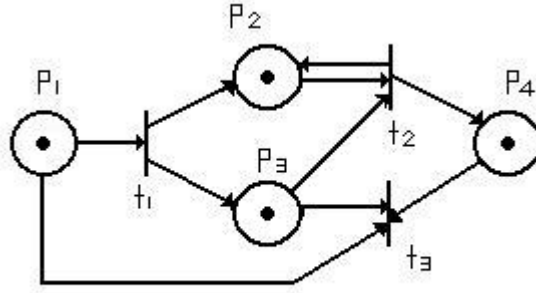
İlk durumu $x_0 = [2, 0, 0, 1]$ olan bir Petri ağını inceleyelim.

Şekil 2.4' en görüldüğü gibi sadece t_1 geçişi mümkündür.



Şekil 2.4 $x_0 = [2, 0, 0, 1]$ durumlu Petri ağı

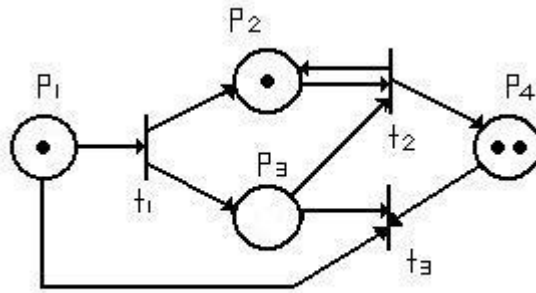
Çünkü geçiş için sadece bir jetonun geçmesine gerek vardır ve $x_0(p_1) = 2$ 'dir. Yani $x_0(p_1) \geq w(p_1, t_1)$ ile (2.9)'da ki koşul sağlanmaktadır. Petri ağında görüldüğü gibi t_1 geçişi yapıldığında p_1 'den bir jeton eksilir ve p_2, p_3 yerlerine birer jeton yerleştirilir. (2.10) denklemi uygulandığında da $x_1 = [1, 1, 1, 1]$ durumu elde edilir. (Şekil 2.5)



Şekil 2.5 $x_1 = [1 , 1 , 1 , 1]$ durumlu Petri ağı

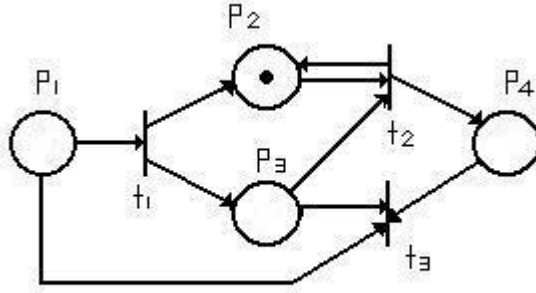
Bu durumda üç geçiş de olanaklıdır. $(t_1 \ t_2 \ t_3)$ Bir sonraki adımda t_2 geçişi yapılırsa p_2 ve p_3 'den birer jeton taşınır. Çıkış yerleri p_2 ve p_4 ' tür.

Bu şekilde bir jeton hemen p_2 'ye geri döner çünkü $p_2 \in I(t_2) \cap O(t_2)$ 'dir. Bir jetonda p_4 'de eklenir. Yeni durum $x_2 = [1 , 1 , 0 , 2]$ olur . (Şekil 2.6)



Şekil 2.6 $x_2 = [1 , 1 , 0 , 2]$ durumlu Petri ağı

Bu durumda t_2 ve t_3 geçişleri mümkün değildir. Sadece t_1 geçişi olanaklıdır. Şekil 2.5 'de üç geçişin mümkün olduğu durumda t_2 geçişi yerine t_3 geçişini yaparsak p_1 , p_3 , p_4 yerlerinden birer jeton taşınır. Görüldüğü gibi t_3 geçişinden sonra çıkış yeri yoktur yeni durum $x_2^1 = [0 , 1 , 0 , 0]$ şeklinde olur. (Şekil 2.7)



Şekil 2.7 $x_2^1 = [0 , 1 , 0 , 0]$ durumlu Petri ağı

Bu durumda geçiş artık olanaklı değildir ve başka durum değişimi olmaz. Yani $[0 , 1 , 0 , 0]$ durumu Petri ağının çıkmazıdır (deadlock). Örnekte jeton sayısı da korunmamaktadır. X_0 durumu 3 ; x_1 , x_2 durumları 4 ve x_2^1 durumu da 1 jetona sahiptir.

2.4 Durum denklemleri

Durum değişimleri Petri ağları ile her bir duruma bir yeni graf karşı düşürülerek nasıl ifade edilir bir önceki bölümde incelendi. Bu gösterim görüldüğü gibi çok pratik değildir. Dinamik yapının incelenmesinde daha uygun bir yaklaşım durum değişimini matematiksel ifade etmektir. Bir yerin durum değerinin bir geçiş olduğunda nasıl değiştiğini 2.10 denkleminde elde edebiliriz. Bir t_j geçişi öncesindeki durumu ;

$$X = [x(p_1) , x(p_2) ,x(p_n)] \quad (2.13)$$

ile gösterirsek bir sonraki durum

$$X^1 = [x^1(p_1) , x^1(p_2) ,x^1(p_n)] \quad (2.14)$$

olur.

j. geçişin olduğu sıradaki durumu ifade etmek için m boyutlu bir canlanabilme (firing) vektörü tanımlayabiliriz.

$$U = [0 , ..., 0 , 1 ,0] \quad j \in \{ 1, 2, ..., m \} \quad (2.15)$$

Petri ağlarının sistem matrisi (m x n) boyutlu A ;

$$A_{ji} = w(t_j , p_i) - w (p_i , t_j) \quad (2.16)$$

olarak tanımlanır[1].

Bu ifade (2.10)'daki $x(p_i)$ 'nin ağırlık farkına benzerdir. Bu tanımlardan sonra vektör durum denklemini yazabiliriz.

$$x^1 = x + u.A \quad (2.17)$$

Bu ifade u girişi sonucunda durum geçiş işlemini tanımlar. Daha önce tanımlanmış olan $f(x, t_j)$ geçiş fonksiyonunun da

$$f(x, t_j) = x + u.A \quad (2.18)$$

eşit olduğu görülür. Fonksiyonda t_j ifadesi sıfırdan farklı olan u'daki j. değeri gösterir. Durum denklemi geçiş işlemleri sonucunda değişen Petri ağının durumunu tanımlamaya yarar.

Örnek 2.3 : Şekil 2.4'de verilen ilk durumu $x_0 = [2 , 0 , 0 , 1]$ olan Petri ağını inceleyelim[1]. (2.16) 'da ki ifade ile sistem matrisi

$$A = \begin{bmatrix} -1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & -1 & -1 \end{bmatrix} \text{ olur.}$$

(2.18) denklemi kullanılarak t_1 geçişi olduğu zaman $u = [1 \ 0 \ 0]$ olur ;

$$\begin{aligned} X &= [2 \ 0 \ 0 \ 1] + [1 \ 0 \ 0] \begin{bmatrix} -1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & -1 & -1 \end{bmatrix} \\ &= [2 \ 0 \ 0 \ 1] + [-1 \ 1 \ 1 \ 0] = [1 \ 1 \ 1 \ 1] \end{aligned}$$

elde edilir. Sonucun bir önceki örnekle aynı olduğu görülür.

Aynı şekilde t_2 geçişi sonucunda elde edilen x_2 durumu

$$\begin{aligned} X_2 &= [1 \ 1 \ 1 \ 1] + [0 \ 1 \ 0] \begin{bmatrix} -1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & -1 & -1 \end{bmatrix} \\ &= [1 \ 1 \ 0 \ 2] \text{ olur.} \end{aligned}$$

Görüldüğü gibi Petri ağında izlenen yol geçiş sırasına göre $\{t^1, t^2, \dots\}$

(t^k k. oluşan geçiş) durumların sıralanmasıdır. $\{x_0, x_1, \dots\}$

x_0 durumu verildiğinde sırayla tüm durumlar :

$$x_{k+1} = f(x_k, t_k) = x_k + u_k A \quad (2.19)$$

denklemi ile üretilebilir (u_k , k. geçişin canlanacağını belirtir.) [1].

3 . PETRİ AĞLARI İLE MODELLEME

Petri ağlarının esas tasarlanma amacı modellemedir. Bir çok sistem (yazılım , donanım , fiziksel sistemler , sosyal sistemler) Petri ağları ile modellenebilir. Petri ağlarında 2 önemli kavram vardır : olaylar ve durumlar. Olaylar sistemde meydana gelebilen hareketlerdir , faaliyetlerdir. Bu hareketler sistemin durumuyla kontrol edilir. Sistemin durumu da belli şartlar kümesi ile tanımlanır. Şartlar sistemin durumunun lojik ifadesidir , doğru ya da yanlış olabilir. Olayların meydana gelmesi için gerekli şartların sağlanması gerekir. Bunların daha iyi anlaşılması için imalat makinası örneği verilebilir[2].

3.1 İmalat Makinasının Petri Ağı ile Modellenmesi

Sipariş geldikçe malzeme üreten bir imalat makinesini Petri ağı ile modelleyebiliriz. Makine sipariş gelene kadar bekliyor, gelince çalışıyor, bitirince de dağıtım için gönderiyor[2].

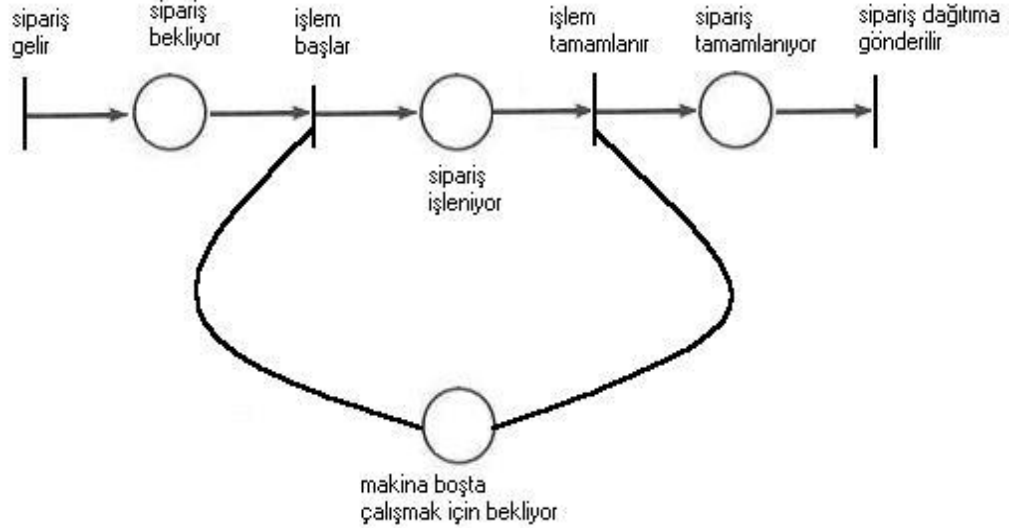
Sistemin durumları :

- a- Makine bekliyor.
- b- Sipariş geldi ve makine bekliyor.
- c- Makine sipariş üzerinde çalışıyor.
- d- Sipariş tamamlandı.

Sistemin olayları :

- a- Siparişin gelmesi.
- b- Makinenin çalışmaya başlaması.
- c- Makinenin siparışı bitirmesi.
- d- Siparişin dağıtıma gönderilmesi.

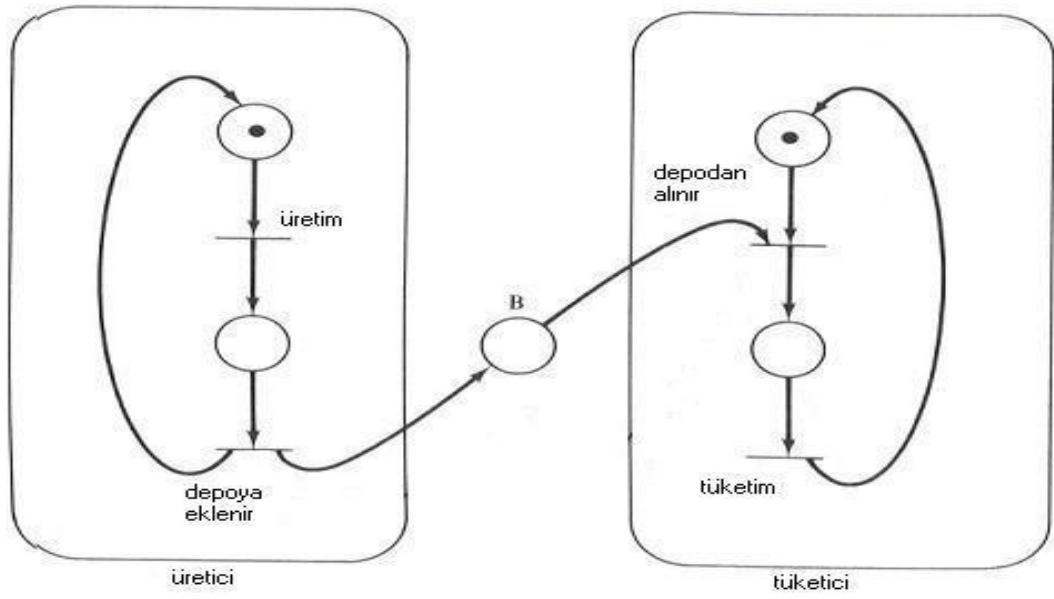
Durumlar yerler ile olaylar da geçişler ile gösterilmiştir. Bir olayın meydana gelmesi ona karşı gelen geçişin sağlanması ile mümkündür. Sistemin hangi durumda olduğu o yerde bir jetonla gösterilir.



Şekil 3.1 Bir imalat makinesinin Petri ağı ile modeli

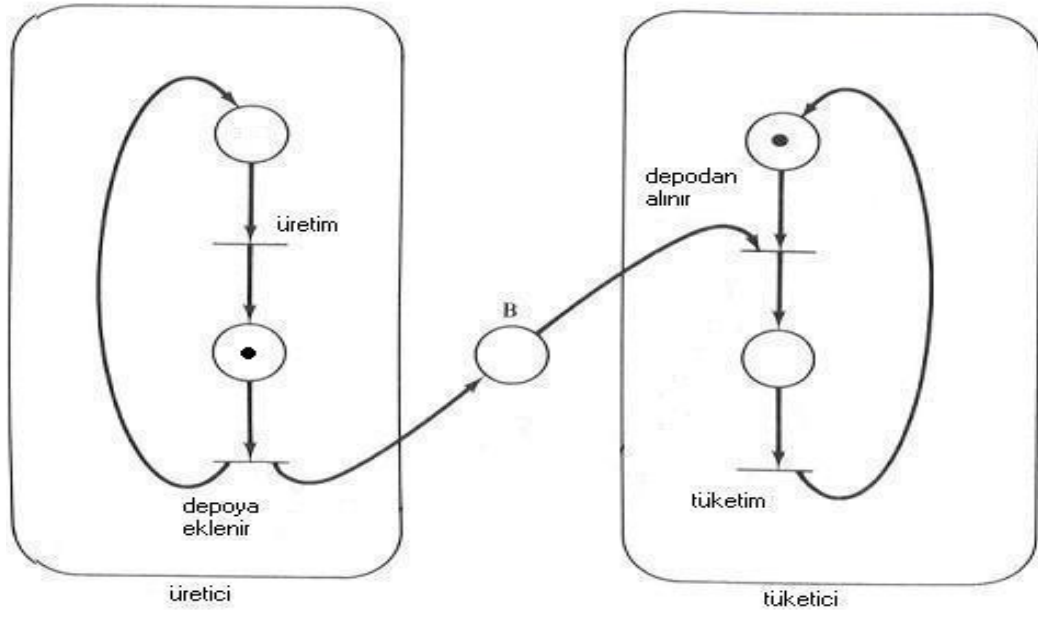
3.2 Üretici Tüketici Probleminin Modellenmesi

Üretici tüketici problemi paylaştırılmış bir alan içerir. Paylaştırılan bu alan depo olarak kullanılır. Üretici bir mal üretir ve bunu depoya koyar. Tüketici ise mal depoya konuncaya kadar bekler. Depoya mal konunca onu alır ve tüketir. Bu problemin Petri ağı modeli ile çözümü aşağıdaki şekillerdeki gibidir[2].



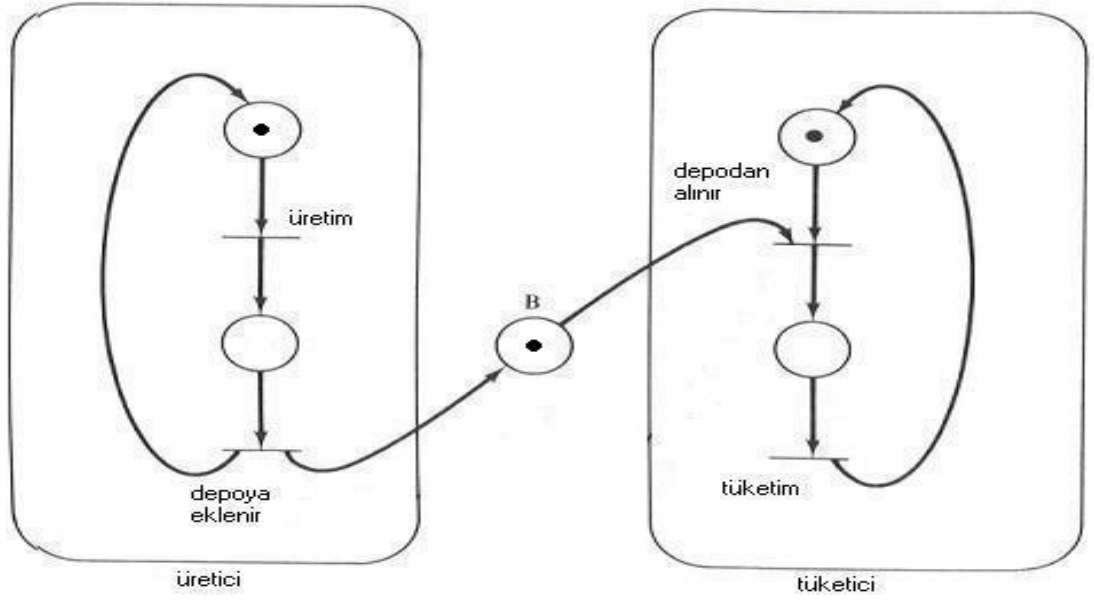
Şekil 3.2 Üretici tüketici problemi (1)

Şekil 3.2’de üretici boşta ve üretmek için beklemektedir. Tüketici B deposuna malın gelmesini beklemektedir.



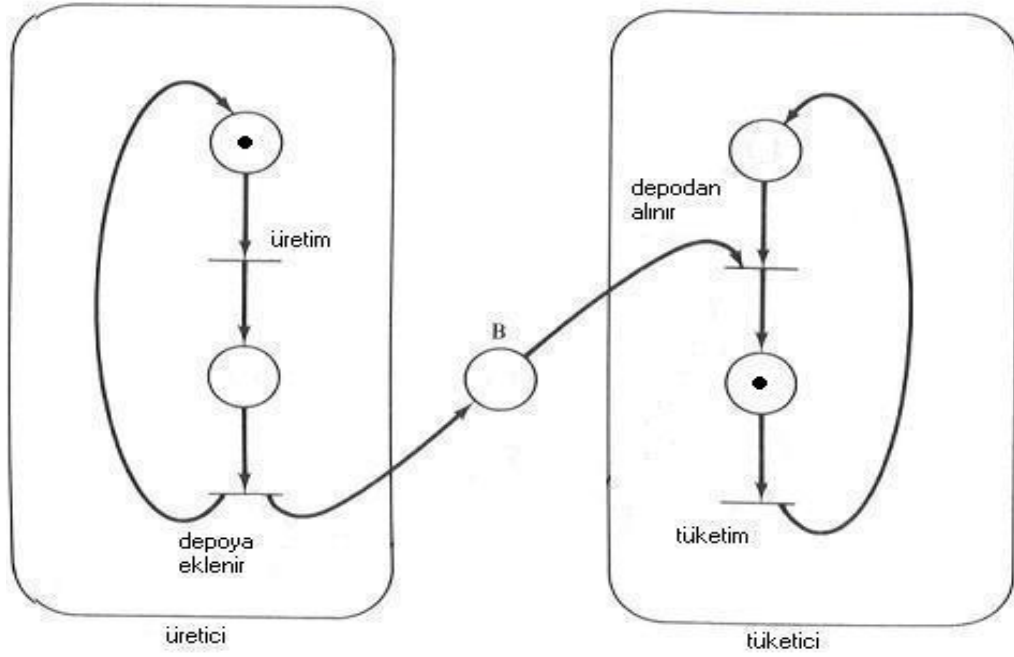
Şekil 3.3 Üretici tüketici problemi (2)

Şekil 3.3’de mal üretilmiştir fakat depoya henüz gönderilmemiştir. Bu yüzden hala tüketici beklemektedir , depodan mal alamamaktadır.



Şekil 3.4 Üretici tüketici problemi (3)

Depoya mal eklenmiştir. Artık tüketicide boşta olduğundan depodan mal alma olayı gerçekleşebilir. Depodan mal almak için gerekli şartlar sağlanmaktadır.

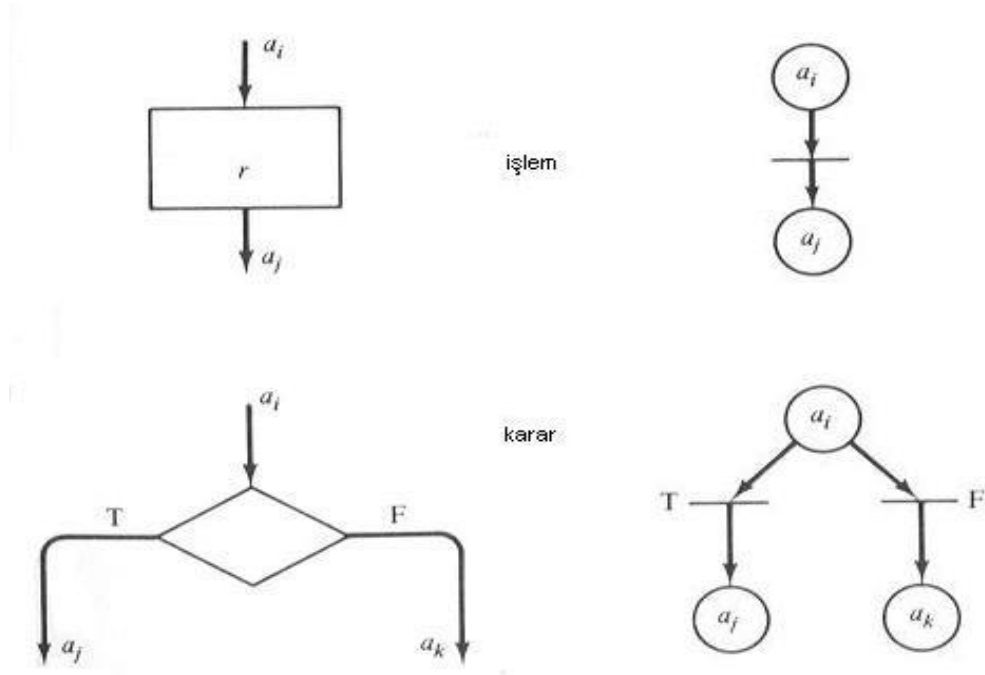


Şekil 3.5 Üretici tüketici problemi (4)

Şekil 3.5’de tüketici malı depodan almıştır ve tüketmektedir. Tüketici bu durumda meşguldür ve depoda da mal yoktur.

3.3 Program Modellenmesi

Petri ağının en çok kullanıldığı yerlerden biride programlardır. Petri ağları programların kontrol yapılarını en iyi şekilde temsil edebilirler. Komut sırasını , bilgi akışını ve hesapları modelleyebilirler. Programların akış diyagramları Petri ağına yaklaşık benzer. Akış diyagramında da iki tip düğüm vardır : Karar merkezleri ve işlemler. Bunların Petri ağına dönüşümü şekil 3.6 ‘da ki gibi olur[2].



Şekil 3.6 Akış diyagramı Petri ağı dönüşümü

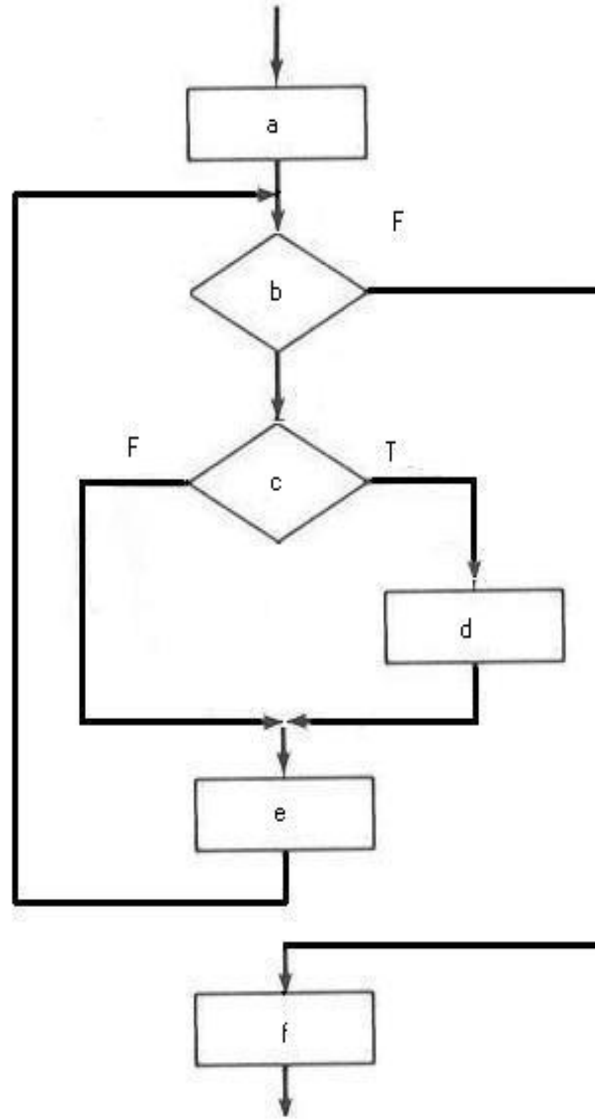
Akış diyagramındaki oklar Petri ağındaki yerlere karşı gelir. Jetonun o yerde bulunması bir sonraki komutun işlenmeye hazır olduğunu gösterir. Geçişler programın kararlara ve işlemlere bağlıdır. Bu kurallara göre aşağıdaki örnek programın akış diyagramını ve Petri ağını çizelim.

Örnek 3.1 : Örnek bir programın Petri ağı ile modellenmesi [2] :

```
Begin
  input (y1);
  input (y2);
  y3 = 1 ;
  while y1 > 0
    do begin
      if odd(y1)
        then begin
          y3 = y3 * y2 ;
          y1 = y1 - 1 ;
        end ;
      y2 = y2 * y2 ;
      y1 = y1 - y2 ;
    end;
  output(y3);
End;
```

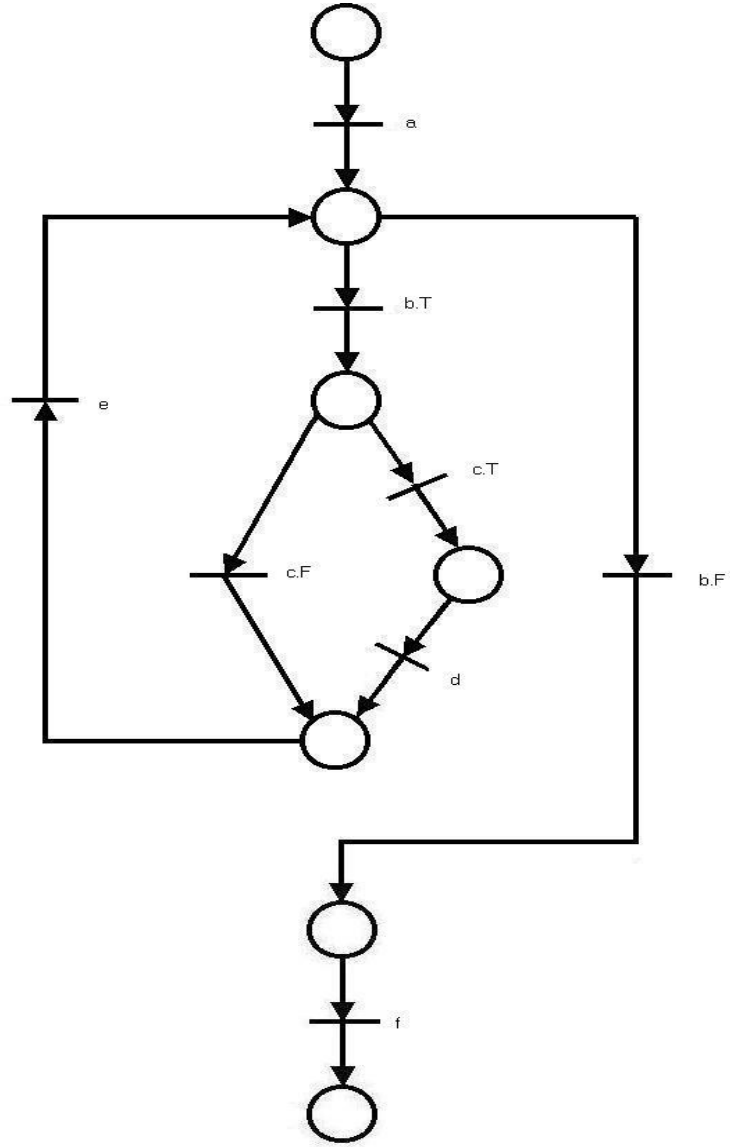
Bu programın işlemleri ve karar merkezleri şekil 3.7'deki akış diyagramında gösterilmiştir.

- a - input(y1); input(y2); y3 = 1
- b - $y1 > 0$?
- c - odd(y1) ?
- d - $y3 = y3 * y2$; $y1 = y1 - 1$;
- e - $y2 = y2 * y2$; $y1 = y1 / 2$;
- f - output(y3) ;



Şekil 3.7 Örnek 3.1'deki programın akış diyagramı

Şekil 3.7'deki akış diyagramının Petri ağı ile temsili, gerekli dönüşümleri yaptıktan sonra şekil 3.8 'deki gibi olur.



Şekil 3.8 Akış diyagramının Petri ağı ile temsili

3.4 Kuyruk Sistemleri İçin Petri Ağı Modeli

Kuyruk işlemleri de Petri ağı yapısı kullanılarak modellenenbilir[1].

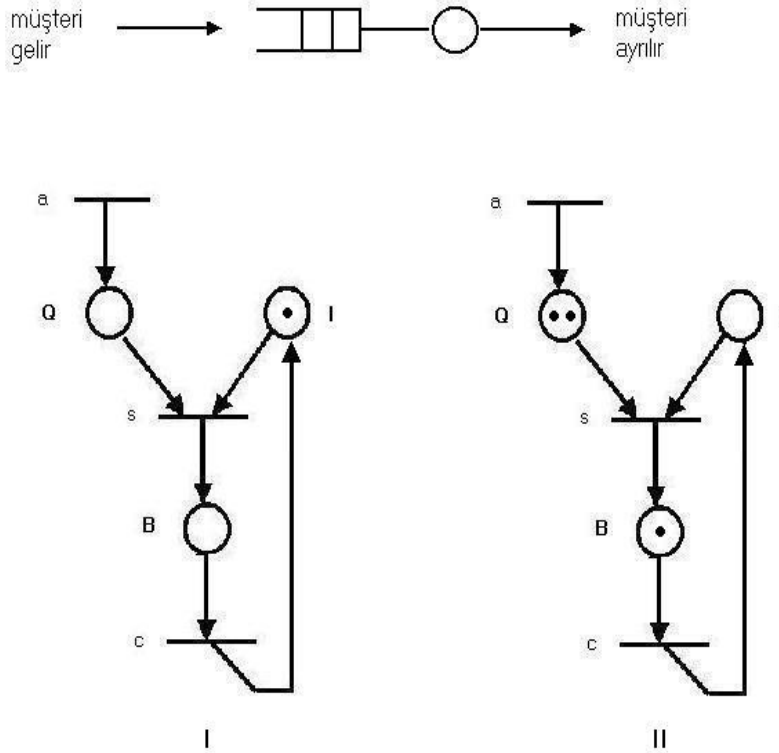
Bu sistem için 3 olay (geçiş) göz önüne alırsak :

a : müşteri gelir.

s : hizmet başlar.

c : hizmet tamamlanır ve müşteri ayrılır.

Bu şekilde geçiş kümesi $T = \{ a, s, c \}$ olur. Burada a geçişi kendiliğinden olur ve geçiş yeri gerektirmez. Diğer yandan s geçişi iki şarta bağlıdır : müşterilerin sırada olması ve sistemin boşta olması (o an hizmet yapılmaması). Bu şartları temsil etmek için iki giriş yeri kullanılır. Q yeri kuyrukta bekleyenleri belirtir ve I yeri sistemin boşta olduğunu belirtir. Sistemin meşgul olduğunu belirtmek için de B yeri kullanılır. Bu şekilde yer kümesi $P = \{ Q, I, B \}$ olur. Şekil 3.9'da tanımlanan geçişler ve yerlerin olduğu basit bir kuyruk sisteminin Petri ağı modeli verilmiştir. Şekil 3.9 'da I ile gösterilen Petri ağının başlangıç durumu $[0, 1, 0]$ 'dır. Bu ağın $\{ a, s, a, a, c, s, a \}$ geçişleri sonundaki hali şekil 3.9'da II ile gösterilmiştir.

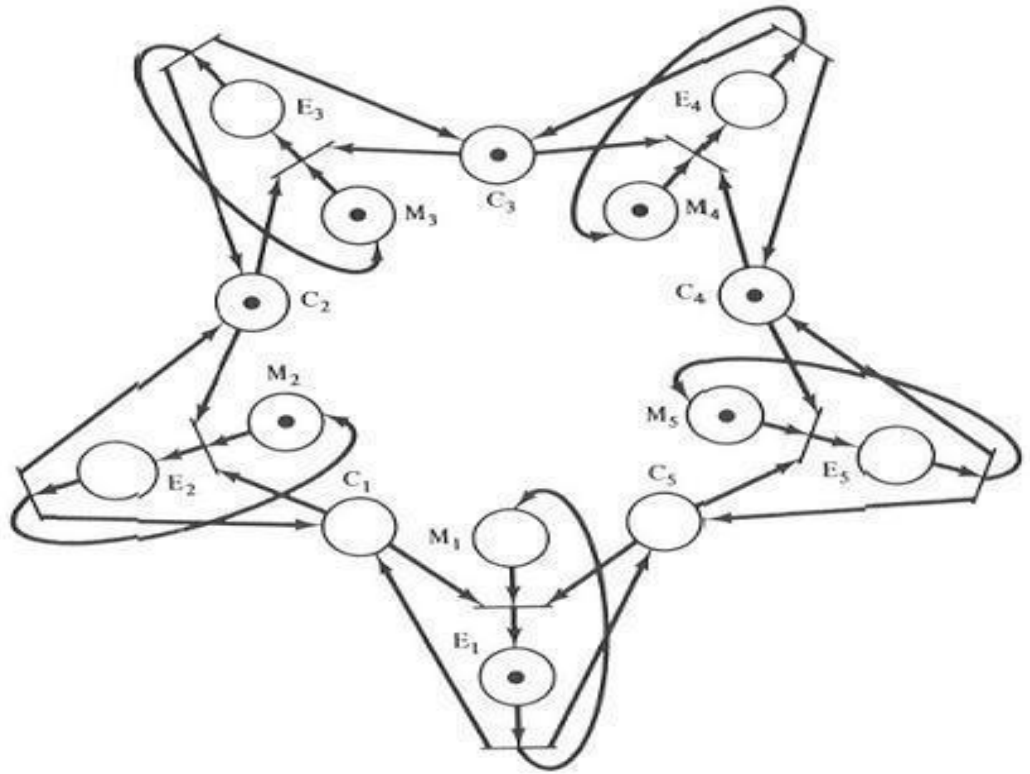


Şekil 3.9 Basit bir kuyruk sisteminin Petri ağı ile modeli

Q 'da jeton bulunmaması kuyruğun boş olduğunu belirtir. I yerinde bir jeton olması sistemin boşta , hizmete hazır beklemekte olduğunu ifade eder. B yerinde bir jeton olması da sistemin müşteriye hizmet ettiğini , meşgul olduğunu gösterir.

3.5 Yemek Yiyen Filozoflar Problemi (Dining Philosophers Problem)

Bu problem 1968'de Dijkstra tarafından ortaya atılmıştır. 5 filozof alternatifli olarak Çin yemeği yiyip düşünüyorlar. Yuvarlak bir masa etrafında oturuyorlar ve aralarında birer çubuk var. Yemek yiyebilmeleri için sağ ve sol ellerinde 1 çubuk olması gerekiyor. Eğer tüm filozoflar sol ellerinde çubukla beklerlerse aç kalırlar. Bu problemin Petri ağı ile modeli şekil 3.10 'daki gibidir [2].



Şekil 3.10 Yemek yiyen filozoflar probleminin Petri ağı ile modeli

C' ler çubukları temsil ediyor. Her filozof M ve E olmak üzere iki yer ile temsil ediliyor.

M : Düşünme modu

E : Yemek yeme modu

Bir filozofun M modundan E moduna geçmesi için iki elinde de çubuk olmalı.

Şekil 3.10 ‘dan görüldüğü 1 numaralı filozof yemek yerken E1 durumunda C1 ve C5 boş olacağından 2 ve 5 numaralı filozoflar yemek yiyemez. 1 numaralı filozof yemek yeme işini bitirince E1 durumundan sonraki geçiş sağlanır ve C1 , M1 , C5 yerleri 1 jeton kazanırlar. Bu durumda 5 numaralı filozof ile 2 numaralı filozof artık yemek yiyebilirler. Şekil 3.10 ‘dan görüldüğü gibi herhangi bir anda sadece 2 filozof yan yana olmamak şartıyla aynı anda yemek yiyebilirler.

4. PETRİ AĞLARININ ANALİZLERİ

Petri ağları ile çok çeşitli sistemlerin (özellikle interaktif olayların meydana gelebileceği sistemler) modellenenebileceğini gösterdik. Petri ağlarının en önemli gücü birbirini takip eden olaylar (concurrent) sergileyen sistemleri modelleyebilmesidir [2]. Eş zamanlı oluşma da doğal ve uygun bir yolla modellenir.

Petri ağları ile modellenen bir sistemin analizi de gereklidir. Modellenen sistemin davranışlarını inceleme açısından bu analizler çok önemlidir. Petri ağlarının analizleri için birçok teknik geliştirilmiştir fakat bir çok problem de hala açıktır. Geliştirilen analiz tekniklerinin kullanışlılığını değerlendirirken ne tip problemlerin çözüme kavuşturulması gerektiğini göz önüne almalıyız. Bu bölümde Petri ağları literatüründe göz önüne alınmış özellikleri ve soruları inceleyeceğiz. Bu özelliklerin çoğu kaynak paylaşımı yapılan sistemlerin Petri ağları ile modellenmesinden ortaya çıkmıştır. Daha iyi ve verimli kaynak paylaşımı yapmak için Petri ağlarının bu özellikleri incelenmiştir.

4.1 Güvenlik (safeness) Özelliği

Donanım cihazlarını modelleyen Petri ağlarında güvenlik özelliği çok önemlidir. Petri ağında bir yerdeki jeton sayısı 1'i kesinlikle geçmiyorsa o yer güvenlidir. O yerde ki jeton sayısı 1 ya da 0 'dır. Bu şekilde yerler mantıksal bir durumu ifade edebilir(Jeton varsa doğru yoksa yanlış şeklinde) [1].

Tanım : Petri ağında ki bir yer $p_i \in P$ $N = (P, T, I, O)$ eğer tüm yeni durumlar için $x^1 \in R(N, x)$ $x^1(p_i) \leq 1$ ise , ilk durumu x ile güvenlidir.

Bir Petri ağında içindeki tüm yerler güvenli ise o Petri ağıda güvenlidir [2].

4.2 Sınırlılık (boundness) Özelliği

Kaynak paylaştıran sistemler Petri ağı ile modellendiğinde jetonlar genellikle müşterileri temsil eder. Örnek olarak kuyruk modelleyen sistemde (şekil 3.9) Q yeri sıraya giren müşterileri temsil eder. Burada kuyrukta bekleyen müşterilerin çok

birikmesi istenmeyen bir durumdur çünkü sıranın kendisine gelmesini bekleyen müşteriler uzun süre beklerler. Klasik sistem teorisinde bir durum değişkeninin çok fazla büyümesi o sistemdeki kararsızlığa işaret eder. Aynı şekilde Petri ağıının bir yerinde biriken jetonlar sistemi kararsızlığa götürür. Sınırlılık özelliği bir yerdeki jeton sayısının kesinlikle pozitif bir tam sayıyı geçmeyeceğine işaret eder[1,2].

Tanım : N Petri ağında x_0 başlangıç durumundaki $p_i \in P$ yeri eğer ulaşılabilir tüm durumlar için $x \in R(N)$ $x(p_i) \leq k$ ise k-sınırlıdır denir [1].

Eğer Petri ağındaki tüm yerler sınırlı ise o ağa sınırlıdır denir. Eğer bir yer 1-sınırlı ise o yer aynı zamanda güvenlidir. Şekil 3.9'daki (kuyruk modeli) Petri ağı sınırlı değildir çünkü Q yerindeki jetonlar keyfi olarak artmaktadır. Petri ağı ile modellenen bir ayrık olaylı sistemde ağın sınırlı olup olmadığı kontrol edilir ve bir sınırlılığa karar verilir. Eğer sınırlılık sağlanamıyorsa model değiştirilmelidir. Çünkü kararsızlığa neden olabilir [1,2].

4.3 Canlılık (liveness) Özelliği

Genellikle Petri ağındaki geçişlerin canlanıp canlanamayacağı ile ilgilenilir. Eğer geçiş hiç canlanmayacaksa ilgili olay meydana gelmeyecek demektir. Bu aynı zamanda kilitlenme (deadlock) özelliği ile de ilgilidir[1,2].

Tanım : X_0 başlangıç durumlu bir N Petri ağında , x_0 durumundan ulaşılan yeni durumlarda her zaman canlanabilir bir geçiş bulunursa o ağa canlıdır denir [1].

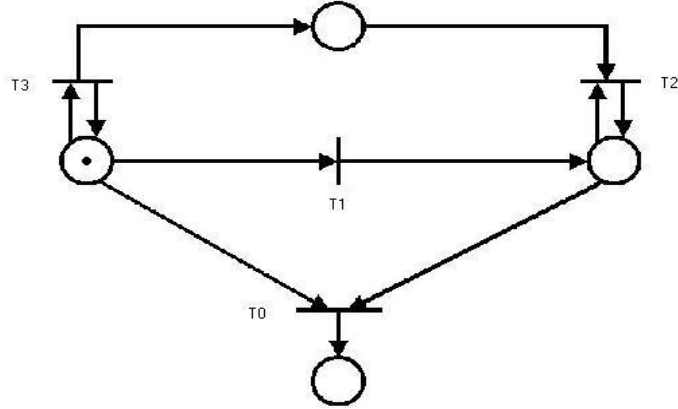
Bu özellik sistemin davranışıyla ilgili önemli bir durumdur.

Canlılık ayrıca 4 seviyede sınıflandırılır [2] :

X_0 başlangıç durumundaki Petri ağındaki bir geçiş :

- Eğer x_0 durumundan hiç canlanamıyorsa ölü ya da L0-canlıdır.
- Eğer x_0 durumundan sadece 1 kez canlanabiliyorsa L1-canlıdır.
- Eğer en az k kere canlanabiliyorsa L2-canlıdır.
- Eğer sonsuz canlanabiliyorsa L3-canlıdır.
- Eğer x_0 durumundan ulaşılabilen tüm durumlar için L1-canlı ise buna L4-canlıdır denir.

Bu seviyeleri şekil 4.1’deki Petri ağında incelersek :



Şekil 4.1 Canlılık özelliğinin değişik seviyelerini gösteren Petri ağı

T_0 geçişi hiçbir zaman canlanamaz , ölüdür. T_1 geçişi 1 kez canlanabilir , bu 1. seviyede bir geçiştir. T_2 ‘nin canlanma sayısı t_3 ‘e bağlıdır. T_3 ‘ün canlanma sayısı kadar t_2 canlanabilir. T_2 canlanmadan önce t_1 ‘in canlanması gerekir. T_1 canlanınca artık t_2 nin canlanma sayısında t_3 ‘e bağlı olarak sabitlenmiş olur. Daha önceden t_3 kaç kere canlanmışsa o kadar t_2 geçişi yapılabilir. Bu yüzden t_2 geçişi 2. seviye bir geçiştir fakat 3 seviye değildir. T_3 geçişi sonsuz canlanabilir. Bu yüzden 3.seviye bir geçiştir. T_3 geçişi 4. seviye olamaz çünkü t_1 geçişi 1 kere yapıldıktan sonra artık t_3 gerçekleşmez.

4.4 Sakınım (conservation) Özelliği

Petri ağlarında önemli bir özellik de jeton sayısının korunmasıdır. Örneğin kaynak kullanımının modellendiği sistemlerde kaynak isteği, kaynağın alıkonulması ve serbest bırakılması gibi durumlar olur. Kaynakların jetonla ifade edildiği bu tip problemlerde koruyucu özellik çok önemlidir. Kaynaklar yaratıldığında veya tüketildiğinde jetonlarda yaratılmalı ve yok edilmelidir. Bu şekilde jeton sayısı ağ içinde sabit kalır[2].

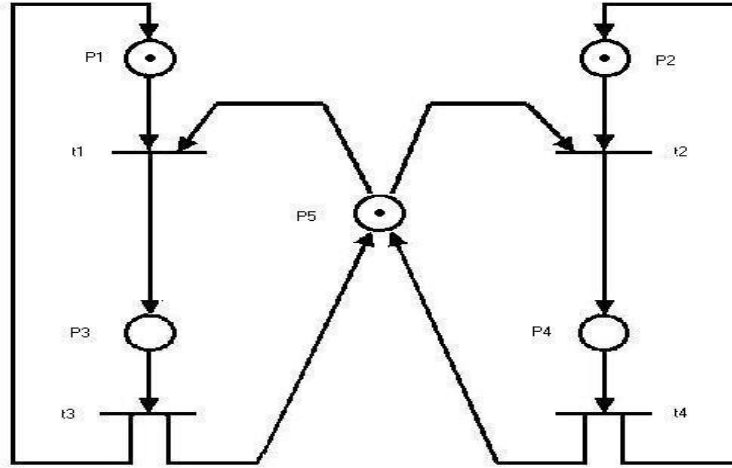
Tanım : x_0 başlangıç durumlu bir N Petri ağında tüm durumlar için $x \in R(N)$ jeton sayısı sabitse o ağ koruyucudur[1].

$$\sum_{p_i \in P} x^1(t_j, p_i) = \sum_{p_i \in P} x(p_i, t_j) \quad (4.1)$$

Kesin koruyucu özelliği çok güçlü bir ilişkidir. Çünkü her geçişin giriş sayısı ile çıkış sayısı eşit olmalıdır [2].

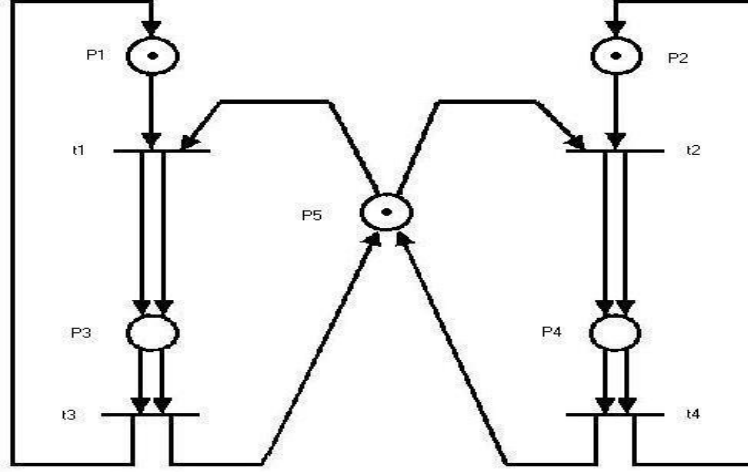
$$|I(t_j)| = |O(t_j)| \quad (4.2)$$

Eğer bu sağlanamazsa ağ içinde jeton sayısı değişir.



Şekil 4.2 Koruyucu olmayan bir Petri ağı

Şekil 4.2 koruyucu değildir çünkü t_1 ve t_2 geçişi jeton sayısını 1 azaltmaktadır. t_3 ve t_4 geçişleri jeton sayısını 1 artırmaktadır. Bu Petri ağını şekil 4.3'e çevirirsek ağ koruyucu olur.



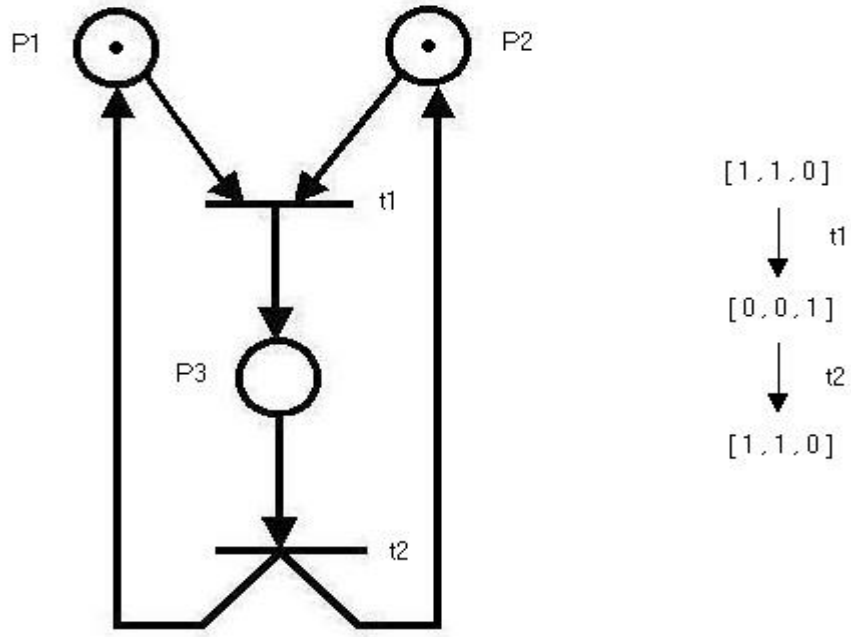
Şekil 4.3 Koruyucu olan bir Petri ağı

4.5 Ulaşılabilirlik Ağacı

Petri ağlarının analiz tekniklerinden biride ulaşılabilirlik (reachability) ağacıdır. Ulaşılabilirlik ağacında düğümler Petri ağının durumlarını , yaylarda geçişleri temsil eder[1,2]. Ulaşılabilirlik ağacı ile Petri ağında, varsa eğer hiç canlanmayan geçişlerin, kilitlenmenin ve tekrar eden durumların açığa çıkması sağlanır. Ulaşılabilecek tüm durumlar elde edilir.

Örnek 4.1 Basit bir ulaşılabilirlik ağacı [1]

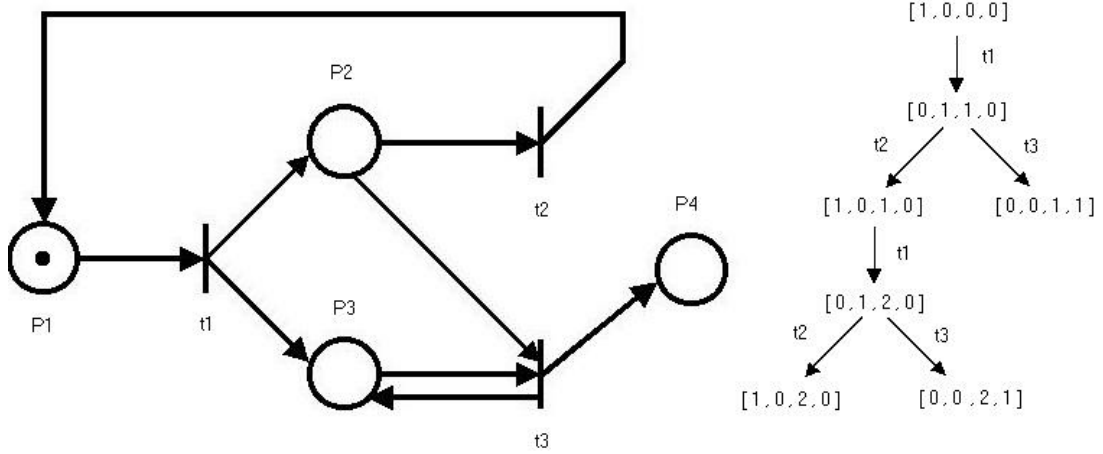
Şekil 4.4 'deki Petri ağının ulaşılabilirlik ağacını çizmek istersek , ilk durum $[1 , 1 , 0]$ ağacın kökünü temsil eder. İlk durumdan ulaşılabilirlik diğer tüm durumları çıkarırsak ağaç elde edilir. Bu örnekte ilk canlanacak geçiş t_1 'dir ve bir sonraki durum $[0 , 0 , 1]$ olur. Bu halde t_2 canlanır ve bir sonraki durum $[1 , 1 , 0]$ olur. Bu aynı zamanda ilk durum olduğundan (ilk durum tekrar ettiğinden) burada durulur.



Şekil 4.4 Örnek 4.1 için ulaşılabilirlik ağacı

Örnek 4.2 : Sonsuz ulaşılabilirlik ağacı[1]

Şekil 4.5 'deki Petri ağının ilk durumu $[1, 0, 0, 0]$ olsun. Ulaşılabilirlik ağacı bu ilk durumda başlar ve sadece t_1 canlanabilir. Bir sonraki durum $[0, 1, 1, 0]$ olur. Bu durumda t_2 veya t_3 canlanabileceğinden ağaçta 2 dal oluştururuz ve sırasıyla bu 2 durumu ($[1, 0, 1, 0]$, $[0, 0, 1, 1]$) belirtiriz. $[0, 0, 1, 1]$ durumunda artık geçiş canlanamaz. Bu yüzden bu duruma çıkmaz ya da uç durum denir. Solda ki dal t_1 geçişine izin verir ve yeni durum $[0, 1, 2, 0]$ olur. Bu durumda artık t_2 veya t_3 canlanabilir ve bir sonraki durumlar $[1, 0, 2, 0]$, $[1, 0, 2, 1]$ olur. Bu kadarı ağacın örneğini çıkarmak için yeterli olur çünkü oluşan her sağ dal sonlanır ve p_3 yerinde yeni bir jeton oluşur. Bu şekilde p_3 yerinin sınırsız olduğunu görürüz. Yani ulaşılabilir durumlar ve ulaşılabilirlik ağacı sonsuzdur. Ağaca bakıldığında $[1, 0, 2, 0]$ durumunun kendinden önceki aynı yol üzerinde $[1, 0, 1, 0]$ durumuna benzediği sadece p_3 yerinde ki jeton sayısının 1 fazla olduğu görülür. Burada $[1, 0, 2, 0]$ durumu, $[1, 0, 1, 0]$ durumunu kapsar diyebiliriz.



Şekil 4.5 Örnek 4.2 için ulaşılabilirlik ağacı

Örneklerde de görüldüğü gibi ulaşılabilirlik ağacını çıkarmak kolaydır. Petri ağında oluşabilecek durumları , çıkmazları problemleri görmek için ağacın sonlu bir temsilini çıkarmak faydalı olur. Sonsuz ulaşılabilirlik ağacının sonlu versiyonuna örtüşebilirlik (coverability) ağacı denir [1,2].

Ulaşılabilirlik ağacını çıkarmak için aşağıdaki algoritma ve notasyonlar kullanılabilir [1].

Algoritmada kullanılan notasyonlar :

1. Kök düğüm : Petri ağına başlangıç durumuna yani ağacın ilk düğümüne kök düğüm denir. Örnek 4.2'nin kök düğümü $[1, 0, 0, 0]$ 'dır.
2. Uç düğüm : Hiçbir geçişin canlanamadığı düğümüne uç düğüm denir. Şekil 4.5'de ağaçtaki $[0, 0, 1, 1]$ düğümü bir uç düğümdür.
3. Eş düğüm : Ağaçta bulunan bir düğümün aynısı tekrar ederse bu düğümüne eş düğüm denir. Örneğin şekil 4.4 'deki $[1, 0, 0]$ düğümü kök düğüm ile aynı olduğundan eş düğümdür.
4. Düğümün kapsanması : $x = [x(p_1), \dots, x(p_n)]$ ve $y = [y(p_1), \dots, y(p_n)]$ durumları ulaşılabilirlik ağacında iki durum olsun. Eğer aşağıdaki koşullar sağlanırsa x, y 'yi kapsar denir ve $x \geq_d y$ şeklinde gösterilir.

Koşullar :

A - $x(p_i) \geq y(p_i)$, $i = 1, \dots, n$ bütün i 'ler için

B - $x(p_i) > y(p_i)$, $i = 1, \dots, n$ en az bir i için sağlanırsa.

Örneğin şekil 4.5 'de $[1, 0, 2, 0] >_d [1, 0, 1, 0]$ diyebiliriz. Fakat

$[1, 0, 2, 0]$ durumu , $[0, 0, 1, 1]$ durumunu kapsamaz.

5. w Sembolü : Durum işaretlemesinde sınırsız bir yeri sonsuz temsil etmede w sembolü kullanılır. Ulaşılabilirlik ağacında bir düğüm kapsanması varsa w sembolü kullanılır.

Eğer $x >_d y$ ise yani tüm i 'ler için $x(p_i) > y(p_i)$ ise $x(p_i)$ değeri yerine w kullanılır. w ile işaretlenmiş bir yere jeton eklenmesinin bir etkisi olmaz çünkü $k = 0, 1, 2, \dots$ için $w + k = w$ demektir. Örnek olarak şekil 4.5 'de $[1, 0, 1, 0] >_d [1, 0, 0, 0]$ için $[1, 0, 1, 0]$ yerine $[1, 0, w, 0]$ diyebiliriz.

Örtüşebilirlik ağacı oluşturma algoritması [1] :

Adım 1: Başlangıç durumu belirlenir. $x = x_0$

Adım 2: Her yeni düğüm için geçiş fonksiyonu $f(x, t_j)$ kullanılarak durumlar elde edilir. $T_j \in T$

Adım 2.1: Eğer $f(x, t_j)$ tüm $t_j \in T$ için tanımsızsa (yani x durumundan hiçbir geçiş canlanmıyorsa) x bir uç düğümdür.

Adım 2.2: Eğer $f(x, t_j)$ tüm $t_j \in T$ için tanımlıysa yeni düğüm yaratılır.

$x^1 = f(x, t_j)$ Sonra :

Adım 2.2.1: Eğer bazı yerler p_i için $x(p_i) = w$ ise $x^1(p_i) = w$ yapılır.

Adım 2.2.2: Eğer kökten başlayarak bir x düğümünden bir y düğümüne yol varsa ve $x^1 >_d y$ koşulu sağlanıyorsa tüm p_i 'ler için $x^1(p_i) = w$ yapılır.

Adım 2.2.3 : Koşul sağlanmıyorsa $x^1 = f(x, t_j)$ ile geçiş sağlanarak devam edilir.

Adım 3 : Eğer yeni düğümler uç düğümse ya da eş düğümse durulur.

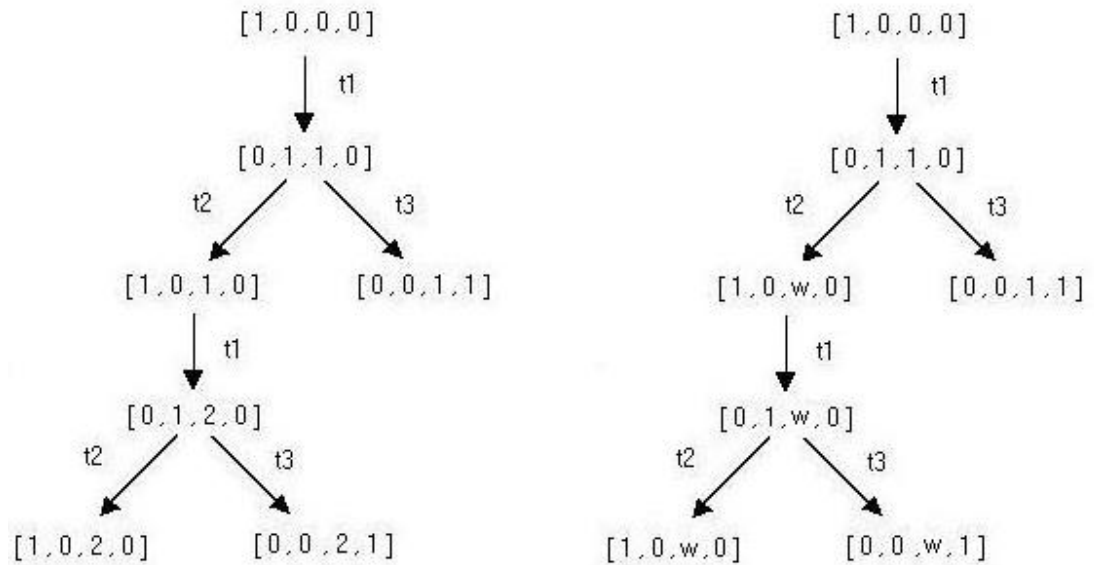
Bu algoritmayı bir örnekle uygulayalım.

Örnek 4.3 : Sonsuz ulaşılabilirlik ağacının örtüşebilirlik ağacı [1]

Şekil 4.5'deki ilk durumu $x_0 = [1, 0, 0, 0]$ olan Petri ağının örtüşebilirlik ağacını çıkartalım. İlk geçiş canlandığında $x^1 = f(x_0, t_1) = [0, 1, 1, 0]$ elde edilir. Eğer bu durumda bir sonra t_3 canlanırsa $[0, 0, 1, 1]$ uç düğümü elde edilir. Eğer t_2 canlanırsa yeni düğüm $[1, 0, 1, 0]$ elde edilir.

Adım 2.2.2 izlenirse $[1, 0, 1, 0] >_d [1, 0, 0, 0]$ olduğu görülür.

Bu yüzden $[1, 0, 1, 0]$ yerine $[1, 0, w, 0]$ diyebiliriz. Bir sonra t_1 canlanırsa adım 2.2.1 'den sembol w değişmeden aynı kalır. Yeni düğüm $[0, 1, w, 0]$, $[0, 1, 1, 0]$ düğümüne eş düğümdür. Artık p_3 yeri w ile gösterilir. T_3 geçişi ile devam edersek $[0, 0, w, 1]$ uç düğümü elde edilir. T_2 geçişi ile de eş düğüm $[1, 0, w, 0]$ elde edilir. Bu şekilde örtüşebilirlik ağacı (şekil 4.6) tamamlanmış olur.



Şekil 4.6 Örnek 4.3 için örtüşebilirlik ağacı yapısı

5. İŞ AKIŞLARININ PETRİ AĞI İLE MODELLENMESİ

İş akışı yönetimi, uygulanabilir bilgisayar bilimlerinde önemli bir araştırma alanıdır. Bir iş akışı yönetim sistemi (WFMS – workflow management system) , yönetimi , modellemeyi ve iş akışlarının yürütülmesini destekleyen bir yazılım sistemidir. İş akışı çalıştırılabilir bir iş prosesidir. Prosesin tanımına iş akışının belirtimi denir. İş akışı yönetim sistemi , iş akışı belirtimine göre yaratılan bir çok örnek iş akışı sistemde çalışırken, bunların yürütülmesinden sorumludur [3].

5.1 İş Akışının Temel Kavramları

Tanım : Görev , kaynak , kaynak sınıfı , doküman [3]

İş akışının anahtar kavramı görevdir. Görev ; önceden belirlenmiş bir zamanda , bir ya da birden fazla kaynakla yapılabilecek bir iş parçasıdır. Görev , daha küçük görev parçalarına bölünemeyecek en küçük parçadır. Kaynak, kendisi ya da başka kaynaklarla birlikte görevi yerine getirir. Kaynaklar görevi icra ederken başlangıcından sonuna kadar görevi yapmaya tahsis edilirler. Kaynak sınıfı bir kaynaklar kümesidir. Her görev ilgili kaynak sınıfından kaynaklara ihtiyaç duyar. Doküman , bir göreve giriş ya da çıkıştır ve iş akışı yönetim sistemi ile yakından ilgilidir. Bir iş parçası olan görev , bir hastanın doktor tarafından muayenesi olabilir, bir sekreter tarafından yazılan mektup olabilir ya da bilgisayar tarafından bir denklem sisteminin köklerinin hesaplanması olabilir. İş akışı yönetim sistemi tarafından bakıldığında görev harici bir iş olarak değerlendirilir. İş akışı yönetim sistemi sadece yönetim konuları ile ilgilenir. Hangi kaynak hangi görevi yürütecek , kaynağa görevi yaparken hangi dokümanlar gerekli gibi. Kaynaklar otomatik cihaz (örneğin yazıcı, yazılım programı) ya da insan (tahsildar, sekreter) olabilir.

Görevi yürütmek için kaynak sağlanırken ilk önce hangi kaynak sınıfından kaynak atanacağı belirlenir. Bu nedenle her görevin kaynak sınıfı belirlenir ve iş akışı yönetim sistemi, yürütmesi için ilgili kaynak sınıfından bir ya da daha fazla kaynak seçer. Dokümanlar iş akışı yönetim sistemi içinde farklı görevler arasında haberleşmeyi sağlayan mesaj araçlarıdır. Dokümanlar iş akışı yönetim sistemi

tarafından yine harici olarak görülürler ve görev yürütülürken diğer fiziksel araçlar gibi muamele görürler.

Tanım : Kaynak yöneticisi [3]

Görevler için ihtiyaç duyulan kaynaklar bir ya da daha fazla kaynak yöneticisi tarafından yönetilirler. Kaynak yöneticisi kaynakların görevlere tahsis edilmesini kontrol ederler. Kaynak yöneticisi kişi, bilgisayar sistemi ya da bunların birleşimi olabilir. Bir kaynak aynı anda bir ya da daha fazla göreve atanmış olabilir. Bir görev birden fazla kaynağa ihtiyaç duyarsa bununla kaynak yöneticisi ilgilenir.

Tanım : İş , iş durumu , iş kimliği , iş özellikleri [3]

Bir iş, verilen prosedüre göre bir miktar çalışmanın yürütülmesinin modellenmesi işlemidir. Bu şekilde iş bir çeşit işlemdir ve olayların sırası olarak veya durumların sırası olarak karakterize edilebilir. İş durumu , zaman içinde bazı anlardaki işin halidir. İş durumu o anda işle ilgili geçmiş ve gelecek bilgileri taşır. İşin yönlendirilmesi ile ilgili olarak iş kimliği ve iş özellikleri kullanılır. Genellikle bir iş , bir prosedür ile ilgilenir fakat aynı prosedür ile bir çok iş de ilgileniyor olabilir. Bir iş başlangıcından sonra ve son duruma ulaşmadan önce aktif durumdadır. Her aktif işin durumu iş akışı yönetim sistemi tarafından kaydedilir.

Tanım : Prosedür , kontrol aktivitesi [3]

Bir prosedür sıralanmış görevler ve kaynak sınıfı kümesidir. Kontrol aktivitesi , prosedürlere göre ve görevlerin sıralamasına göre işlerin yönlendirilmesini belirler. Öncelik ilişkisine göre kontrol aktivitesi , görev ve prosedürler sıralanır. Bir prosedürün bir giriş ve bir çıkışı vardır. Bu giriş ve çıkışlar dokümanlardır ve görevler arasındaki haberleşmeyi sağlarlar. Eğer birden fazla giriş ve çıkış gerekirse kontrol aktivitesi bu dokümanları birleştirir veya dağıtır.

Tanım : İş akışı , İş akışı yönetim sistemi [3]

Bir iş akışı sıralanmış işler kümesidir. İşleri sıralamanın birkaç yolu vardır. Örneğin x ve y işleri için y başlamadan önce x 'in hazır olması gerekebilir. Bir iş akışı yönetim sistemi iş akışlarını yöneten bir bilgisayar sistemidir. Aşağıdaki fonksiyonları sağlar :

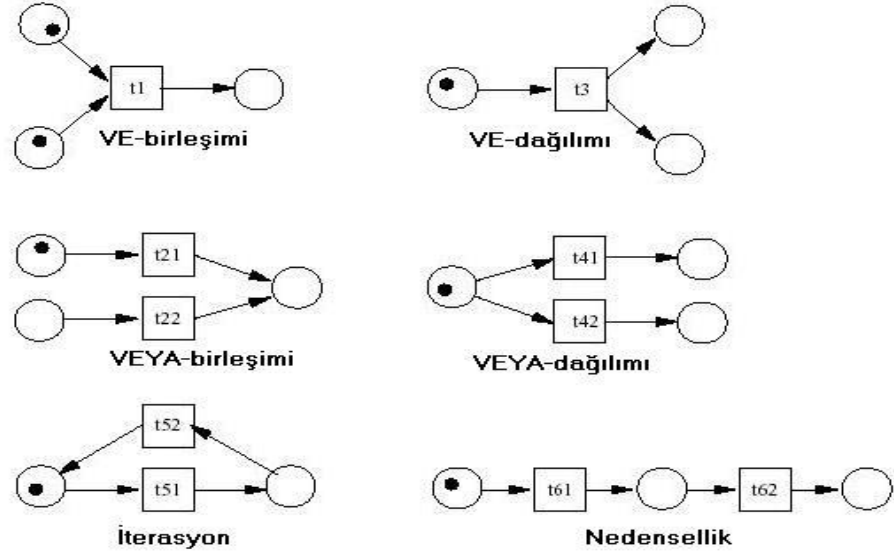
- Görevlerin , prosedürlerin ve işlerin tanımı
- İşlerin görevlerini yerine getirmek için ihtiyaç duyulan bilginin işlenmesi
- Kaynakların yönetimi
- Prosedürler ve kaynaklar arasında iş bilgilerinin yönlendirilmesi
- İş akışının denetimi

5.2 İş Akışlarının Petri Ağı ile Modellenme Nedenleri

İş akışı için Petri ağı tabanlı bir çalışma, iş akışının modellenmesine ve analizine kavramsal standartlar getirerek yardımcı olabilir. Petri ağı tabanlı iş akışı yönetim sistemleri kullanmanın 3 önemli nedeni vardır [4,5,6] :

1 - Grafıksel yapıya rağmen içimsel anlamı olması

Petri ağı tabanlı iş akışı yönetim sistemi kullanmanın ilk nedeni , iş mantığını içimsel ve grafıksel bir şekilde temsil edebilmesidir. Klasik Petri ağının anlamı, semantiği genel olarak tanımlanabilir. İş akışı yönetim sistemi tarafından tanımlanan kavramların Petri ağı ile modellenebileceği gösterilebilir. Şekil 5.1'de iş akışı yönetim sistemi tarafından tanımlanan 6 iş akışı kavramının Petri ağında nasıl ifade edildiği gösterilmiştir. Görevler geçişlerle ve nedensel ilişkiler de yerler ile modellenir [4,5].



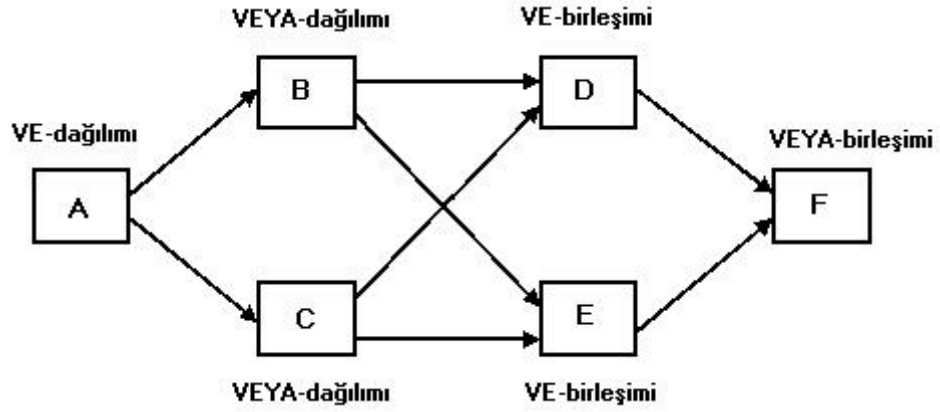
Şekil 5.1 İş akışı kavramlarının Petri ağında ifadesi

Bu yapılar 1996 yılında iş akışı yönetim koalisyonu (WFM - workflow management coalition) tarafından tanımlanmıştır. WFM 1993 yılında iş akışı standartlarını kurmak ve geliştirmek için kurulmuş uluslar arası bir organizasyondur [4].

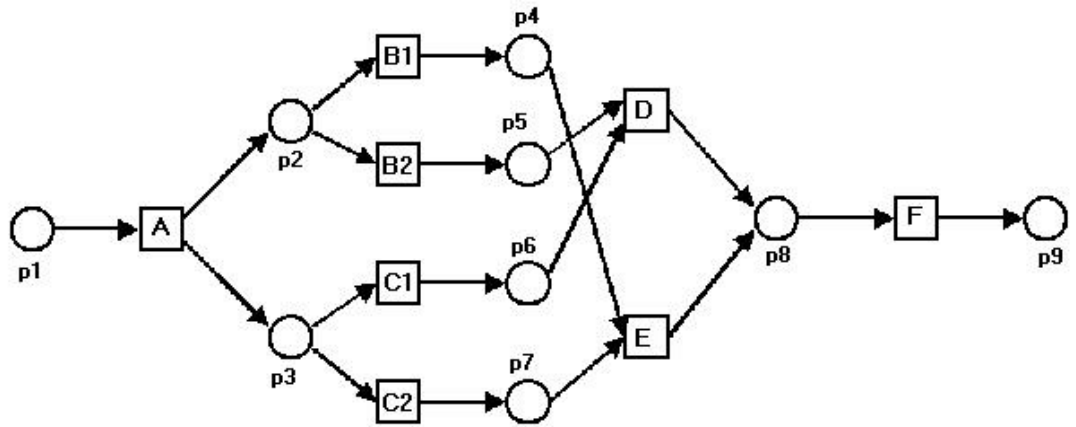
Bazı iş akışı yönetim sistemleri anlamı % 100 açık anlaşılır olmayan farklı yapılar kullanmaktadır. Bu gibi problemlerden kaçınmak içinde biçimsel anlamı açık olan Petri ağı tabanlı iş akışı yönetim sistemleri kullanılmalıdır [5,6].

2 - Olay tabanlı yerine durum tabanlı olması

Petri ağları bir iş akışının durumlarını açıkça modelleyebilir ve durum tabanlı bir modelleme tekniğidir. İş akışlarını modellemede sıkça kullanılan veri akış diyagramları da olay tabanlı bir modelleme tekniğidir. Olay tabanlıda geçişler açıkça modellenir fakat görevler arasındaki durumlar gizlenir. Şekil 5.2 'deki diyagram bir iş akışı prosedürünü tanımlar . A , B , C , D , E , F görevleri açıkça temsil edilmektedir. Fakat durumlar görülmemektedir. Eğer bu olay tabanlı tanımlı, durum tabanlı Petri ağına dönüştürürsek şekil 5.3 elde edilir. Şekil 5.3 'de görevler geçişlerle ve durumlarda yerler ile modellenmiştir. Durum tabanlı tanımları kullanmanın avantajları sırasıyla verilmiştir.



Şekil 5.2 Bir iş akışı prosedürünün olay tabanlı tanımı



Şekil 5.3 Bir iş akışı prosedürünün durum tabanlı tanımı

Durum tabanlı bir tanım , bir görevin yapılabilir olması ile o görevin icra edilmesini açık bir şekilde ayırmaya olanak sağlar. Çünkü bir görevin yapılmasının mümkün olması , o görevin hemen yapılacak olması anlamına gelmez. Bu ayrımı yapmak çok önemlidir. Bunu örneklendirmek için görevlerin başlamasına neden olan durumları (trigerring) daha detaylı bir şekilde incelemek gerekir. Bir görev örneğinin başlamasına neden olacak durumlar gerçekleştiğinde artık görev icra edilebilir. İş akışlarında 4 çeşit başlamaya neden olan durum tanımlanabilir [5] :

- Otomatik : İnsan hareketi gerektirmeyen bir başvuru ile görevin başlatılması sağlanabilir. Görev mümkün olan sırası geldiğinde başlatılır.
- Kullanıcı : Görev bir katılımcı kişi ile başlatılabilir. İş akışı yönetim sistemlerinde başlamaya hazır görevler bulunabilir. Kullanıcı bunlardan seçtiklerini başlatabilir.
- Mesaj : Dış olaylar bir görevin başlamasını sağlayabilir. Örnek mesajlar telefon, faks, elektronik posta olabilir.
- Zaman : Mümkün olan bir görevin belli bir saatte başlaması da sağlanabilir. Önceden belirlenmiş bir saatte görev başlatılır. Örneğin dokümanları silme görevi her 15 saatte bir yapılabilir.

Bu 4 çeşit başlamaya neden olan durumdan sadece otomatik olanda görevin mümkün olması ile hemen başlaması görülür. Diğer başlamaya neden olan durumlarda aradaki durumları açıkça ortaya koymak çok önemlidir.

Durumları açıkça kesin olarak modellemenin bir başka sebebi de aynı zamanda oluşan (concurrent) görevlerin olma olasılığıdır. Eğer iki görevin belli bir zamanda yapılması artık mümkün olursa bu iki görev aynı zamanda oluşan eş zamanlı görevlerdir. Fakat böyle bir durumda bunlardan sadece birinin başlatılması gerekebilir. Bir çok iş akışı yönetim sistemi bu durumu modelleyemez. Çünkü ara durumlar gizli tutulur , gösterilemez. Sonuç olarak yapılması mümkün olan görevlerin başlatılması gerekir ve başlatılıncaya kadar o durumda kalabilir, bekletilebilir. Petri ağı ile bu durumlar kolayca modellenebilir [6].

Bazen belli bir durumu geri almak gerekebilir. Bir çok olay tabanlı iş akışı yönetim sistemi için bu karşılaşılan zor bir durumdur. Bu durum örneğin kişinin görev listesinden yapılması mümkün olan görevlerin çıkarılması olabilir. Petri ağı tabanlı bir iş akışı yönetim sisteminde geri alma çok kolaydır. Sadece o yerdeki jetonlar kaldırılır ve buna karşı gelen yere aktarılır [6].

Günümüzde bir iş akışı yönetim sistemi genellikle tek bir departmanda kullanılmaktadır. Gelecekte bir çok iş akışı sisteminin birleşimi olan iş akışı yönetim sistemlerinin kullanılması bir gerçek olacak. Sonuç olarak sık sık iş akışları arasında durumların birinden diğerine aktarılması ihtiyacı olabilecektir. Bu durum tabanlı iş

akışları için çok kolaydır. İş akışının ilgili durumundaki bütün jetonlar diğer iş akışındaki karşı gelen duruma aktarılır. Olay tabanlı iş akışı yönetim sistemlerinde durum değiştirmek çok zordur. Bu nedenlerle olay tabanlı iş akışı yönetim sistemi yerine durum tabanlı iş akışı yönetim sistemi kullanmanın bir çok avantajı vardır. Olay tabanlı sistemler birçok kısıtlamalara maruz kalırken durum tabanlı sistemler çevreyi takip eder ve yönlendirir [5].

3 - Analiz tekniklerinin fazla olması

Petri ağlarının incelenmesinde birçok analiz tekniğinin olması Petri ağı tabanlı iş akışı yönetim sistemlerinin lehine olan çok önemli bir özelliktir. Petri ağları için geliştirilen analiz teknikleri genellikle bazı özellikleri ispatlamak (güvenlik , değişmezlik , kilitlenme gibi) ve performans ölçütlerini hesaplamak için (cevap zamanı , bekleme süreleri , alıkoyma oranları gibi) kullanılır. Bu şekilde alternatif iş akışlarını değerlendirmek mümkün olur. İş akışı grafi yapısını kurduğumuzda istediğimiz bir özelliğin tutulup tutulmadığını doğrulama imkanımız olur. Örneğin kilitlenme durumunu ya da istenmeyen durumları tespit edebiliriz. Petri ağının temelini iyi oluşturmak için bu teknikleri kullanabiliriz. Ayrıca iş akışı prosedürünün karşılaması gereken aşağıdaki ihtiyaçları tespit edebiliriz [5,6] :

- İş akışındaki durumlara katkıda bulunmayan hiçbir görevin olmaması.
- Her hangi bir durum için prosedürün er geç sona ermesi.
- Özel bir durum için prosedür sona erdiğinde , bu duruma referans olan koşulların ortadan kalkması

Açıkçası Petri ağları için geliştirilen birçok analiz tekniği , Petri ağı tabanlı bir iş akışı yönetim sistemindeki iş akışı işlemini çok çeşitli yollarla analiz etme imkanı verir. Pratikte elde edilen tecrübelerle de Petri ağı yapısının bir çok özelliğinin iş akışı yönetiminde çok kullanışlı olduğu tespit edilmiştir[6].

5.3 İş Akışı Kavramlarının Petri Ağları ile Belirtimi

Bir iş akışının belirtimi , bilgisayar bilimlerinde yoğun olarak kullanılan Petri ağları ile yapılabilir. İş akışlarının Petri ağları ile tanımlanması aşağıdaki avantajları sağlar [7] :

1. Petri ağları ayrık dinamik modelleri temsil etmeye çok uygundur. Bir iş akışı da ayrık bir modeldir çünkü sürekli zaman ekseniiyle karşılaştırıldığında zaman içinde aktivitelerin yürütüldüğü noktalar ayrık bir küme oluşturur.

2. Petri ağı sağlam matematiksel tanımlara sahiptir. Bu tanımların sentaksı ve anlamı kesin olarak bellidir.

3. Eğer bir iş akışı Petri ağı ile ifade edilirse belirgin özellikleri, örneğin son işaretlemenin (ulaşmak istenen nokta) ulaşılabilirliği matematiksel olarak ispatlanabilir. Petri ağının gerçekleşmesiyle ilgili çok sayıda analiz metodu vardır.

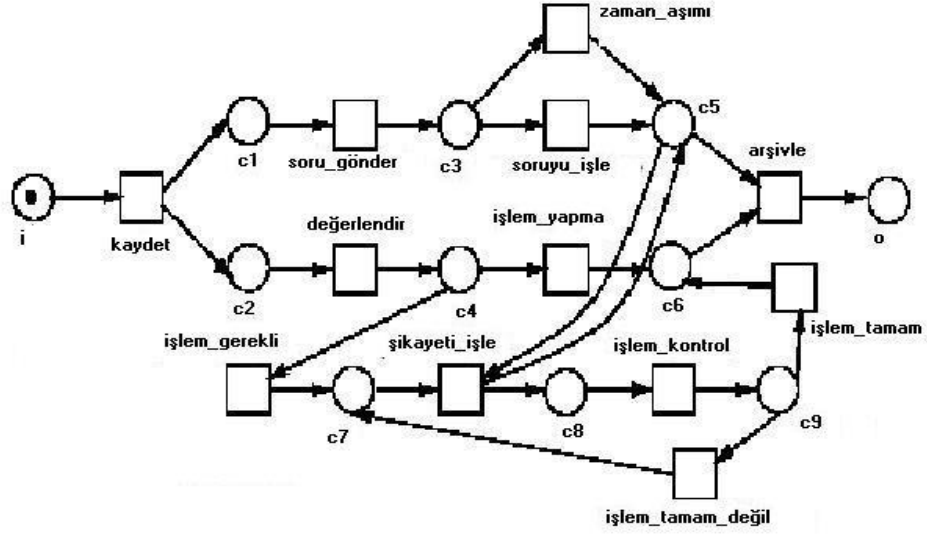
Bu sebeplerden dolayı iş akışı belirtiminde Petri ağları kullanılmaktadır.

Petri ağları ile belirtilen bir iş akışına Petri ağı iş akışı, ağ yapısına da iş akışı ağı (workflow net) denir. Petri ağı iş akışı aşağıdaki karakteristiklere sahiptir [7] :

- İş akışındaki aktiviteler Petri ağındaki geçişlere karşı gelir. Bir aktivitenin yürütülmesi , karşı gelen geçişin canlanması demektir.
- Petri ağının işaretli hali , iş akışının o anki durumunu temsil eder. Jetonlar iş akışının durumunu hangi aktivitelerin aktif hale geleceğini belirtirler . Akış ilişkisi jetonların ağ içinde nasıl hareket edeceğini söyler. Bir iş akışının kontrol akışı , akış ilişkisi ve başlangıç işaretlemesi ile belirlenir.

Petri ağları iş akışı kavramlarını belirtmeye ve modellemeye çok uygundur. İş akışı sürecinde yürütülmesi gerekli görevler ve bunların hangi sırada olacakları belirtilir. Bu iş akışı işlemleri Petri ağı terimleri ile açıkça modellenebilir. Görevler geçişlerle , şartlar yerler ile ve durumlarda jetonlarla modellenir [7].

İş akışı kavramlarını Petri ağları ile ifade etmek için örnek olarak bir şirkete gelen şikayetlerin değerlendirilmesi iş akışını göz önüne alalım [8].(Şekil 5.4)



Şekil 5.4 Şikayet değerlendirilmesi iş akışı ağı

İlk olarak şikayet kaydedilir (görev kaydet). Daha sonra paralel olarak şikayet eden kişiye soru gönderilir (görev soru_gönder) ve şikayet değerlendirilir (görev değerlendir). Eğer şikayet eden kişi iki hafta içinde cevap gönderirse soruyu_işle görevi yürütülür. Eğer iki hafta içinde cevap dönmezse (görev zaman_aşımı) sorular değerlendirilmez. Değerlendirme sonucuna göre şikayet işlenir veya işlenmez. Şikayetin işlenebilmesi için ayrıca (görev şikayeti_işle) soruyu_işle ya da zaman_aşımı görevlerinden birinin tamamlanması gerekir. İşlem kontrol görevi ile şikayet işlenmesi kontrol edilir. Son olarak arşivle görevi yürütülür.

Kaydet , soru_gönder , değerlendir , soruyu_işle , zaman_aşımı , şikayeti_işle , işlem_kontrol ve arşivle görevleri geçişlerle modellenmiştir. İşlem_tamam , işlem_tamam_değil , işlem_gerekli , işlem_yapma geçişleri iki olası cevabın çıktığı olayları modellemek için kullanılmıştır. Görevler arasındaki durumları modellemek içinde yerler kullanılmıştır.

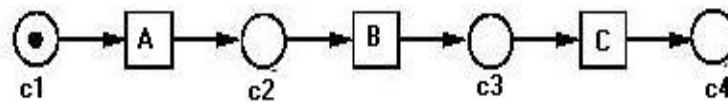
Örneğin c2 yeri değerlendirme yapılmaya hazır durumuna karşı gelir. (i) durumu başlangıç , (o) durumu da bitiş durumudur. Şekil 5.4’de verilen iş akışı işlem tanımı tek bir şikayetin başlangıç ve bitişini gösterir.

Tanım : İş akışı tanımını modelleyen bir Petri ağına iş akışı ağı (Workflow net) denir. İş akışı ağı iki koşulu gerektirir [8] :

1 - Bir iş akışının bir giriş yeri (i) ve bir çıkış yeri (o) bulunur. (i) 'deki bir jeton iş akışının başlayacağını belirtir. (o)'da ki bir jeton iş akışının bittiğini belirtir.

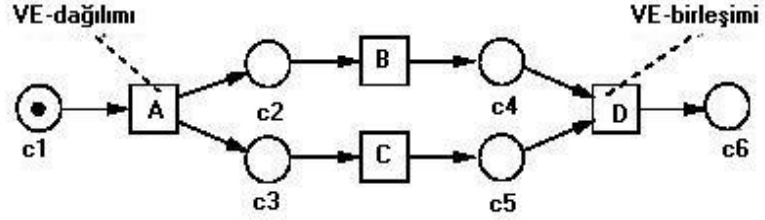
2 - Bir iş akışı ağında boşta ve işe yaramayan görev ve durumlar yoktur. Her görev (geçiş) ve durum (yer) iş akışında kullanılmalıdır. Bu yüzden her geçiş ve yer (i)'den (o)'ya giden yolda bulunmak zorundadır.

İş akışının modellenmesinde açık bir şekilde durumları temsil etmek çok önemlidir. Petri ağıda iş akışındaki tüm durumları açık bir şekilde temsil edebilir. Örneğin bir görev , bir görevin mümkün olması ve bir görevin yürütülmesi ayrı ayrı olaylardır. Bu karışıklığı önlemek için iki terim daha kullanılır : iş parçası ve aktivite. Bir iş parçası belli bir durum için artık gerçekleşmesi mümkün olan görevdir. Yani yürütülmek için hazırdır. Aktivite ise iş parçasının yürütülmesidir. Artık görev belli bir durum için yürütülür. Bu kavramları ve farklılıkları Petri ağı ile ifade etmek çok kolaydır. Bir görev bir geçişe karşı gelir . Bir iş parçası gerçekleşmesi mümkün olan bir geçişe (giriş yerleri jetonla dolu olan geçiş) karşı gelir. Aktivite ise geçişin canlanmasına karşı gelir. Örneğin şekil 5.4'de c5 ve c7 yerleri jeton ile dolu iken şikayeti_işle geçişi mümkün olur ve bu artık bir iş parçasıdır. C5 ve c7 'deki jetonlar tüketilerek geçiş sağlandığında bu geçiş bir aktivite olur. Şekil 5.1'de gösterilen yapılar ile ardışıl şarta bağlı , paralel ve döngülü yollar iş akışı ağında kullanılabilir[8]. Ardışıl yol birbiri ile nedensel ilişkide olan görevleri belirtmek için kullanılır. Şekil 5.5 'de görüldüğü gibi A görevi tamamlandıktan sonra B yapılabilir. C2 yeri A 'dan sonra oluşan durumu ve B'den önce olması gereken durumu ifade eder.



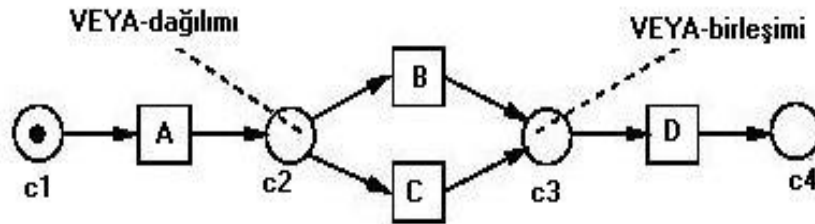
Şekil 5.5 İş akışı ağında ardışıl yol

Paralel yollar çalışma sırası önemli olmayan görevler için kullanılır. Örneğin şekil 5.6'da B ve C'nin yürütülmesi gerekir fakat yürütülme sırası önemli değildir. Paralel yol için iki blok kullanılır : VE-dağılımı ve VE-birleşimi [8].



Şekil 5.6 İş akışı ağında paralel yol

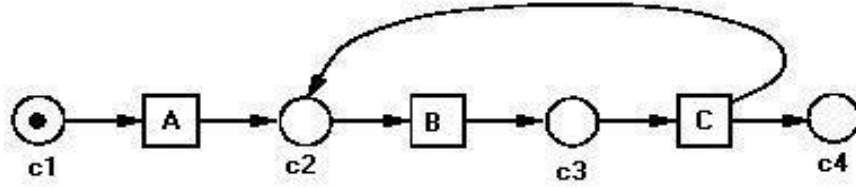
Şarta bağlı yollar ise o anki koşullara göre değişebilen görevler için kullanılır. İki alternatiften biri seçilir. Bunun için iki blok kullanılır : VEYA-dağılımı ve VEYA-birleşimi. Şekil 5.7'de görüldüğü gibi A görevi ya B ile ya da C ile devam edebilir. B ve C arasında bir seçim yapılır. Bu görevlerden biri yapıldığında D görevi için C3 durumu sağlanır.



Şekil 5.7 İş akışı ağında şarta bağlı yol

İterasyonlu yollar bir görevin sonucunda önceden yürütülmüş başka bir görevin tekrar yürütülmesinin gerektiği durumlarda kullanılır[8].

Şekil 5.8’de görüldüğü gibi C görevi B görevinin sonucunu kontrol eden bir görevdir. Bu kontrole göre B görevinin yeniden yürütülmesi gerekebilir. B bir ya da daha fazla yürütülebilir. İterasyon genelde iş akışı ağında bulunması istenmeyen bir yol çeşididir.



Şekil 5.8 İş akışı ağında iterasyonlu yol

5.4 Petri Ağı ile Modellenen İş Akışlarının Analizleri

İş akışlarını modelleyen Petri ağlarının iş akışlarına özel olarak çeşitli analiz teknikleri vardır. İş akışı işlem tanımı eğer hatalar içerirse bu müşteri tepkisine , güvensizliğe , uzun bekleme sürelerine , yavaş servise ve çeşitli kapasitelerin aşılmasına neden olabilir. Bu nedenle iş akışında , doğruluk, etkinlik ve verimlilik çok önemlidir. İş akışı hayata geçirilmeden önce analiz edilmelidir.

İş akışlarının temel olarak 3 tip analizi yapılır[8] :

- 1 - Geçerlilik : İş akışının planlandığı gibi davranıp davranmadığı test edilir.
- 2 - Gerçeklenme : İş akışının doğru bir şekilde kurulup kurulmadığı incelenir.
- 3 - Performans analizi :Uygun zamanda ihtiyaçların karşılanması, kaynakların kullanım süreleri gibi değerlendirmeler yapılır.

İş akışlarına özel analiz tekniklerini sırasıyla inceleyelim.

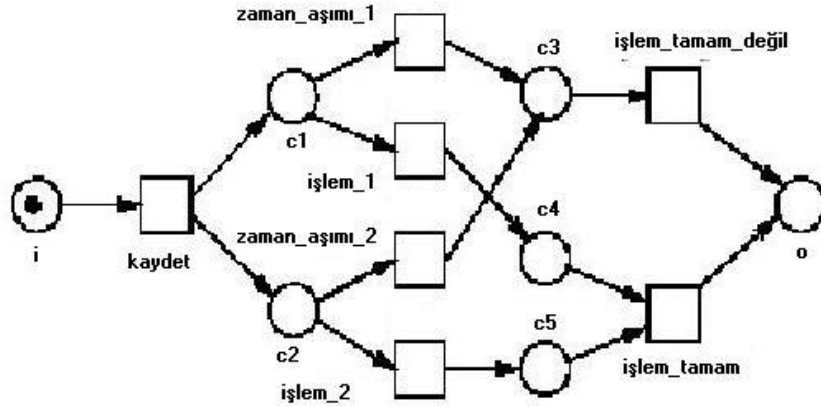
5.4.1 Sağlamlık (soundness) özelliği

Tanım : Petri ağı ile modellenen bir iş akışının sağlam olabilmesi için aşağıdaki koşulların sağlanması gerekir[8].

1 – Başlangıç yerinden (i) ulaşılabilen tüm durumlar için bitiş durumu (o) ‘ya varışı sağlayacak geçişler olmalıdır.

2 – Bitiş durumu (o), başlangıç durumu (i) ‘den ulaşılabilen en son yer olmalıdır. İş akışı sona erdiğinde (i) yerinde sadece bir jeton bulunmalıdır.

3 – İş akışı ağında ölü geçişler (hiç kullanılmayan) bulunmamalıdır.



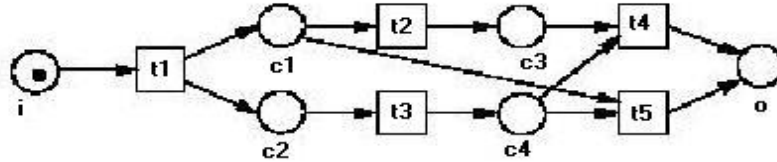
Şekil 5.9 Sağlam olmayan iş akışı ağı

Şekil 5.9 'yi göz önüne alırsak bu iş akışı ağı sağlam değildir. Çünkü zaman_aşımı_1 ve işlem_2 geçişleri olursa ya da zaman_aşımı_2 ve işlem_1 geçişleri olursa (C4 ve C5 'te 1'er jeton olmayacağından) iş akışı sonlanamaz. Bunun sağlanmadığı durumlarda işlem_tamam geçişi gerçekleşmez. Eğer zaman_aşımı_1 ve zaman_aşımı_2 geçişleri olursa işlem_tamam_değil görevi 2 defa yürütülür ve (o) yerine 2 jeton gelir. Bu sonlanma ise bir iş akışı ağı için doğru değildir çünkü iş akışı tek bir örnek için bittiğinde en son bitiş yerinde bir jeton olmalıdır. Şekil 5.4 sağlam bir iş akışı ağıdır.

Sağlamlılık özelliğinin yapısal karakteri incelendiğinde 2 alt sınıf iş akışı daha ortaya çıkmaktadır. Bunlar serbest seçimli iş akışı ağları ve iyi yapıli iş akışı ağlarıdır.

5.4.1.1 Serbest Seçimli İş Akışı Ağları

Bir çok iş akışı yönetim sistemlerinde görevler arasındaki tüm durumlar açık olarak temsil edilememektedir. Bu iş akışı yönetim sistemleri yapı bloklarından VE-dağılımı , VE-birleşimi , VEYA-dağılımı ve VEYA-birleşimi kullanarak iş akışı tanımlarını belirtirler. VE-dağılımı ve VE-birleşimi iş akışı ağında paralel yollar için, VEYA-dağılımı ve VEYA-birleşimi iş akışı ağında şartlı yollar için kullanılır. Eğer Petri ağı terimleriyle iş akışında VEYA-dağılımı modellenirse bu aynı yeri giriş olarak paylaşan geçişlere karşı gelir[8]. Bu yapıyı kullanan iş akışına da serbest seçimli iş akışı denir. Örneğin Şekil 5.7 bir serbest seçimli iş ağıdır. Şekil 5.4 ise serbest seçimli değildir çünkü arşivle ve şikayeti_ işle görevleri aynı yeri giriş olarak paylaşmaktadır fakat bu geçişlerin diğer girişleri birbirinden bağımsız ve farklıdır. Açıkçası paralel , ardışıl , şartlı ve iterasyonlu yollar serbest seçim özelliğini bozmadan modellenebilir. İş akışlarının serbest seçimli Petri ağı olarak modellenmesi görevler arasındaki karışıklığı önler. Örneğin Şekil 5.10'da paralellik ve seçim karışmıştır. C1 ve C2 yerleri dolu iken t5 geçişi gerçekleşemez çünkü C4 boştur. Bu yüzden t2 ve t5 geçişleri arasında gerçek bir seçim yapılamaz. T3 geçişi sağlandıktan sonra ancak t2 ve t5 geçişleri arasında gerçek bir seçim yapılabilir. Bu yüzden Şekil 5.10'daki iş akışı ağı serbest seçimli değildir.

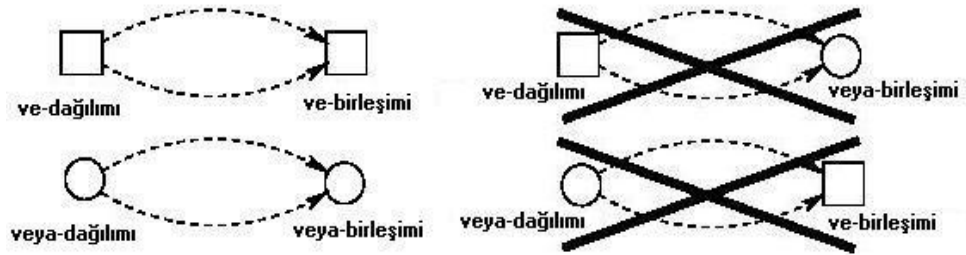


Şekil 5.10 Serbest seçimli olmayan iş akışı ağı

Serbest seçimli iş akışı ağları sağlam olmaya yatkındırlar ve kolay anlaşılırlar. Serbest seçimli olmayan iş akışı ağları ise potansiyel olarak hatalı olabilirler. Yürütülme sırasında kilitlenme görülebilir ve takip edilmeleri zordur[9,10].

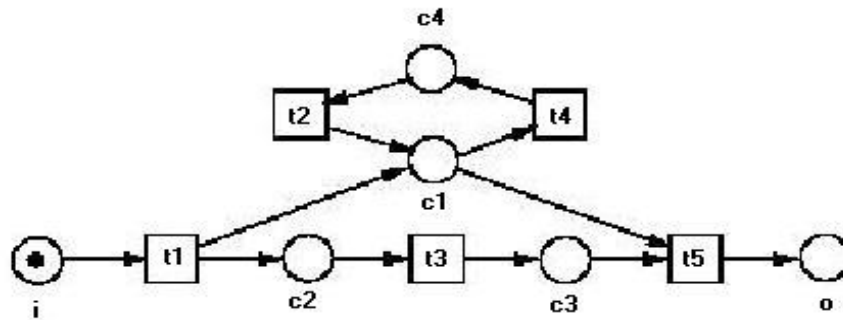
5.4.1.2 İyi Yapılı İş Akışı Ağları

İş akışı ağlarının yapılarında VE/VEYA-dağılımları ve VE/VEYA-birleşimi dengeli olmalıdır. Yani VE-dağılımı ile başlayan iki paralel akış başlangıcı VEYA-birleşimi ile birleşmemelidir. VEYA-dağılımı ile başlayan iki alternatif akış da VE-birleşimi ile birleşmemelidir. Şekil 5.11’de gösterildiği gibi VE-dağılımı VE-birleşimi ile bitmeli, VEYA-dağılımı da VEYA-birleşimi ile bitmelidir. Bu koşulları sağlayan iş akışı ağına iyi yapı (well-structured) iş akışı ağı denir[8].



Şekil 5.11 İş akışı ağında iyi ve kötü yapılar

Şekil 5.4 ‘de görülen iş akışı ağı iyi yapı değildir. Çünkü VE-dağılımı kaydet, VEYA-birleşimi C5 yerinde tamamlanmaktadır. Şekil 5.12 ise iyi yapı bir iş akışı ağıdır. Fakat serbest seçimli değildir.



Şekil 5.12 İyi yapı iş akışı ağı

İyi yapılı ve serbest seçimli iş akışı ağları benzer özelliklere sahiptirler. İş akışı ağının bu iki özellikten birini sağlaması iş akışı ağının sağlam olduğunu gösterir. Fakat iyi yapılı iş akış serbest seçimli olmayabilir ya da serbest seçimli bir iş akışı da iyi yapılı olmayabilir. Ayrıca serbest seçimli ya da iyi yapılı olmayan bir iş akışı da sağlam olabilir[8]. (örnek Şekil 5.4)

Bu iki yapısal karakter iş akışı işlem tanımlarının analizinde çok kullanışlıdır. Bu özellikler iş akışı ağlarında sağlanması istenen özelliklerdir. Bunlardan en az birini sağlayan bir iş akışı verimli bir şekilde kullanılabilir. Serbest seçimli ya da iyi yapılı iş akışı ağını bozan yapıların gerekli olup olmadığı araştırılmalıdır. Çünkü serbest seçimli ya da iyi yapılı olmayan iş akışı ağlarında hata bulunma olasılığı yüksektir[12].

6. EĞİTİM TALEBİ İŞ AKIŞININ MODELLENMESİ VE ANALİZİ

Günümüzde şirketlerde kullanılan en önemli iş akışlarından birisi personelin eğitim taleplerini karşılayan iş akışıdır. Bu iş akışı süreci şirketin büyüklüğüne göre ve personel sayısına göre değişir. Eğitim talebi iş akışı sayesinde personeller gerekli konularda eğitim ihtiyaçlarını karşılarlar. Personelin belli bir zamanda, bir konuda eğitim alarak hazır bulunması ya da eğitim aldığı konuda çalışması gerekebilir. İş süreçlerinde zaman çok önemli olduğundan eğitim talebi iş akışının da performansı iyi olmalıdır. Cevap alma süreleri , görevlerin bekleme süreleri , kapasite oranları , güvenlik ve kilitlenme gibi özellikler genel olarak iş akışı süreçlerinde incelenmesi, analiz edilmesi gereken konulardır.

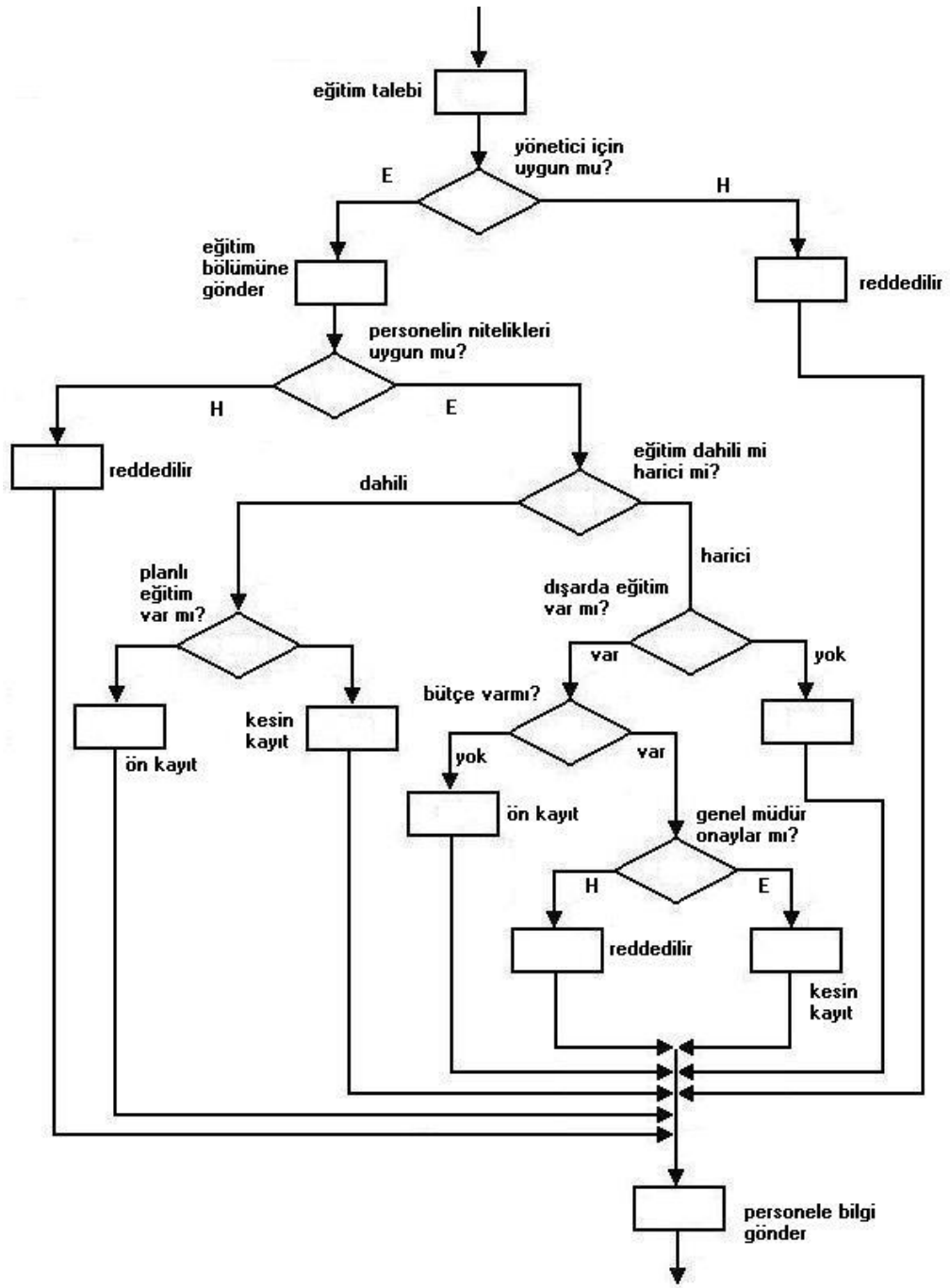
Bu çalışmada bir şirkette kullanılan süreçler elde edilerek eğitim talebi iş akışı Petri ağı ile modellenmiştir. Ayrıca bu model iş akışı ve Petri ağı analiz teknikleri ile analiz edilerek incelenmiştir.

6.1. Eğitim Talebi İş Akışının Petri Ağı ile Modellenmesi

Her iş akışında olduğu gibi eğitim talebi iş akışının da bir giriş (başlangıç noktası) ve bir çıkış (bitiş noktası) bulunmaktadır. Eğitim talebi iş akışı bir personelin eğitim talebinde bulunması ile başlar. Personele değerlendirme sonuç bilgisi gönderilmesi ile biter. (Talebiniz reddedildi, eğitime ön kaydınız yapıldı, eğitime kesin kaydınız yapıldı gibi). Personelin eğitim talebi bulunduğu bölümün yöneticisi tarafından değerlendirilir. Bölüm yöneticisi bu talebi uygun bulmayarak reddeder veya onaylar. Eğer reddederse kişiye nedeniyle birlikte reddedilme bilgisi gönderilir. Eğer bölüm yöneticisi bu talebi onaylarsa bu talebi eğitim bölümüne bildirir. Eğitim bölümü personelin istediği eğitimi almaya niteliklerinin uygun olup olmadığını değerlendirir. Örneğin talep edilen eğitimden önce alınması gereken başka eğitimler olabilir ya da personelin İngilizce biliyor olması gerekebilir. Eğitim bölümü bunları değerlendirilerek personelin nitelikleri uygun değilse nedeniyle birlikte reddedildiniz bilgisi gönderir. Eğer uygunsa eğitim bölümü alınacak eğitimin şirket içimi yoksa dışımı olduğuna karar verir. Şirket içinde verilebilecek bir eğitimse planlı bir

eğitimin olup olmadığına bakar. Eğer planlı bir eğitim varsa personelin kesin kaydını yapar ve eğitim planıyla birlikte kesin kaydınız yapıldı bilgisini gönderir. Eğer şirket içinde planlı bir eğitim yoksa planlanınca gönderilmek üzere personelin ön kaydı yapılır. Eğer eğitim şirket dışında bir eğitim ise dışarıda alınabilecek eğitimin olup olmadığı araştırılır. Eğitim bulunamazsa, gönderilmek üzere personelin ön kaydı yapılır ve bilgi gönderilir. Dışarıda uygun eğitim bulunursa eğitim bölümü bu sefer bütçe değerlendirmesi yapar. Eğer eğitim masraflarını karşılayacak bütçe yoksa gönderilmek üzere personelin ön kaydı yapılır ve bilgi gönderilir. Bütçe varsa geriye sadece genel müdürün onayı kalır. Genel müdür onaylarsa dışarıdaki bir eğitime personelin kesin kaydı yapılır. Eğer onaylamazsa reddedilme bilgisi personele gönderilir.

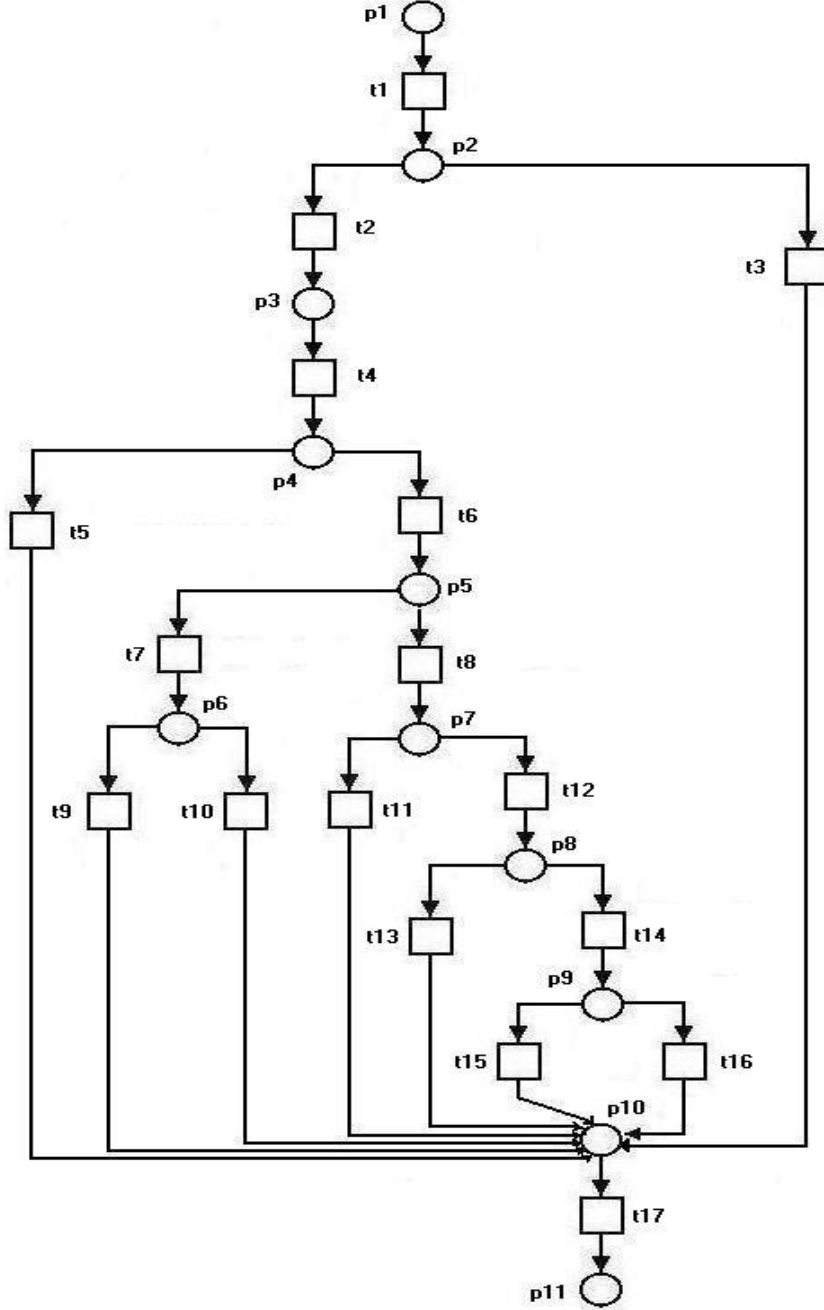
Görüldüğü gibi iş akışı olumlu ya da olumsuz kişiye bilgi gönderilmesi ile sona ermektedir. Bu bilgiler şunlar olabilir : Şu tarihlerde ve şu yerdeki eğitime kesin kaydınız yapıldı , şirket içindeki şu eğitime şu tarihler için kesin kaydınız yapıldı, ilerde gönderilmek üzere şu eğitime ön kaydınız yapıldı, genel müdür tarafından eğitim talebiniz reddedildi gibi. Süreçleri belirtilen eğitim talebinin akış diyagramını çıkartırsak şekil 6.1 elde edilir.



Şekil 6.1 Eğitim talebi iş akışı akış diyagramı

Daha önce Petri ağı ile modellemeye örnekler verdiğimiz 3.3 Program modellemesi bölümünde bir akış diyagramının Petri ağı ile nasıl ifade edileceğini göstermiştik. O

bölümdeki şekil 3.6’da verilen dönüşüm kuralları ile eğitim talebi akış diyagramını Petri ağına dönüştürürsek şekil 6.2 elde edilir.



Şekil 6.2 Eğitim talebi iş akışının Petri ağı ile modeli

Elde ettiğimiz bu Petri ağına bir iş akışı ağı diyebiliriz. Çünkü iş akışı olma koşullarını yerine getirmektedir. Eğitim talebi iş akışının bir giriş yeri (p1) ve bir

çıkış yeri (p11) vardır. Bu iş akışı ağında boşta ve işe yaramayan görev ve durumlar bulunmamaktadır. Her yer ve geçişler iş akışında kullanılmaktadır. Bütün yer ve geçişler p1'den p11'e giden yollar üzerinde bulunmaktadır. Şekil 6.2 'deki Petri ağında 11 yer (durum) ve 17 geçiş (görev) vardır. Bunların anlamları aşağıda verilmiştir.

Yerler (durumlar) :

P1 : Başlangıç durumudur. Eğitim talebinden önceki durumdur.

P2 : Eğitimin talep edildiği ve yöneticinin olumlu ya da olumsuz karar vereceği durumdur.

P3 : Yöneticinin uygun görüp talebi eğitim bölümüne gönderdiği durumdur.

P4 : Eğitim bölümünün artık değerlendirme yapacağı durumdur. Personelin nitelikleri değerlendirilme aşamasındadır.

P5 : Personelin niteliklerinin uygun bulunduğu durumdur.

P6 : Şirket içinde planlı eğitimin olup olmadığına bakılacak durumdur.

P7 : Şirket dışında uygun eğitimin olup olmadığına bakılacak durumdur.

P8 : Bütçenin kontrol edilme aşamasına gelinecek durumdur.

P9 : Genel müdürün değerlendirme yapacağı durumdur.

P10 : Personele bilgi gönderilecek durumdur.

P11 : Bitiş durumudur. Personele bilgi gönderilmiştir.

Geçişler (görevler) :

T1 : Personel eğitim talebinde bulunur.

T2 : Yönetici personelin eğitim talebini eğitim bölümüne gönderir.

T3 : Yönetici personelin eğitim talebini reddeder.

T4 : Personelin nitelikleri değerlendirilir.

T5 : Eğitim bölümü talebi reddeder.

T6 : Eğitim bölümü talebi onaylar.

T7 : Şirket içi planlı eğitimin olup olmadığı araştırılır.

- T8 : Şirket dışında uygun eğitimin olup olmadığı araştırılır.
- T9 : Şirket içi planlı eğitim yoktur. Ön kayıt yapılır.
- T10: Şirket içi planlı eğitim vardır. Kesin kayıt yapılır.
- T11: Şirket dışı uygun eğitim yoktur. Ön kayıt yapılır.
- T12: Şirket dışı uygun eğitim vardır. Masraflar için bütçe değerlendirmesi yapılır.
- T13: Bütçe yoktur. Personelin daha sonra gönderilmek üzere ön kaydı yapılır.
- T14: Bütçe vardır. Talep genel müdürün onayına gönderilir.
- T15: Genel müdür onaylar. Personelin kesin kaydı yapılır.
- T16: Genel müdür onaylamaz. Personelin eğitim talebi reddedilir.
- T17: Personele sonuç bilgisi gönderilir.

Eğitim talebi iş akışı ağının durum denklemlerini çıkarmak istersek sistem matrisi A, (2.16) denklemi kullanılarak aşağıdaki gibi elde edilir. T_1 geçişi öncesindeki durum x_0 aşağıdaki gibi olsun.

$$x_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

(2.18)'da verilen $f(x, t_j) = x + u.A$ geçiş fonksiyonu ile x_1 durumunu hesaplayalım. T_1 geçişi olduğundan (2.15)'e göre

$$u = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \quad \text{olur.}$$

$$x_1 = x_0 + u.A \text{ işlemini yaptığımızda}$$

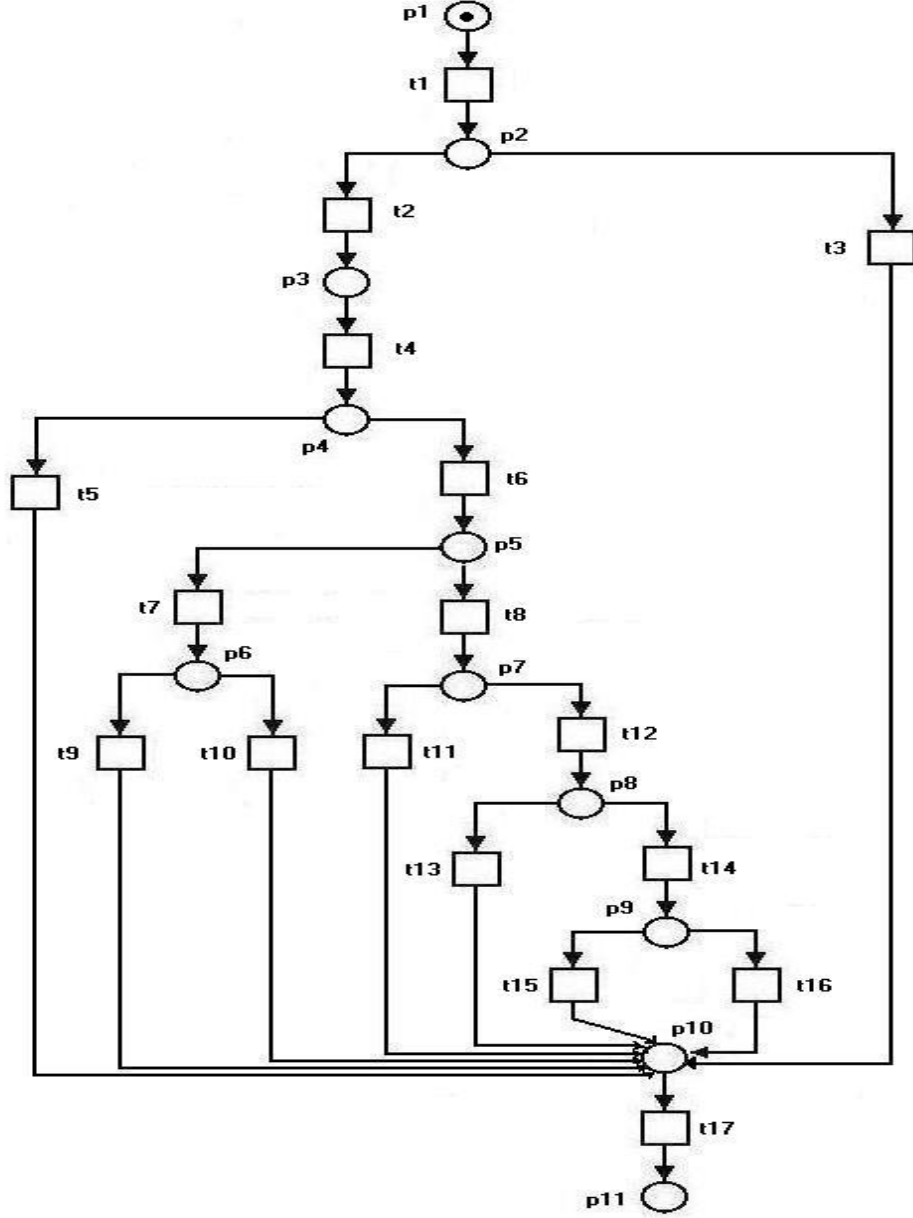
$$x_1 = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \quad \text{elde edilir.}$$

Diğer tüm durumlarda geçiş fonksiyonu kullanılarak sırasıyla elde edilebilir.

$$A = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \end{pmatrix}$$

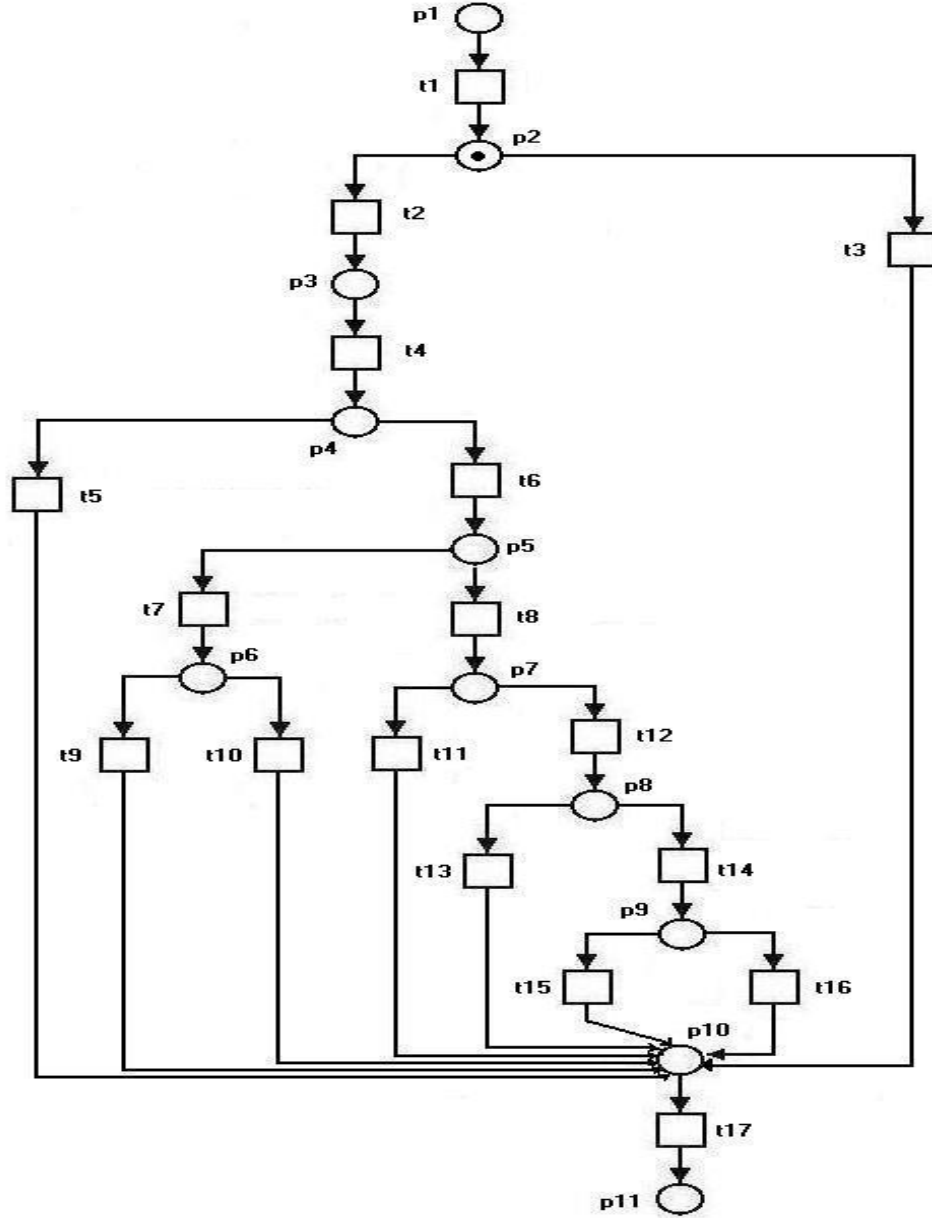
Bir personelin eğitim talebinin değerlendirilip, şirket dışında bir eğitime kesin kaydının yapıldığını varsayalım. Bunun gerçekleşme aşamalarını Petri ağı modeli üzerinde inceleyelim ve durumlarını belirtelim.

Şekil 6.3 iş akışının başlangıç durumudur. Başlangıç yerinde (p1) bir adet jeton bulunur. Durum vektörü $x_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$



Şekil 6.3 Eğitim talebi iş akışının başlangıç durumu

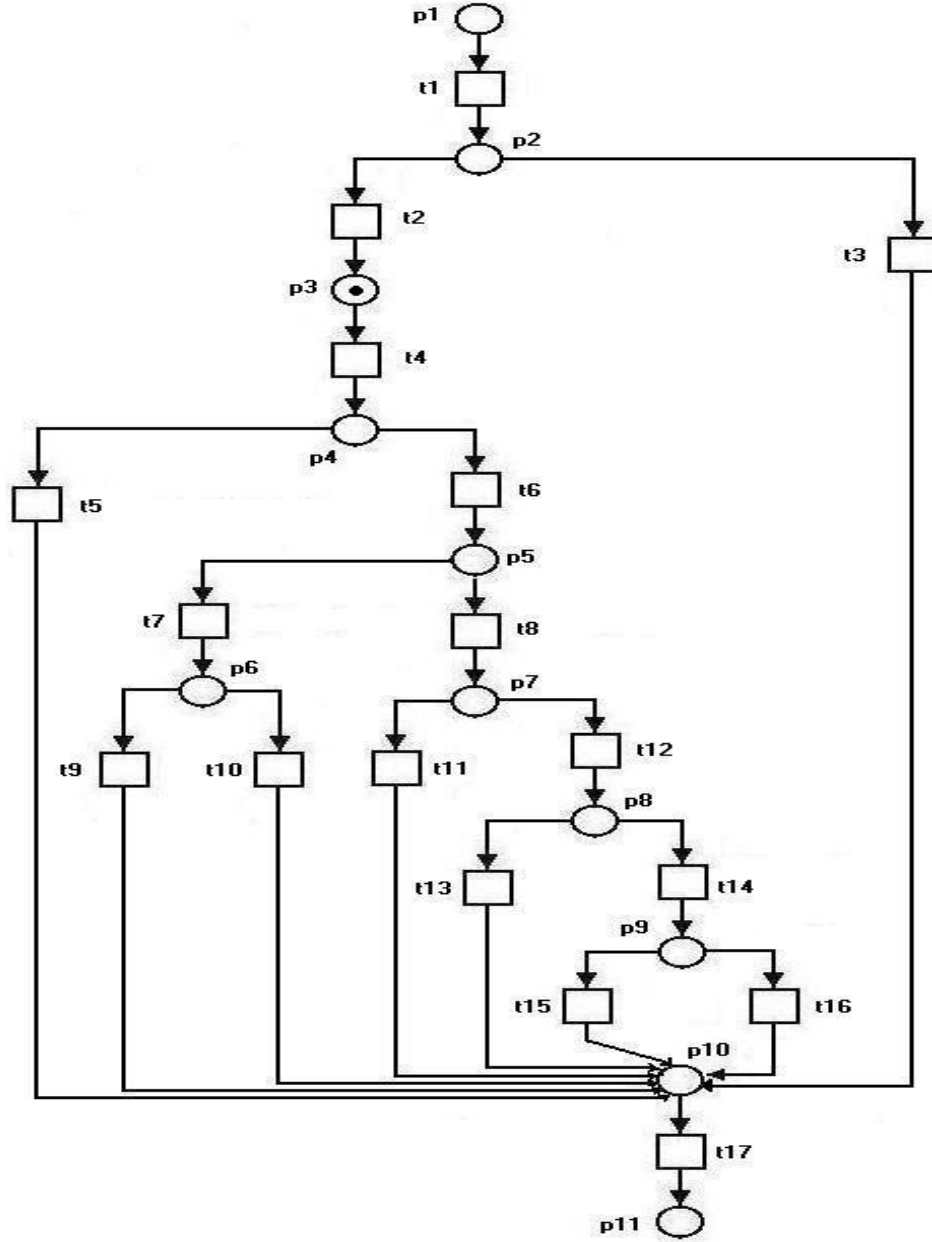
Personel eğitim talep eder ve t1 görevini yürütür. Böylece t1 geçişi canlanır ve jeton p2 yerine geçer. (Şekil 6.4) Durum vektörü $x_1 = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$



Şekil 6.4 Personelin eğitim talep ettiği durum

Yönetici eğitim talebini, eğitim bölümüne gönderir yani t2 görevi yürütülür. Bu şekilde t2 geçişi canlanır ve jeton p3 yerine geçer. (Şekil 6.5)

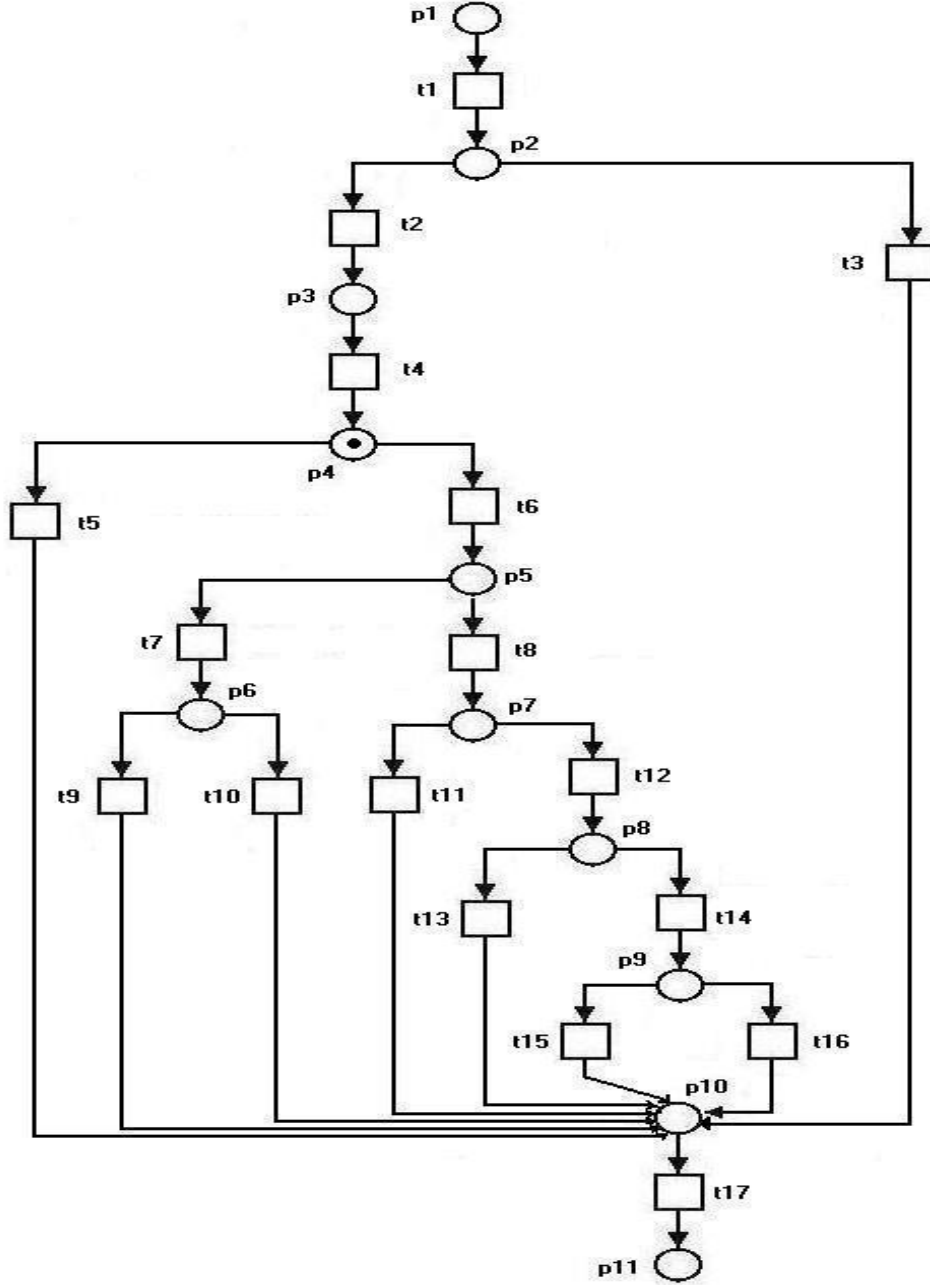
Durum vektörü $x_2 = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$



Şekil 6.5 Talebin eğitim bölümüne gönderildiği durum

Personelin niteliklerinin değerlendirildiği t4 görevi yürütülür. T4 geçişi sonrası jeton p4 yerine geçer. (Şekil 6.6)

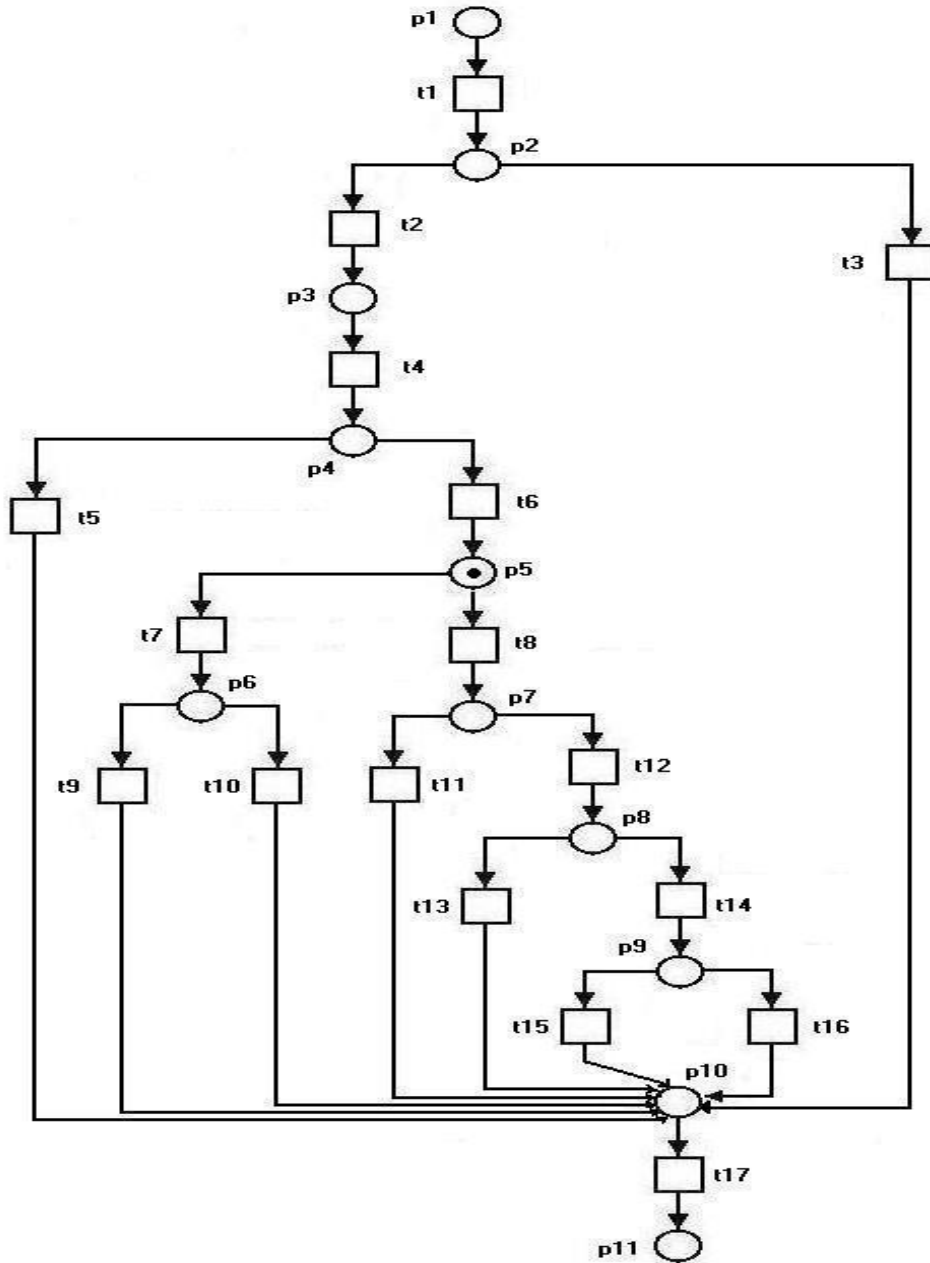
Durum vektörü $x_3 = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$



Şekil 6.6 Personelin niteliklerinin değerlendirildiği durum

Eğitim bölümü talebi onaylarsa t6 görevi yürütülür ve jeton p5 yerine geçer.

(Şekil 6.7) Durum vektörü $x_4 = [0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$

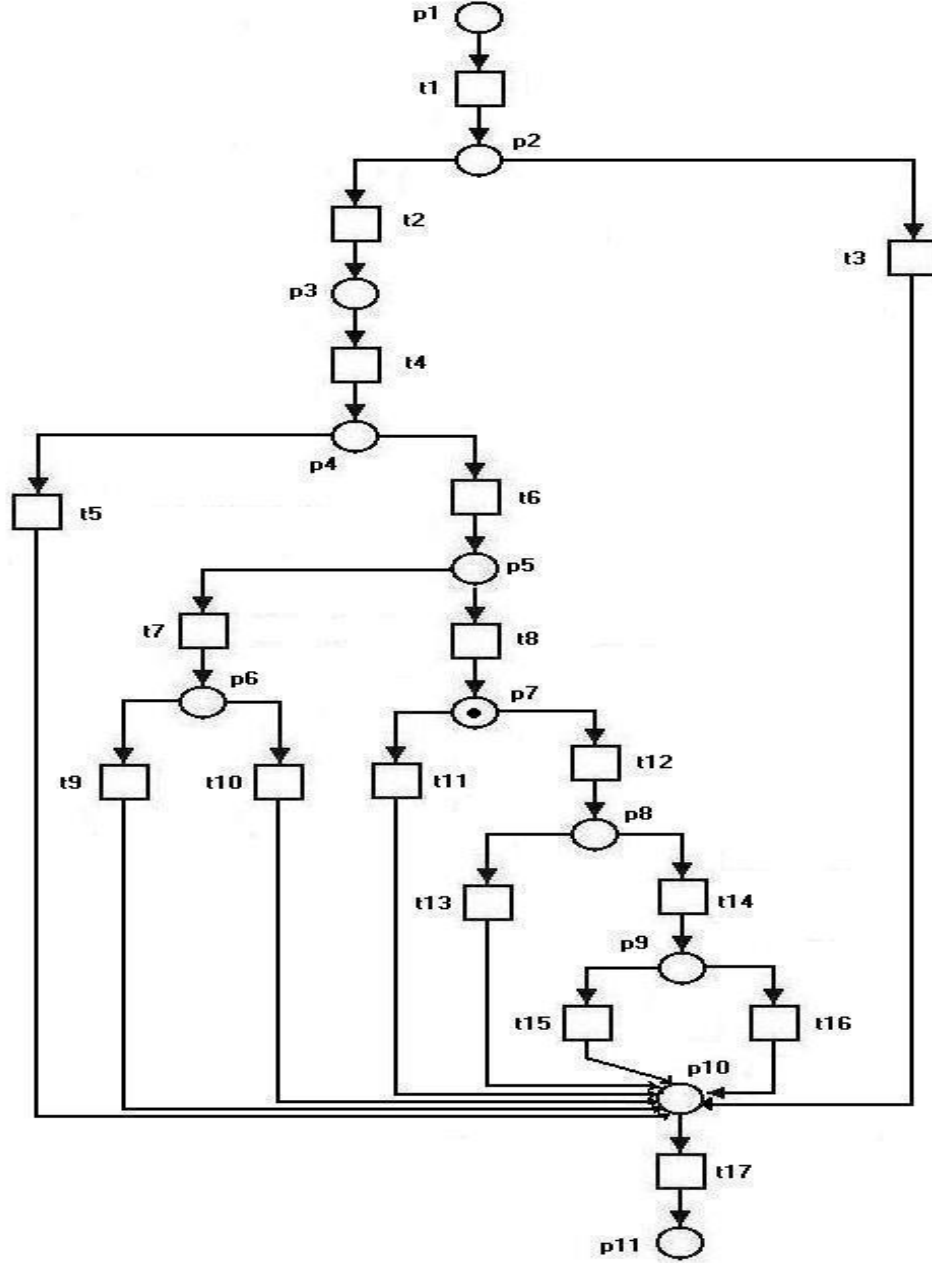


Şekil 6.7 Personelin niteliklerinin uygun bulunduğu durum

Eğitimin dahili mi , harici mi olduğuna bakılır. Harici ise t8 görevi yürütülür.

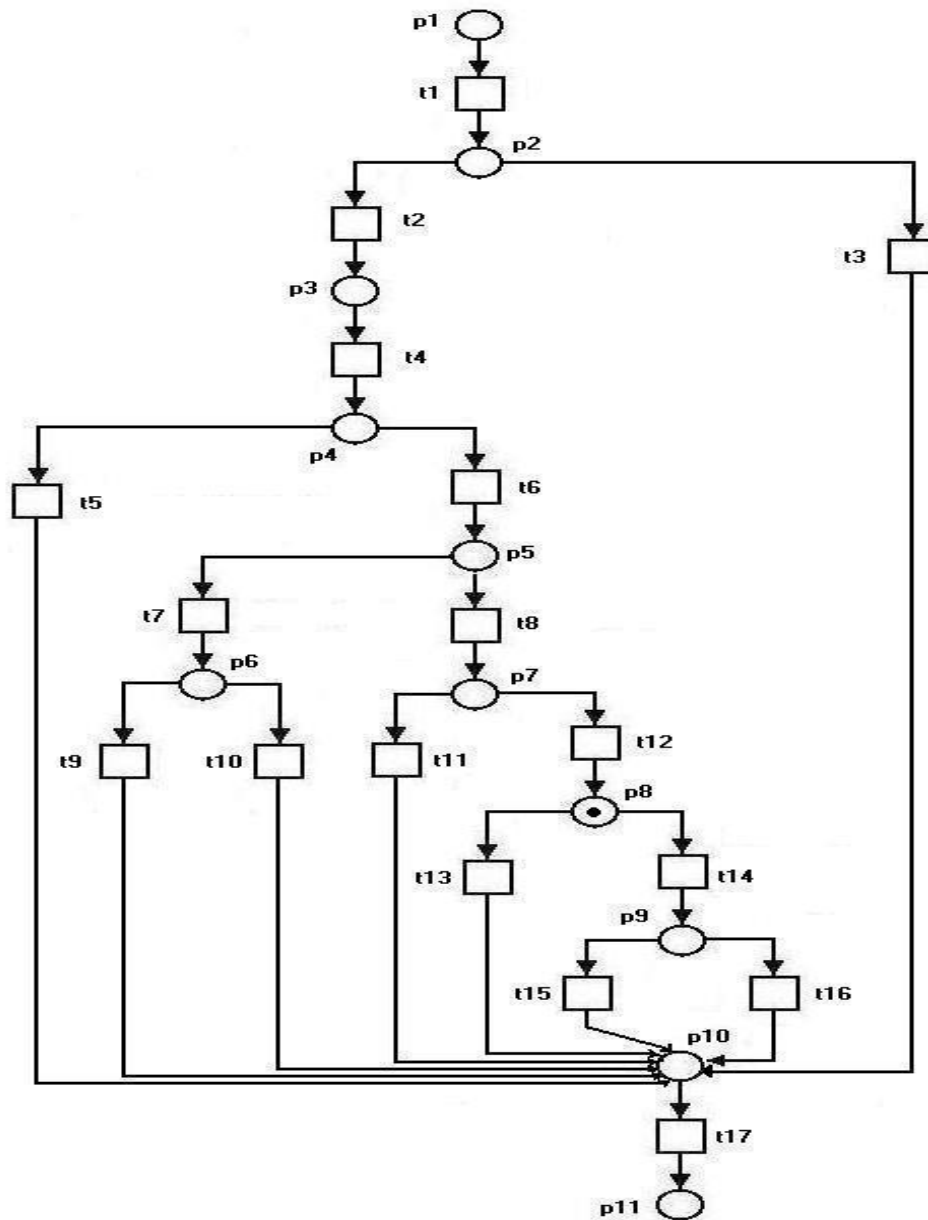
Dışarıda uygun eğitim olup olmadığına bakılır. Jeton da p7 yerine geçer.

Durum vektörü $x_5 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]$



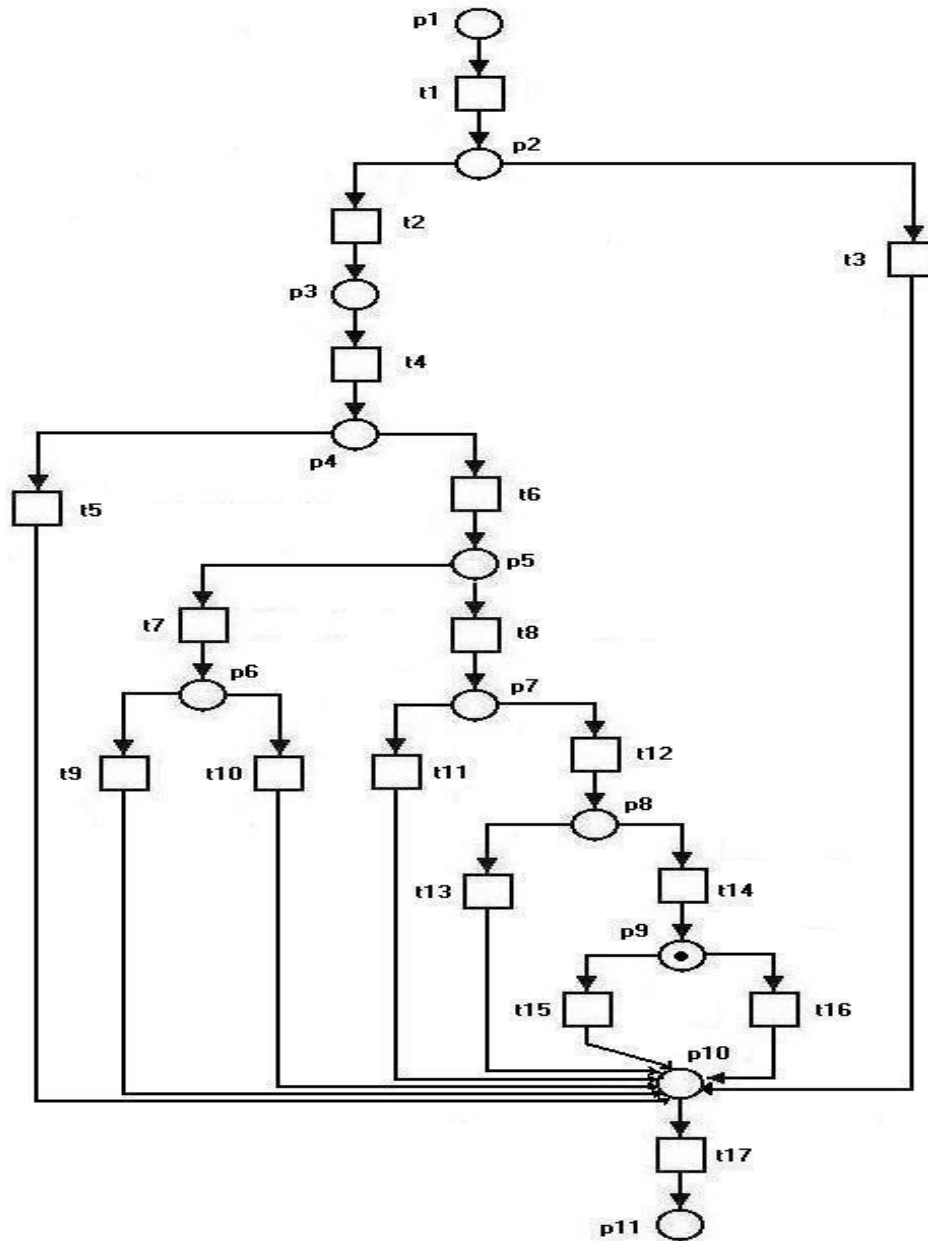
Şekil 6.8 Dışarıda uygun eğitimin olup olmadığına bakılacak durum

Durum vektörü $x_6 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]$



66

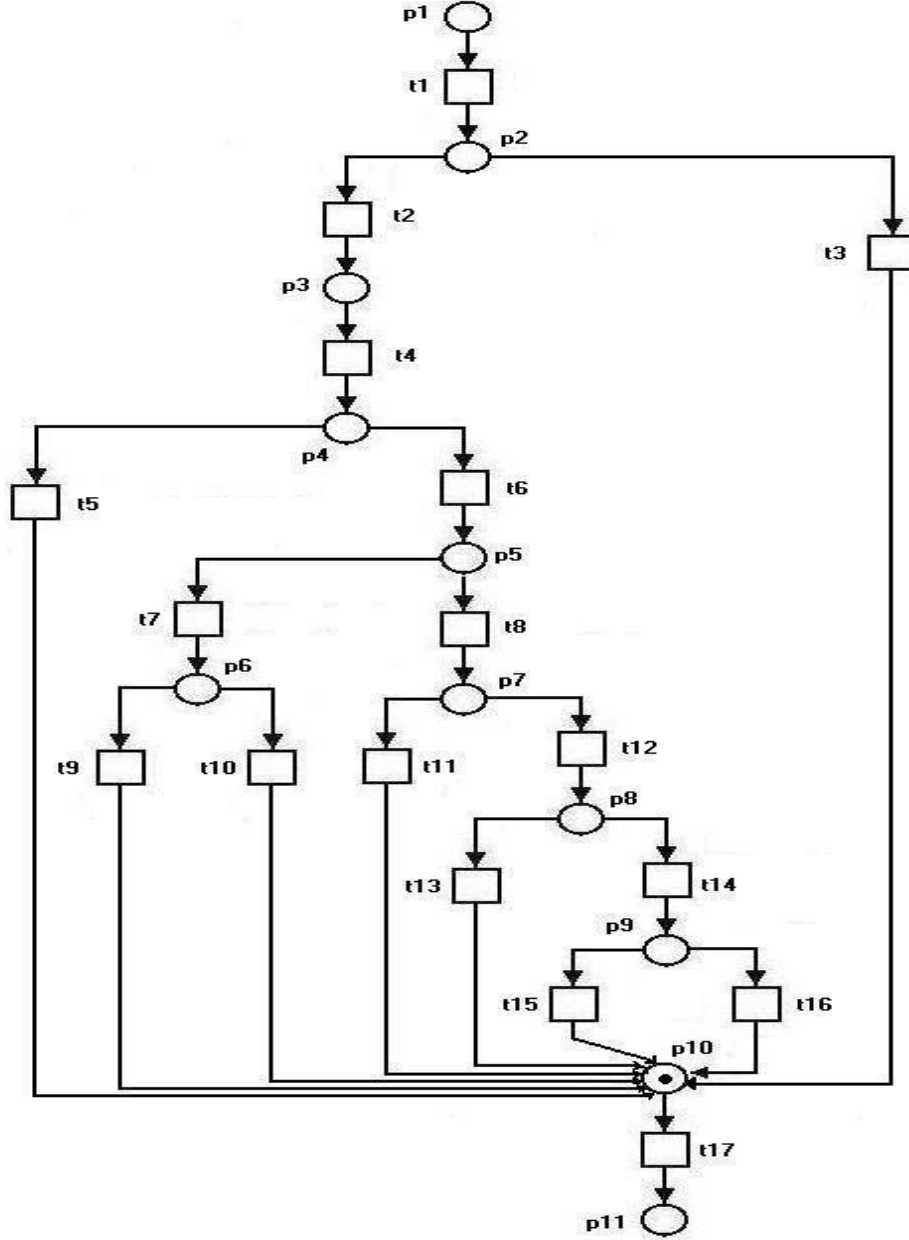
Durum vektörü $x_7 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0]$



67

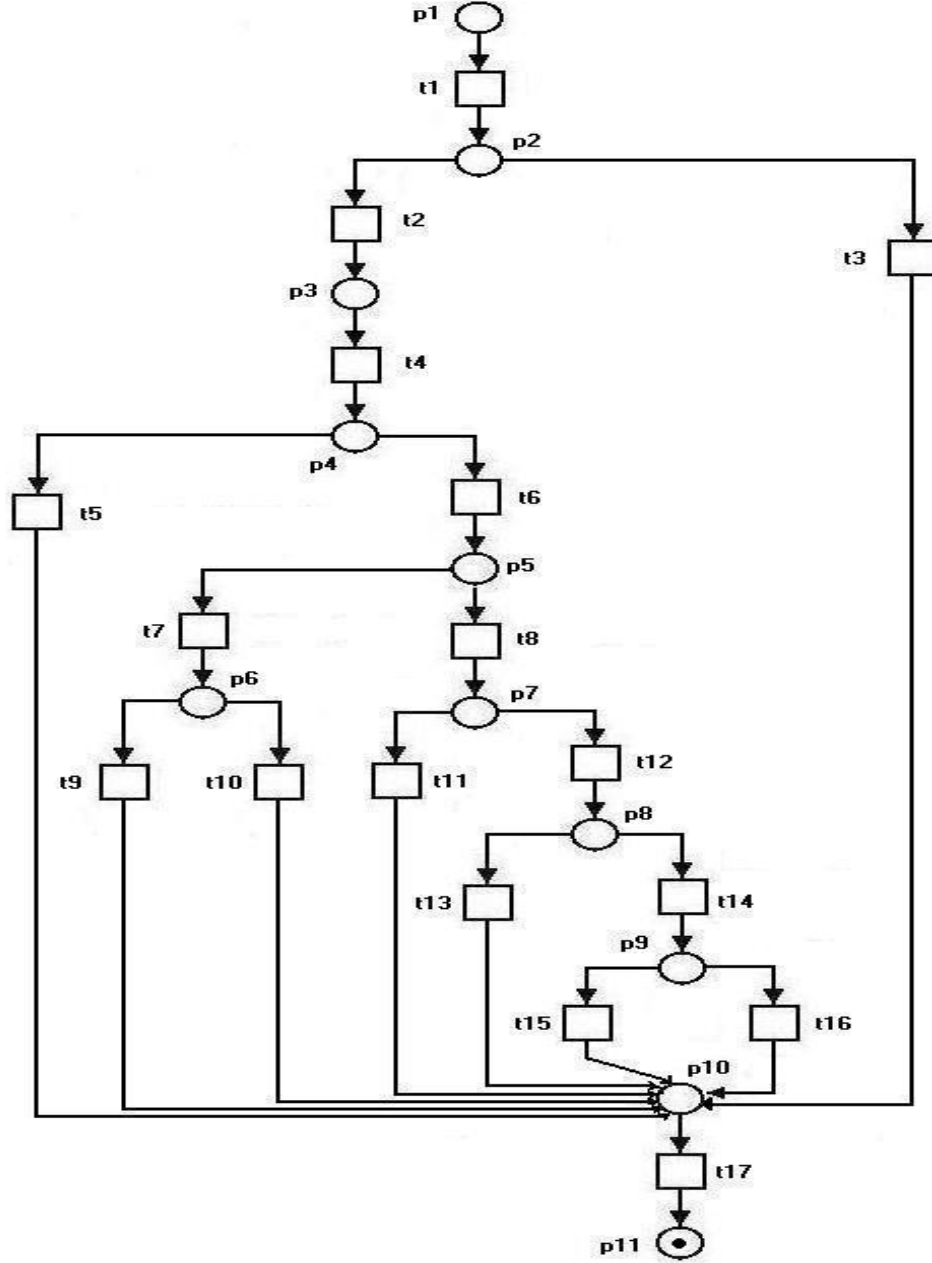
Genel müdür değerlendirme yapar. Eğer onaylarsa t15 görevi yürütülür ve jeton p10 yerine geçer. Personele bilgi gönderilmesi aşamasına gelinmiştir.

Durum vektörü $x_8 = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0]$



Şekil 6.11 Personele bilgi gönderilecek durum

T17 görevi ile personele bilgi gönderilir. T17 geçişi sağlanınca jeton p11'e geçer ve iş akışı tamamlanır. Durum vektörü $x_9 = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]$



Şekil 6.12 Bitiş durumu (personele bilgi gönderilmiştir)

6.2 Eğitim Talebi İş Akışı Ağının Analizi

İş akışı analiz tekniklerine göre eğitim talebi iş akışı ağını (şekil 6.2) analiz edersek ilk önce sağlam olduğunu görürüz. Bir iş akışının sağlam olabilme koşullarını bölüm 5.4.1 'de vermiştik. Eğitim talebi iş akışı ağında başlangıç yerinden (p1) ulaşılabilen tüm durumlar için bitiş durumu (p11) 'ya varışı sağlayacak geçişler vardır. Bitiş durumu (p11) , başlangıç durumu (p1) 'den ulaşılabilen en son yer yerdir. İş akışı sona erdiğinde (p11) yerinde her koşulda sadece bir jeton bulunur. Ayrıca eğitim talebi iş akışı ağında ölü geçişler (hiç kullanılmayan) bulunmamaktadır.

Eğitim talebi iş akışı ağında VEYA-dağılımları kullanıldığından serbest seçimli bir iş akışı ağıdır. Ayrıca iyi yapılıdır . Çünkü VEYA-dağılımı ile başlayan yollar , p10 yerinde VEYA-bileşimi ile birleşmektedirler. Dolayısıyla bu iş akışı ağı verimli bir şekilde kullanılabilir.

7. SONUÇ

Bu çalışmada Petri ağlarının özellikleri ve analiz yöntemleri incelenerek bir ayrık olaylı sistemin modellenmesine uygulanmıştır. Petri ağlarının özellikleri ve analiz yöntemleri incelenmiştir. Ayrık olaylı sistemlerin Petri ağları ile modellenmesini cazip kılan, bir çok analiz tekniğinin geliştirilmiş olmasıdır. Bu çalışmada bu analiz teknikleri incelenerek ayrık olaylar içeren bir iş akışı, Petri ağı modeli üzerinde incelenmiştir.

İş akışlarının Petri ağları ile modellenme nedenleri, bunun avantajları, iş akışlarında kullanılan yapıların Petri ağına belirtimi verilmiştir. Petri ağları ayrık dinamik modelleri temsil etmeye çok uygundur. Bir iş akışı da ayrık bir modeldir çünkü sürekli zaman eksenine karşılaştırıldığında zaman içinde aktivitelerin yürütüldüğü noktalar ayrık bir küme oluşturur. Petri ağı sağlam matematiksel tanımlara sahiptir. Bu tanımların sentaksı ve anlamı kesin olarak bellidir. Eğer bir iş akışı Petri ağı ile ifade edilirse belirgin özellikleri, örneğin son işaretleme (ulaşılacak istenen nokta) ulaşılabilirliği matematiksel olarak ispatlanabilir. Petri ağının gerçekleşmesiyle ilgili çok sayıda analiz metodu vardır. Ayrıca Petri ağı ile modellenen iş akışlarının analizi de incelenmiştir. İş akışı tanımını modelleyen bir Petri ağına iş akışı ağı (Workflow net) denir. Petri ağı ile modellenen bir iş akışının sağlam olabilmesi için gerekli koşullar verilmiştir. Sonuç olarak bir ayrık olaylar içeren iş akışları, Petri ağı teorisiyle verimli bir şekilde modellenip incelenebilir.

KAYNAKLAR

- [1] **Cassandras, G.C. and Lafortune, S.,** 1999. Introduction to Discrete Event Systems, Kluwer Academic Publishers , M A.
- [2] **Peterson, L. James,** 1981. Petri Net Theory and The Modeling of Systems Prentice-hall , New Jersey .
- [3] **W.M.P. van der Aalst, van Hee, K M and Houben, G J ,** 1994. Modelling and analysing workflow using a Petri-net based approach, *Proc. 2 nd Workshop on Computer-supported Cooperative Work, Petri nets and related formalisms* 31-50
- [4] **N. R. Adam, V. Atluri, and W.-K. Huang,** 1998. Modeling and Analysis of Workflows Using Petri Nets, *Journal of Intelligent Information Systems, Special Issue on Workflow and Process Management*, **10(2)**, 131–158,
- [5] **W.M.P. van der Aalst,** 1996. Three Good Reasons for Using a Petri-net-based Workflow Management System, *Proceedings of the International Working Conference on Information and Process Integration in Enterprises (IPIC'96)*, Cambridge, Massachusetts, 179–201
- [6] **W.M.P. van der Aalst,** 1996. Petri-net-based Workflow Management software *Proceedings of the NFS Workshop on Workflow and Process Automation in Information Systems*, Athens, Georgia, 114–118
- [7] **Konstantin Knorr,** 2000. Dynamic Access Control through Petri Net Workflows, *In proceedings of the 16th Annual Computer Security Applications Conference*, New Orleans, 159-167
- [8] **W.M.P. van der Aalst,** 1998. The Application of Petri Nets to Workflow Management, *The Journal of Circuits, Systems and Computers*, **8(1)** , 21-66

- [9] **W.M.P. van der Aalst**, 1996. Structural Characterizations of Sound Workflow Nets, *Computing Science Reports* , Eindhoven University of Technology, Eindhoven, 96/23
- [10] **W.M.P. van der Aalst**, 1997. Verification of Workflow Nets, Application and Theory of Petri Nets 1997, volume 1248 of Lecture Notes in *Computer Science*, Berlin, 407–426.
- [11] **W.M.P. van der Aalst**, 1995. A class of Petri net for modeling and analyzing bussiness processes, *Computing Science Reports*, Eindhoven University of Technology, Eindhoven, 95/26
- [12] **W.M.P. van der Aalst**, 1998. Finding Errors in the Design of a Workflow Process: A Petri-net-based Approach, volume 98/7 of *Computing Science Reports*, Lisbon, Portugal, 60-81
- [13] www.daimi.aau.dk/Petrinets/

ÖZGEÇMİŞ

ÇELİKMIH Kadir

1977 yılında Konya’da doğdu. Liseyi Özel Diltaş Lisesi’nde bitirdikten sonra 1994 yılında İ.T.Ü. Kontrol ve Bilgisayar Mühendisliği Bölümü’nü okumaya hak kazandı. 1998 yılında buradan mezun olduktan sonra aynı yıl içinde İ.T.Ü. Kontrol ve Bilgisayar Mühendisliği Yüksek Lisans eğitimi almaya hak kazandı. 1999 yılında Detay Danışmanlık’ta SAP Abap/4 danışmanı olarak işe girdi ve halen aynı şekilde çalışmalarını sürdürmektedir.