

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**CPN TOOLS PROGRAMI KULLANILARAK SMTP (BASİT POSTA
AKTARIM PROTOKOLÜ)’NİN MODELLENMESİ**

YÜKSEK LİSANS TEZİ

Mehmet Fikri OTMAN

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Mühendisliği Programı

OCAK 2014

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**CPN TOOLS PROGRAMI KULLANILARAK SMTP (BASİT POSTA
AKTARIM PROTOKOLÜ)’NİN MODELLENMESİ**

YÜKSEK LİSANS TEZİ

**Mehmet Fikri OTMAN
(504041238)**

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Mürvet KIRCI

OCAK 2014

İTÜ, Fen Bilimleri Enstitüsü'nün 504041238 numaralı Yüksek Lisans Öğrencisi **Mehmet Fikri OTMAN**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**CPN Tools Programı Kullanılarak SMTP(Basit Posta Aktarım Protokolü)’nin Modellenmesi** ” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. Mürvet KIRCI**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Ece Olcay GÜNEŞ**
İstanbul Teknik Üniversitesi

Doç. Dr. Kamuran Nur BEKİROĞLU
Yıldız Teknik Üniversitesi

Teslim Tarihi : **16 Aralık 2013**
Savunma Tarihi : **17 Ocak 2014**

Hocalarına,

ÖNSÖZ

Tez çalışmamda rehberliğini ve değerli zamanını esirgemeyen, bilgi ve önerilerini her zaman paylaşan değerli danışman hocam Doç.Dr. Mürvet KIRCI'ya bugünlere gelmemde emeği geçmiş tüm değerli hocalarıma teşekkürü bir borç bilirim.

Ayrıca değerli yardımlarından dolayı Emre ÖZDİLEK'e teşekkür ederim.

Aralık 2013

Mehmet Fikri OTMAN
Elektronik ve Haberleşme Müh.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖNSÖZ.....	vii
İÇİNDEKİLER.....	ix
KISALTMALAR.....	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET.....	xvii
SUMMARY.....	xix
1. GİRİŞ	1
1.1 CPN Tools’da Uygulanan Petri Ağları Sınıfı	1
1.1.1 Petri Ağ Grafiği ve CPN ML.....	2
1.2 Hedef ve CPN Tools’da Temel Fonksiyonlar.....	5
1.2.1 CPN Tools hedefi.....	5
1.2.2 CPN Tools’da temel işlevler.....	6
1.3 CPN Tools’da Organizasyon.....	6
1.3.1 Ana pencere alanları.....	7
1.3.2 Araçlarla(tools) çalışmak.....	8
1.3.3 İçeriğe duyarlı menüler.....	9
1.3.4 Modelin yapısı.....	10
1.3.5 Yardım sisteminin organizasyonu.....	11
1.3.6 Geribildirim CPN Tools’da yansıtma.....	11
1.4 CPN Tools Araç Kutusu.....	14
1.4.1 Ağ araçları.....	14
1.4.2 Oluşturma araçları.....	15
1.4.3 Benzetim(Simülasyon) araçları.....	16
1.4.4 Diğer araçlara(tools) genel bakış.....	16
1.5 CPN ML Temelleri.....	17
1.5.1 Basit renk kümeleri.....	18
1.5.2 Bileşik renk kümeleri.....	22
1.5.3 Değişkenler ve sabitler bildirgesi.....	23
1.5.4 Fonksiyonlar.....	23
1.5.5 Rastgele fonksiyonlar.....	25
1.5.6 Multi-sets.....	26
1.5.7 Zamanlı Multi-setler.....	27
1.6 Modellerin Dilinin Tanımlanması.....	28
1.6.1 Yer yazıtları.....	28

1.6.2 Ok yazıtları.....	29
1.6.3 Geçiş yazıtları.....	29
2. SMTP (BASİT POSTA AKTARIM PROTOKOLU).....	33
2.1 SMTP Temelleri.....	33
2.2 SMTP Model Ve Protokol.....	33
2.3 User Agent(Kullanıcı Ajanı) (UA).....	34
2.3.1 E-posta gönderimi.....	34
2.3.2 E-posta başlığı formatı.....	35
2.3.3 E-posta alımı.....	35
2.3.4 Ertelenmiş iletim.....	35
2.3.5 Takma adlar.....	36
2.4 MTA(Mail Transfer Agent).....	36
2.4.1 SMTP e-posta alışverişi akışı.....	36
2.4.2 SMTP komutları.....	37
2.4.3 E-posta servisi çeşitleri.....	38
2.4.4 SMTP yanıtları.....	38
2.5 SMTP Sunucu.....	38
2.6 On-Demand Mail Relay(Talep Üzerine E-posta Rölesi).....	38
2.7 MIME(Multipurpose Internet Mail Extensions).....	39
2.7.1 MIME-version.....	40
2.7.2 Content-type.....	40
2.7.3 Content-transfer-encoding.....	40
2.7.4 Content-Id.....	41
2.7.5 Content-Description.....	41
2.7.6 MIME güvenlik şeması.....	41
2.8 E-Posta İletim Tipleri.....	41
2.9 E-Posta Erişim Modları.....	42
3. SMTP CPN Tools MODELLERİ.....	45
4.SONUÇ VE ÖNERİLER.....	51
ÖZGEÇMİŞ.....	55

KISALTMALAR

CPN	: Renkli Petri Ağları
CPN Tools	: Renkli Petri Ağları Araçları
KA	: Kullanıcı Ajanı
MIME	: Çokamaçlı İnternet Posta Uzantıları
ML	: Programlama Dili
MTA	: Posta Aktarım Aracısı
SMTP	: Basit Posta Aktarım Protokolü
UA	: Kullanıcı Ajanı

ÇİZELGE LİSTESİ

Sayfa

Çizelge 1.1 : İnternete Bağlanma Modelinde Kullanılan Düğümlerin Anlamları21

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : Yer.....	2
Şekil 1.2 : Geçiş.....	2
Şekil 1.3 : CPN Tools (Renkli Petri Ağı Araçları) Temel Bileşenleri.....	3
Şekil 1.4 : CPN Tools için Örnek Bir Uygulama (Başlangıç Durumu).....	3
Şekil 1.5 : Kitap Alma Geçişi Ateşlendikten Sonraki Durum.....	3
Şekil 1.6 : CPN Tools Açılış Penceresi.....	7
Şekil 1.7 : CPN Tools Ana Penceresi.....	7
Şekil 1.8 : CPN Tools Öğelerin Açılmış Durumu.....	8
Şekil 1.9 : Create Aracı.....	8
Şekil 1.10: Geçiş Seçilmiş Durum.....	9
Şekil 1.11: Yeni Ağ Oluşturma Aracı Tıklanmış Durumda.....	9
Şekil 1.12: Farenin Sağ Tıklanması Durumu.....	10
Şekil 1.13: Araç Kutusunu Kapama İşlemi.....	10
Şekil 1.14: Yeni Bir Sayfa Oluşturma İşlemi.....	11
Şekil 1.15: Araç Kutusunda Tanımlı Paletler.....	14
Şekil 1.16: Ağ İşlemleri.....	15
Şekil 1.17: Oluşturma Paleti.....	15
Şekil 1.18: Yer ve Geçişlerin Ayrıntıları.....	16
Şekil 1.19: Benzetim(Simulasyon) Paleti.....	16
Şekil 1.20: İnternete Bağlanma CPN Tools Modeli Başlangıç Durum.....	20
Şekil 1.21: İnternete Bağlanma Modeline Ait Tanımlamalar.....	20
Şekil 1.22: İnternete Bağlanma Modeli 1.Örnek Ateşleme Sonucu.....	21
Şekil 1.23: İnternete Bağlanma Modeli 2.Örnek Ateşleme Sonucu.....	22
Şekil 1.24: If Yapısına Ait Uygulama Örneği.....	25
Şekil 1.25: Seçme Olayı Geçişinin Ateşlendiği Örnek Bir Görünüm.....	25
Şekil 1.26: Yer Ayrıntıları.....	28
Şekil 1.27: Ok Yazıtı Gösterimi.....	29
Şekil 1.28: Geçiş Ayrıntıları.....	29
Şekil 3.1 : SMTP Protokolünün CPN Tools Modeli.....	45
Şekil 3.2 : SMTP Protokolünün CPN Tools Modeline Ait Tanımlamalar.....	45
Şekil 3.3 : SMTP Protokolünün Zamanlanmış CPN Tools Modeli.....	47
Şekil 3.4 : SMTP CPN Tools Zamanlanmış Modeline Ait Tanımlamalar.....	48

CPN TOOLS PROGRAMI KULLANILARAK SMTP (BASİT POSTA AKTARIM PROTOKOLÜ)’NİN MODELLENMESİ

ÖZET

Eşzamanlı sistemlerin geliştirilmesindeki güçlük dikkate alındığı zaman sistemin uygulanabilir bir modelinin inşa edilmesinin ne kadar gerekli olduğu ortaya çıkmaktadır. Modelin kurulması ve simülasyonunun gözlemlenmesi genellikle söz konusu sistemin işleyişiyle ilgili yeni fikirler vermektedir. CPN (Renkli Petri Ağları) eşzamanlı sistemlerin modellerinin tasarlanması ve özelliklerinin incelenmesi için oluşturulan grafiksel bir dildir. Ayrık olayların modellenmesinde yaygın olarak kullanılmaktadır. CPN Tools programı CPN modellerinin yapılandırılmasını ve incelenmesini sağlayan bilgisayar yazılımı olarak dikkat çekmektedir. Renkli Petri ağları (CPN) tipik uygulama alanları arasında haberleşme protokolleri, veri ağları, telekomünikasyon sistemleri sayılabilir. Bu tez çalışmasının birinci bölümünde Petri Ağ grafiği ve CPN ML ye kısaca değinilmiş , basit bir örnek üzerinden tanımlamalar verilerek CPN Tools programına ilişkin çalışma mantığı açıklanmaya çalışılmıştır. Hedef ve CPN Tools’da kullanılan temel fonksiyonlardan söz edilmiştir. CPN Tools programının organizasyonu ve kullanılan araçlar anlatılmaya çalışılmıştır. CPN Tools programının temel bileşenleri olan yer, geçiş ve ok kavramları açıklanmıştır. CPN ML temelleri anlatılmaya çalışılmıştır. İnternete bağlanma olayının modellendiği diğer bir örnekte ise internete erişmek için eşzamanlı olarak IP alan bilgisayarlardan oluşan sistemin benzetimi gösterilmiştir.

İnternet kullanımının vazgeçilmez bir gereklilik haline geldiği günümüzde internet üzerinden e-postaların alımı ve iletimi SMTP(Simple Mail Transport Protokol)(Basit Posta Aktarım Protokolü) kullanılarak gerçekleştirilmektedir. E-posta dağıtımı gönderici SMTP sunucusu ile alıcı SMTP sunucusu arasında komutlar , yanıtlar ve posta bilgisinin aktarımı esasına dayanan bir SMTP işlemidir. Bir e-postanın göndericiden alıcıya ulaşmasının sağlanması sürecinde gönderici SMTP sistemi e-postayı internet ortamına göndermektedir. Alıcı SMTP sistemi ise e-postayı verilerin karşılıklı aktarıldığı bir iletim ortamından kabul ederek posta kullanıcı temsilcisine aktarmakta ya da posta kullanıcı temsilcisinin erişebileceği dosya sisteminde saklamaktadır. Tezin ikinci bölümünde SMTP protokolü anlatılmıştır. SMTP protokolünün temellerine değinilmiş ve SMTP protokolünün e-postaları farklı hostlar arasında iletmek amacıyla kullanılmakta olan yaygın bir mekanizma olduğu vurgulanmıştır. SMTP’nin çalışma modeli ele alınarak kısaca açıklanmıştır. Ayrıca SMTP protokolünün temel bileşenleri olan UA(Kullanıcı Ajanı) ve MTA(Posta Aktarım Ajanı) üzerinde durulmuştur. MIME(Multipurpose İnternet Mail Extentions) yapısı kısaca açıklanmıştır. E-posta erişim modları anlatılmıştır.

Üçüncü bölümde SMTP protokolünün CPN Tools kullanılarak oluşturulan modelleri açıklanmıştır.

Son Bölüm olan dördüncü bölümde hazırlanan çalışmanın sonuçları ve önerilere yer verilmiştir.

MODELLING OF SMTP (SIMPLE MAIL TRANSFER PROTOCOL) BY USING CPN TOOLS PROGRAMME

SUMMARY

When we consider the difficulty in the development of concurrent systems, the necessity of building a practicable model emerges. Establishment of the model and observation of the simulation generally gives new ideas concerning the operation of the mentioned system. CPN (Coloured Petri Nets) is a graphical language which is built for the design of concurrent systems and analysis of their characteristics. It is widely used in the modelling of discrete events. CPN tools programme attract attention as the computer software which provides construction and analysis of CPN models. Among typical implementation fields of Coloured Petri Nets (CPN) we can mention communication protocols, data nets, telecommunication systems. In the first chapter of this thesis study it is dealt shortly Petri net graphic and CPN ML, on a simple example it is tried to explain the operation mentality of CPN Tools programme by giving declarations. It is talked about the objective and basic functions used in CPN Tools. It is tried to describe organization of CPN Tools programme and tools used in it. Place, transition and arc concepts which are basic components of CPN Tool programme are explained. Fundamentals of CPN ML are tried to describe. In another example which models the event of connecting to the internet, it is demonstrated the simulation of system of computers that takes IP concurrently to reach the internet.

Nowadays that usage of the internet became an indispensable necessity, receiving and transmission of e-mails via internet is realized by using SMTP (Simple Mail Transport Protocol). E-mail delivery is an SMTP operation which is based on the principle of commands, replies and mail information transfer between sender SMTP server and receiver SMTP server. In the process of an e-mail's arrival from sender to receiver, sender SMTP system sends e-mail to the internet environment. Receiver SMTP system either transfers e-mail to mail user agent by accepting it from a transmission environment where data is mutually transferred or stored in the file system where mail user agent may reach. In the second chapter of the thesis SMTP protocol is described. Fundamentals of SMTP protocol are dealt and it is emphasized that SMTP protocol is a widespread mechanism used for sending e-mails between different hosts. Operation model of SMTP is shortly explained. Additionally UA (User Agent) and MTA (Mail Transfer Agent) which are basic components of SMTP protocol are mentioned. MIME (Multipurpose Internet Mail Extensions) structure is described briefly. E-mail access modes are explained. In the third chapter SMTP protocol models produced by using CPN Tools, are explained.

In the fourth chapter which is the last chapter, results and suggestions are given. CPN Tools is a computer programme which is used to design and analysis of CPN models. By using CPN tools with the help of the simulation it is possible to observe the behaviour of a system as a model. CPN Tools is successful to drive the performance of concurrent communication systems. In Coloured Petri Nets it is possible to see that different types of tokens are used. The concept of a coloured Petri net of CPN Tools is complicated. To represent the durations of actions in real-life objects timed Petri nets use the concept of model's time. It is mentioned that in CPN Tools the language of models' description constitutes a combination of Petri net graph and programming language CPN ML (Markup Language). It is explained that in CPN Tools special programming language is considered for description of attributes of net elements. CPN Tools is a useful tool in models' design and analysis. In modelling of telecommunications, CPN Tools is used for specification and verification of communication protocols, estimation of networks throughput and QoS, design of telecommunication systems. In the second part of the thesis it is explained that SMTP protocol is widely used to send e-mails from server to server through the internet environment.

Electronic mail (e-mail) is one of the most popular network services nowadays. Most e-mail systems that send mail over the Internet use simple mail transfer protocol (SMTP) to send messages from one server to another. The messages can then be retrieved with an e-mail client using either post office protocol (POP) or Internet message access protocol (IMAP).

SMTP is used as the common mechanism for transporting electronic mail among different hosts within the transmission control protocol/Internet protocol (TCP/IP) suite. It is an application layer protocol. Under SMTP, a client SMTP process opens a TCP connection to a server SMTP process on a remote host and attempts to send mail across the connection. The server SMTP listens for a TCP connection on a specific port (25), and the client SMTP process initiates a connection on that port. Mail that arrives via SMTP is forwarded to a remote server, or it is delivered to mailboxes on the local server. POP3 or IMAP allow users download mail that is stored on the local server. The end-to-end method is used between organizations, and the store-and-forward method is chosen for operating within organizations that have TCP/IP and SMTP-based networks.

A SMTP client will contact the destination host's SMTP server directly to deliver the mail. It will keep the mail item from being transmitted until it has been successfully copied to the recipient's SMTP.

The client SMTP is the one, which initiates the session (that is, the sending SMTP) and the server is the one that responds (the receiving SMTP) to the session request. An SMTP-based process can transfer electronic mail to another process on the same network or to another network via a relay or gateway process accessible to both Networks. An e-mail message may pass through a number of intermediate relay or gateway hosts on its path from a sender to a recipient. The exchange of mail using TCP is performed by an MTA. The MTA maintains a mail queue so that it can schedule repeat delivery attempts in case a remote server is unable.

Also the local MTA delivers mail to mailboxes, and the information can be downloaded by the UA. The sender-SMTP will generate commands, which are replied to by the receiver-SMTP.

Both the SMTP client and server should have two basic components: UA and local MTA. There are few cases of sending electronic-mail messages across networks. In the first case of communication between the sender and the receiver across the network, the sender's UA prepares the message, creates the envelope, and puts message in the envelope. The MTA transfers the mail across the network to the TCP-port 25 of the receiver's MTA.

In the second case of communication between the sending host (client) and the receiving host (server), relaying could be involved. In addition to one MTA at the sender site and one at the receiving site, other MTAs, acting as client or server, can relay the electronic mail across the network. The e-mail gateway can also receive electronic mail in the SMTP format, change it to another format, and then send it to the MTA of the client that does not use the TCP/IP protocol suite. In various implementations, there is the capability to exchange mail between the TCP/IP SMTP mailing system and the locally used mailing systems.

Sending mail through a mail gateway may alter the end-to-end delivery specification, because SMTP will only guarantee delivery to the mail-gateway host, not to the real destination host, which is located beyond the TCP/IP network. The UA is a program that is used to send and receive electronic mail.

Electronic mail is sent by a series of request–response transactions between a client and a server. The header includes a number of key words and values that define the sending date, sender's address, where replies should go, and some other information. The UA periodically checks the content of the mailboxes. It informs the user about mail arrival by giving a special notice. When the user tries to read the mail, a list of arrived mail packages is displayed. Each line of the list contains a brief summary of the information about a particular package in the mailbox. The SMTP protocol allows delayed delivery, and the message can be delayed at the sender site, the receiver site, or the intermediate servers.

In the case of delaying at the sender site, the client has to accommodate a spooling system, in which e-mail messages are stored before being sent. A message created by the user agent is delivered to the spool storage.

Upon receiving the message, the server-MTA stores it in the mailbox of the receiver. In this case, the receiver can access the mailbox at any convenient time. The SMTP mechanism allows one name, an alias, to represent several e-mail addresses.

MTAs transfer actual mail. The system must have the client MTA for sending e-mail and the server MTA for receiving mail.

The SMTP protocol (RFC 821) defines how commands and responses must be sent by the MTAs. The client sends commands to the server, and the server responds with numeric reply codes and optional human-readable strings.

The SMTP server sends and receives mail from other Internet hosts using the SMTP. The SMTP server processes all incoming and outgoing mail. Outgoing mail is spooled until the SMTP server can confirm it has arrived at its destination; incoming mail is spooled until users access it by using a POP3 or IMAP4 mail client. Spooling allows the transfer from client and server to occur in the background.

On-demand mail relay (ODMR), also known as authenticated TURN (ATRN), is an e-mail service that allows a user to connect to an Internet service provider (ISP), authenticate, and request e-mail using a dynamic IP address (instead of static IP addresses used in a “traditional” SMTP model) from any Internet connection.

Multipurpose Internet Mail Extensions is a supplementary protocol that allows non-ASCII data to be sent through e-mail. MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers them to the client MTA to be sent through the Internet. The message at the receiving side is transformed back to the original data.

The MIME protocols define five header lines that can be added to the original header section to define the transformation parameters: MIME-version, contenttype, content-transfer-encoding, content-id, and contentdescription.

The S/MIME is a security scheme for the MIME protocol. It was developed by RSA Security and is an alternative to the pretty good privacy (PGP) encryption and digital signature scheme that uses public-key cryptography.

To reach its final destination, an e-mail message should be handled by a mail server, the mail access protocol, and the mail client.

1. GİRİŞ

CPN (Renkli Petri Ağları) eşzamanlılık, iletişim ve senkronizasyonun önemli rol oynadığı sistemlerin bir modelleme dilidir. Petri ağları modelleme , eşzamanlılık , iletişim ve senkronizasyon için grafik gösterimin temel ilkelerini sağlar. Bir sistemin CPN modeli, sistemin durumunu ve sistemin durumunu değiştirmesine neden olabilen olayları(geçişler) temsil eden uygulanabilir bir modeldir. CPN dili, modellerin bir grubu olarak bir modelin organize olmasını sağlar ve modellenen sistemde olayların uygulanması için geçen zamanı temsil edebilmesini olanaklı hale getiren zaman konseptini içerir.

CPN Tools CPN modellerinin yapılandırılması ve analizini sağlayan bilgisayar aracıdır. CPN Tools kullanarak, simülasyon yardımıyla modellenen davranışlarını araştırmak ve durum uzayı yöntemleri ve model kontrolü ile özellikleri doğrulamak ve simülasyon esaslı performans analizlerini yürütmek mümkündür. CPN Tools ile kullanıcı etkileşimi araç paletleri ve işaretleme menüleri gibi etkileşim tekniklerini kullanan CPN modelinin grafiksel gösteriminin doğrudan kullanılması esastır (Url 1).

CPN Tools modellerin gösterimi için Petri ağları dilini kullanan özel bir benzetim sistemidir. Sistem Danimarka'da Aarhus üniversitesinde geliştirilmiştir ve <http://www.daimi.au.dk/CPNTools/> web sayfası aracılığıyla ticari olmayan organizasyonlar için yüklemesi ücretsiz dağıtımı sağlanmıştır. Servisin seviyesi CPN Tools'un girişim sistemi olarak sınıflandırılmasına izin verir. Gerçek hayattaki pek çok projede özellikle de telekomünikasyon alanındaki projelerde kullanılmıştır.

1.1 CPN Tools'da Uygulanan Petri Ağları Sınıfı

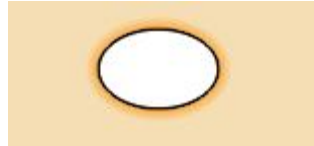
CPN Tools modellerin tanımlanması için Petri ağ sınıfı önerir. Standart sınıflamaya göre böyle ağlar hiyerarşik zamanlı renkli petri ağları olarak adlandırılır. Geniş kapsamlı bir algoritmik sistem meydana getirdikleri kanıtlanmıştır.

Renkli Petri Ağlarının en basit bir kavramı farklı tipte jetonlar kullanmasıdır. Böyle ağlar çoğunlukla genelleştirilmiş renkli ağlar olarak adlandırılır çünkü jetonun tipi dil programlamasındaki gibi soyut(teorik) data tipi şeklinde tanımlanır. Zamanlı Petri ağları gerçek hayat nesnelerindeki eylemlerin sürelerini göstermek için modelin zamanı kavramını kullanır. Geçişin ateşlenmesinin aniden oluştuğu Petri Ağı zamanlanmış ağdaki geçişin ateşlenmesi belirli(kesin) süre ya da zamanlanmış gecikme ile ilgilidir. Gerçek hayat nesnelerinin zamanlanmış karakteristiklerinin analizine izin verir. Hiyerarşik(aşamalı) ağlar komplike modellerin inşasını temin eder. Böyle ağlarda bir eleman başka bir ağla gösterilebilir.CPN Tools’da bir geçiş başka bir ek ağla yapılandırılabilir. Bu sebepten iç içe bir yapıya sahip oluruz :Ağ içinde ağ. Hiyerarşi seviyelerinin sayısı temel sınırlamaya sahip değildir.

1.1.1 Petri Ağ Grafiği ve CPN ML

CPN Tools’da modellerin tanımlanması dili Petri ağ grafiği ve CPN ML programlama dilinin bir kombinasyonunu oluşturur.

Petri Ağının grafiği iki bölümlü yönetilen grafiklerdir. Daireler ya da ovaler gibi çizilmiş yerler ve dikdörtgen veya çubuk şeklinde çizilmiş geçişler.

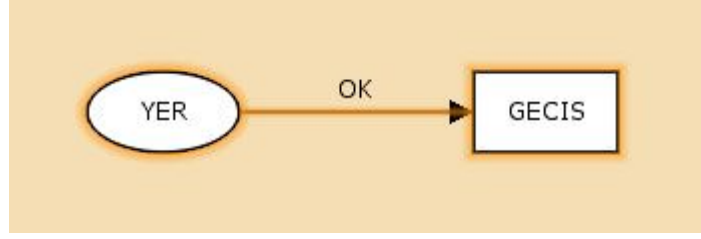


Şekil 1.1: Yer.



Şekil 1.2: Geçiş.

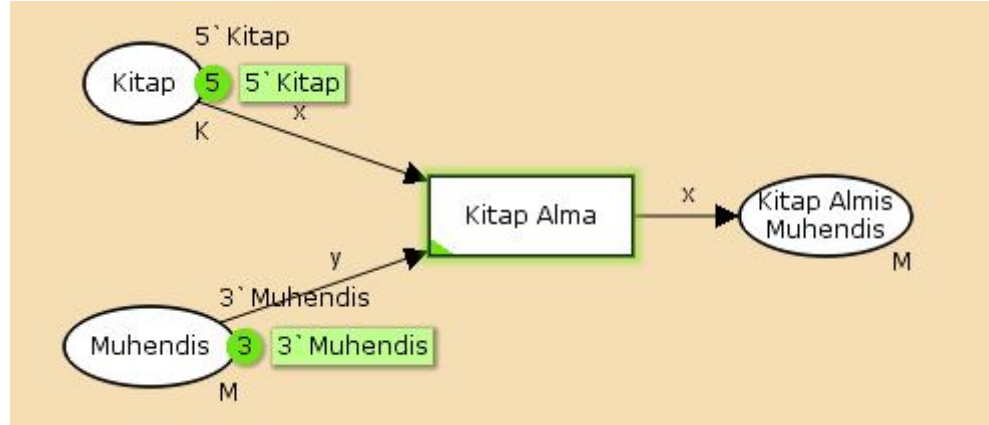
Oklar yerler ve geçişleri birleştirmek için kullanılır.



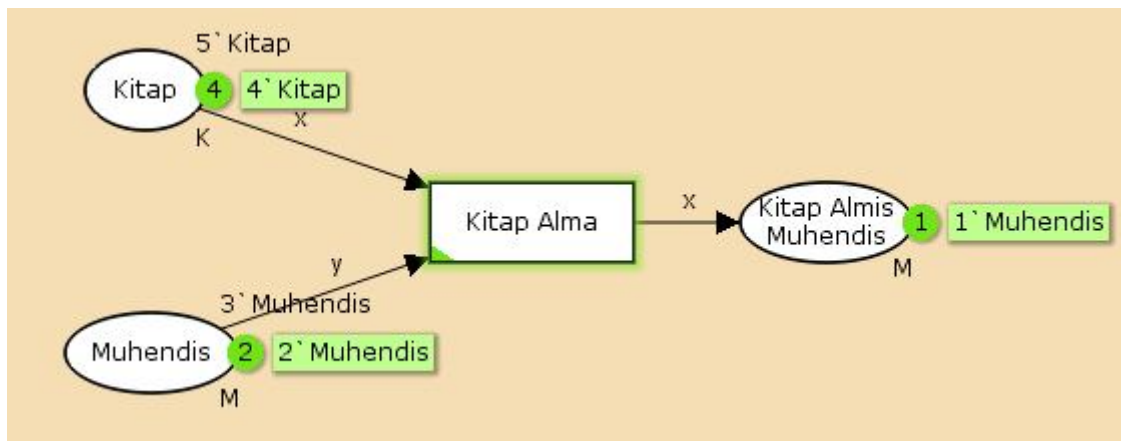
Şekil 1.3: CPN Tools (Renkli Petri Ağı Araçları) Temel Bileşenleri.

Petri ağlarında jeton kavramına da aynı zamanda dikkat edilmelidir. Jeton yerin içinde konumlanan ve geçiş ateşlemesinin sonucu olarak hareket eden dinamik bir nesnedir.

Klasik Petri ağlarında tüm jetonlar basit, temel ve aynıdır. Renkli Petri ağlarında jetonların tipleri ayırt edilebilir. Mühendisler tarafından alınan kitap örneğini göz önüne alalım, iki çeşit jetona sahibiz: *Muhendis*, *Kitap*. Bir Mühendis kitap aldıktan sonra kitap almış bir mühendis olur:



Şekil 1.4:CPN Tools için Örnek Bir Uygulama (Başlangıç Durumu).



Şekil 1.5: Kitap Alma Geçiş Ateşlendikten Sonraki Durum.

Gerçekten, CPN Tools'da özel programlama dili ağ elemanlarının niteliklerinin tanımlanmasını kapsar. Bu dil renk kümelerinin, değişkenlerin, sabitlerin,

fonksiyonların ve prosedürlerin (süreçlerin) bildirimini sağlar. Yukarıdaki örnekte böyle bildirimler kullanılmıştır.

Colset K=unit with Kitap;

Colset M=unit with Muhendis;

Var x:K;

Var y:M;

İki renk kümesi tanımlanmıştır: Kitap ile **K** ve Muhendis ile **M**. "**Muhendis**" ve "**Kitap Almis Muhendis**" yerleri Muhendis jetonlarıyla M tipindendirler. "**Kitap**" yeri **Kitap** jetonuyla K tipindendir. Geçiş ateşlemek için bir Muhendis ve bir kitabın ikisine de sahip olmalısınız. x ve y değişkenleri Yerlerden jetonları çıkarmak(almak) ve çıkış yerine yeni jeton yerleştirmek için kullanılır.

Örnek farklı tipteki jetonların işleyebileceği yolları göstermektedir. Haberleşme sistemleri modellerinde renk kümeleri daha karmaşık olabilir. Klasik Petri Ağlarına rağmen, yerler, geçişler ve oklar renkli Petri Ağları'nda niteliklere sahiptir. Yukarıdaki örnekte, yer "muhendis" adına sahiptir, M renk kümesi ilk işaretlemeye -3 muhendis ile güncel (geçerli) işaretleme-onlardan biri kitap aldıktan sonra 2 muhendis ile gösterilir. x değişkeni değişken renk kümesine göre keyfi(rasgele) bir kitap seçimine izin verir; y değişkeni keyfi rasgele bir muhendisin seçimine izin verir. "Muhendis" yerinden x değişkeni ile çıkarılan aynı muhendis "**Kitap Almis Muhendis**" yerine yerleştirilecektir çünkü çıktı olan "**Kitap Alma**" geçişinin çıkış oku olan aynı x değişkeni ile kaydedilmiştir.

1.2 Hedef ve CPN Tools'da Temel Fonksiyonlar

1.2.1 CPN Tools hedefi

CPN Tools modellerin tasarım ve analizi için kullanılmaktadır. Çeşitli mühendislik alanlarda karmaşık nesnelerin gelişiminde uygulanabilen bir sistemdir. Üretim ve iş yönetimi, askeri operasyonları planlaması ve kontrolü, üretim sistemlerinin ve robotlar ve füze gibi araçların kontrolü için yaygın olarak kullanılır. CPN Tools , ms Windows ve UNIX platformlarında uygulanır ve erken kullanılan sistem tasarımı-CPN yeni nesil özünü oluşturmaktadır. Telekomünikasyona gelince, CPN Tools protokolleri özelleştirme ve doğrulama, telekomünikasyon aygıtları ve ağ tasarımı için kullanılır.

Erken bir model, sadece kendi kalkınma sürecinde aygıtların veya ağların karakteristiklerinin tahmini için kullanıldı.

Modele dayalı kalkınmada ilk basit modeli sırayla sistemin nihai belirtimine dönüştürülmüştür. Geliştirme süreci kendi üretim veya yükleme süreçleri için gereken teknik belirtim kadar ayrıntılı oluncaya kadar modeli gerçek yaşam sisteminin daha da ayrıntılar veren süreci oluşturmaktadır.

Bu yaklaşımın avantajı, tasarımının her aşamasında ve karakteristiklerini tahmininde sistemi analiz imkanı vardır. Gerçek hayatta karmaşık nesneleri için en iyi duruma getirme görevi resmi çözümü oldukça zor ve çoğu durumda pratik olarak imkansız olduğundan en iyiye çok yakın sistemlerin tasarımına izin verir.

CPN Tools renkli zamanlı hiyerarşik petri ağları evrensel algoritmik sistemdirler. Böylece keyfi rasgele nesnelerin tanımlanmasına izin verirler. Ayrıca Renkli Petri Ağları'nın dili sistemlerin ayrıntılarıyla belirtimi özellikle bileşenleri arasında karmaşık etkileşimli sistemler için uygundur.

Zaman uyumsuz olaylar kavramı sistemlerin davranışlarındaki doğal paralelliği koruyarak açıklama yolu sağlar. Paralel işlemcilerinde ileri uygulama veya bilgisayar veri akışı mimarileri için çok uygundur.

Özel (donanım veya yazılım) işlemcilerde Petri Ağları kullanıldığında CPN Tools uygulamasının en yüksek avantajlı durumu elde edildi.

1.2.2 CPN Tools’da temel işlevler

CPN Tools temel işlevleri sıralananlardan oluşur:

- Model oluşturma (düzenleme)
- Kendi simülasyonu üzerinden modellerin davranış analizi
- Modelin durum uzay analizi ve oluşturulması
- Modellerinin oluşturulması için Renkli Petri Ağları’nın özel grafik editörü sağlanır.

Düzenleyici bilgisayarın ekranında petri ağlarının çizilmesine ve ağların unsurlarının niteliklerinin ve CPN ML dilinde yazılmış ek bildirimlerin girilmesinde izin verir.

Model birkaç sayfadan oluşabilir.Sayfalar hiyerarşik bir yapı sağlamak için birbirleri ile bağlantılıdır.

Basit yeterli modeller için tam durum uzayının üretimi (erişim grafi) mümkündür. Örneğin telekomünikasyon protokollerinin doğrulanması için en iyi yoldur. CPN Tools durum uzayının oluşmasını ve Petri Ağları’nın sınırlılık ve canlılık gibi standart özellikleri hakkında sonuçların sunulduğu otomatik raporu sağlar. Ayrıca CPN ML temelinde özel dil, kullanıcının ilgilendiği durum uzayının standart olmayan özellikleri hakkında sorguların tanımlanması ihtiyacı karşılanır. Ne yazık ki karmaşık modeller için durum uzayı çok büyük olabilir.

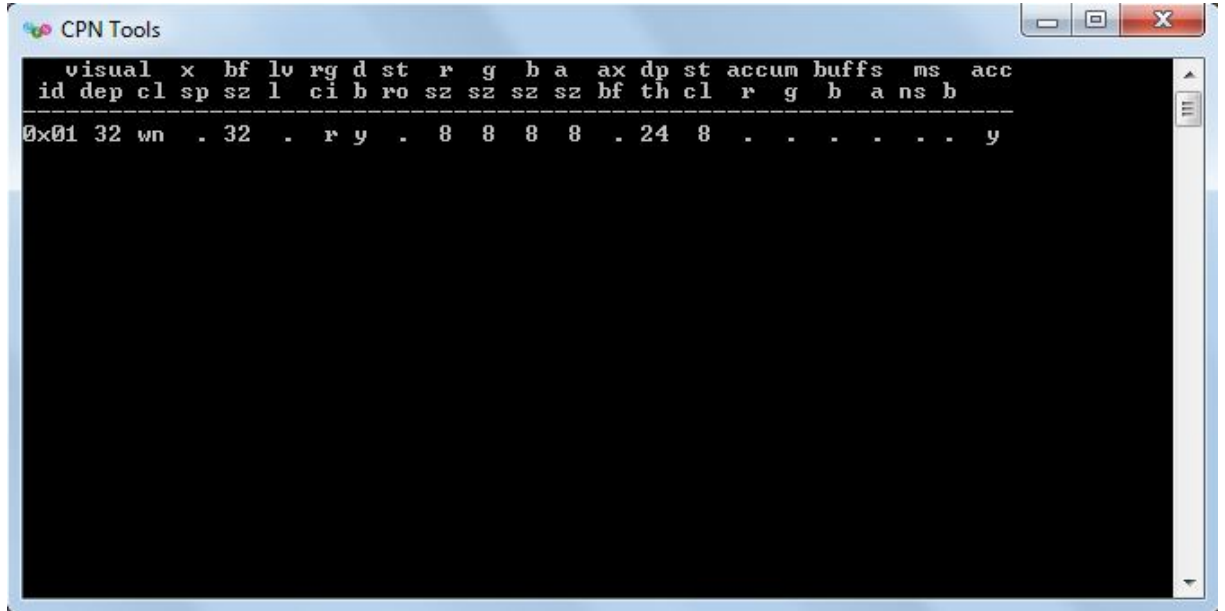
Karmaşık modellerin analizinin tek yolu davranışının simülasyonu(benzetimidir). CPN Tools modelin özelleştirilmiş sayıda adımların otomatik simülasyonunda (benzetiminde) olduğu kadar modelin hata ayıklaması için adım adım benzetimi sağlar.Büyük zaman aralıklarında benzetim modelin istatistiksel analizinin yoludur. Verim ve Qos gibi Ağ karakteristiklerinin kestirimi için kullanışlıdır.

1.3 CPN Tools’da Organizasyon

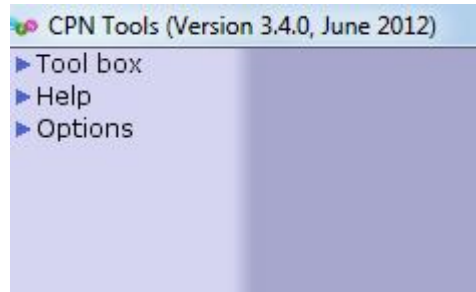
CPN Tools’da yeni bir grafiksel etkileşim kavramı uygulanmıştır. Araç kutularının araçlarını ve içeriğe duyarlı menuları kullanarak modellerin düzenlenmesini ve hızlı girişini mümkün kılar. Fareyle çift yönlü çalışma özel olanağı sağlanmıştır. Bu durumda sol fare menü ile etkileşim için ve sağ fare petri ağların çizimi ve düzenlenmesi için kullanılırken paletlerden araçların seçimi için kullanılır.

1.3.1 Ana pencere alanları

Başlatmadan sonra CPN Tools'un iki penceresi ekranda görünür.İlki siyah penceredir ;yardımcı penceredir ve alt işlevler başladıklarında çıkış mesajları için hizmet eder.



Şekil 1.6:CPN Tools Açılış Penceresi.



Şekil 1.7: CPN Tools Ana Penceresi.

İki alan içerir:çalışma alanı –gri ve içerik-beyaz. İçerik araçlar kutusu, yardım ve seçeneklerden oluşur; altlarında ağların tanımlamaları yerleştirilmiştir; yukarıdaki örnekte yüklü ağ yoktur. Çalışma alanında ağın sayfaları göz önüne getirilmiştir. CPN Tools ile etkileşim için ana pencerede grafiksel imleç(kürsör) vardır. Sistemde her zaman küçük üçgen üzerine tıklanarak açılabilen öğe anlamına gelir. Örneğin indeksteki bütün öğeleri açabiliriz.



Şekil 1.8:CPN Tools Öğelerin Açılmış Durumu.

1.3.2 Araçlarla(tools) çalışmak

Bir aracın(tool) paletini açmanın yolu onu indeksten çalışma alanına fareyle sürüklemektir. Create aracını açalım:



Şekil 1.9:Create Aracı.

Aracın paleti ekranda belirir. Araçlar paletinden bir alet almak için üzerine tıklamalıyız. Ardından imleç(küresör) karşışdüşen şeklini alır. Örneğin Create paletinden geçiş(transition) seçeriz.



Şekil 1.10:Geçiş Seçilmiş Durum.

Aracı(tool) terk etmek için palet üzerinde geriye doğru sürüklemeliyiz ve fare üzerinde tıklamalıyız ya da Esc tuşunu klavyede basmalıyız.

Her araç(tool) indekste karşı düşen dikdörtgen üzerinde tıklanarak gösterilen ve değiştirilen kendi özel seçeneklerine sahiptir.

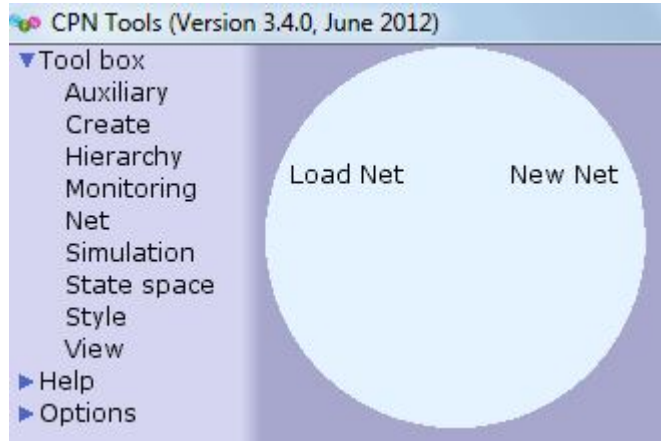
1.3.3 İçeriğe duyarlı menüler

Kullanışlı etkileşim için CPN Tools, farenin sağ tuşuna basarak ekranda görünen bir takım içerik duyarlı menüleri sağlar. Menüler, adlandırılmış sektörleriyle daire şekline sahiptir. Menü ekranda tutmak için gerekli öğeyi seçmek için fareyi hareket ettirirken düğmeyi basılı tutmak gerekir. İçerik duyarlı menü öğeleri çoğu durumda paletleri araçlarında çoğaltır. Örneğin Net paletinin new net'ini kullanarak yeni ağ oluşturabiliriz:



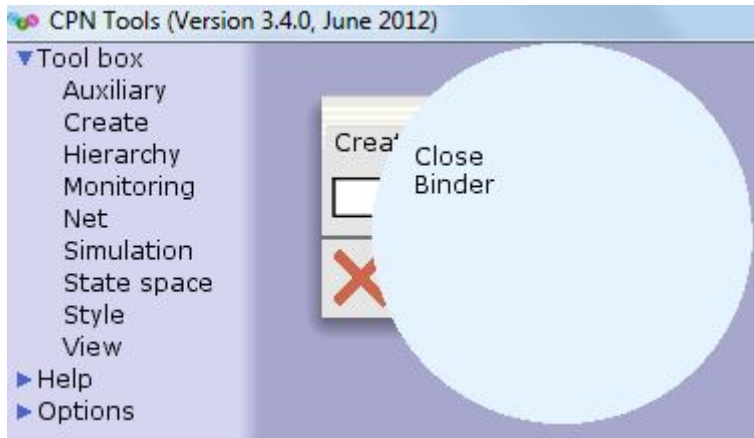
Şekil 1.11:Yeni Ağ Oluşturma Aracı Tıklanmış Durumda.

Bu da içeriğe duyarlı bir menü ile gerçekleştirilebilir:



Şekil 1.12:Farenin Sağ Tıklanması Durumu.

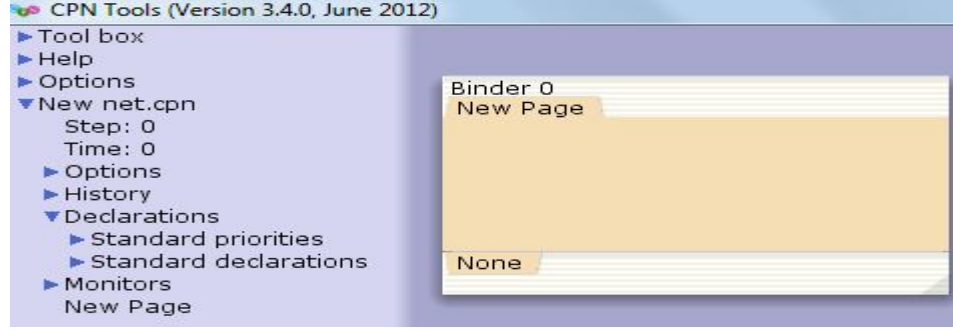
İçeriğe duyarlı menüler etkileşimi daha doğal ve hızlı yapar. Sadece bir nesne üzerinde farenin sağ düğmesi basılmalı ve gerekli eylem seçilmelidir. Bu şekilde araç kutusu kapatılabilir:



Şekil 1.13:Araç Kutusunu Kapama İşlemi.

1.3.4 Modelin yapısı

Modeller, CPN Tools'da adlandırılmış ağlardır. Açıklamaları, dizinde(indeks) standart öğelerin altında bulunmaktadır. Oluşturulmasından sonra yeni ağı göz önünde bulunduralım:



Şekil 1.14:Yeni Bir Sayfa Oluşturma İşlemi.

CPN Tools’da her ağ sıralananlara sahiptir:

İsim-.cpn tipiyle uyan dosya adı

Adım- bir simülasyonda yürütülen adımların sayısı

Zaman- Geçerli model zamanı

Tarihçe - nette gerçekleştirilmiş olan komutların listesi

Bildirimler(Açıklamalar) – renk kümelerinin , İşlevler, sabit değerlerin bildirimleri(açıklamaları)

Sayfalar-Ağın sayfalarının adları

Araçları (tools) kullanarak içinde ağ çizilebilir. Bir ağın bir sayfasının açılması için onun dizinden çalışma alanına sürüklenmesi gerekir. İçerik duyarlı menüler yeni tanımlamalar ve yeni sayfalar oluşturulması için kullanılır.

1.3.5 Yardım sisteminin organizasyonu

CPN Tools yardım için 3 türe sahiptir:Konuşma balonu, çevrimdışı yardım ve çevrimiçi yardım. Konuşma balonları karşılık gelen bir öğede birkaç saniye için imleci tuttuğunuz zaman ekranda belirir. İşaretlenmiş nesneyi tanımlarlar.

Dizin yardım öğesini tıklatarak yardımlımetin(hypertext) tarayıcı CPN Tools yardımıyla başlar:

Ağ örnekleri hesaba katılarak başarılmış CPN Tools üzerinde pek çok bilgiyi içerir. İhtiyaç halinde yardımcı sistem bu bilgiyi çağırır ancak sadece bilgisayarınız internete bağlı olduğu durumda ulaşılabilir.

1.3.6 Geribildirim CPN Tools’da yansıtma

CPN Tools geçerli sistem durumunu yansıtan grafiksel geribildirim sağlar. Sıralananlar gibi grafiksel geri bildirim türleri vardır.

- Konuşma baloncukları
- Durum baloncukları
- Auralar
- İmleç simgesini değiştirme.

Bir konuşma balonu içerik duyarlı bilgiler sağlayan sarı bir dikdörtgendir. Uygun bir nesnenin üzerinde imleç taşındığında diğerleri hafif bir gecikmeyle bazı konuşma balonları otomatik olarak görünür. Örneğin, bir sözdizimi hatası bildirim üzerine imleci hareket ettirmek hata iletisi içeren bir konuşma balonu görünmesine neden olur.

Konuşma balonları sıralananları göstermek için kullanılır:

- Sözdizimi denetimi sırasında hata mesajları
- Ağları simüle ederken hata mesajları
- Paletleri ve toolglasses araçları için araç ipuçları
- Durum baloncukları için detaylı bilgi
- ML aracı değerlendirme uygulamasının sonucu
- Kayıtlı bir ağa tam yol.Tam yolunu görmek için indekste(dizin) ağın adının üzerinde imleç hareket ettirilmelidir.

Durum kabarcıkları zaman zaman indeksin(dizin) alt kısmında görünen renk kodlu kabarcıklardır. İmleç karşılık gelen konuşma balonunu görmek için bir durum balonu üzerinde hareket ettirilebilir.

Durum balonları aşağıdaki renklerden birine sahiptir:

- Yeşil bir işlemin başarıyla tamamlandığını gösterir.
- Kırmızı bir işlem yürütülürken bir hata ortaya çıktığını gösterir.
- Hafif mor uzun bir simülasyon gibi zaman alıcı bir işlemin şu anda yürütülmekte olduğunu gösterir.

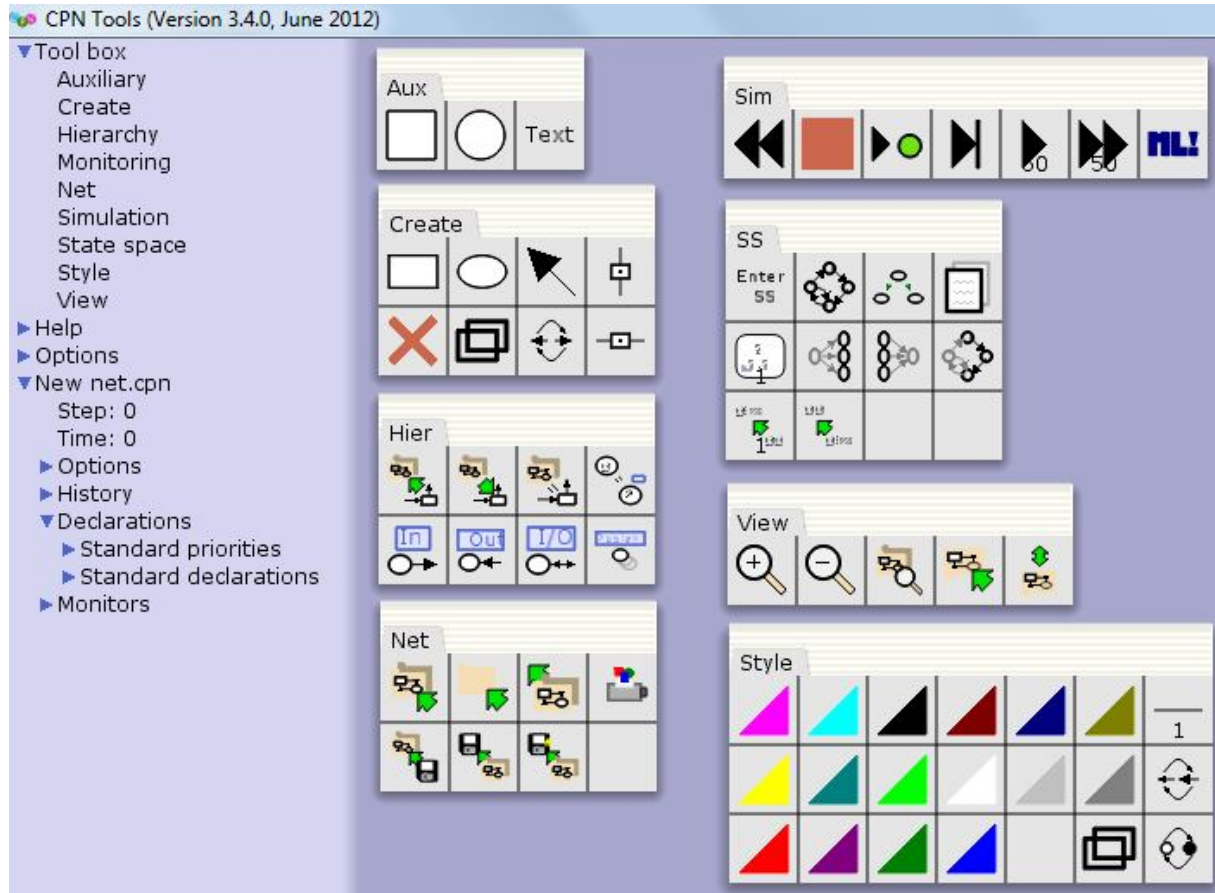
Renk kodlu auralar, belirli özelliklere sahip nesneler vurgulamak veya farklı türde nesneler arasındaki ilişkileri göstermek için kullanılır. Auralar yerler, geçişler, yazıtlar, bildiriler, sayfa sekmeleri ve sayfa adları ve ağ adları gibi dizin girişleriyle ilişkilidir:

Auralar ařağıdaki renklere sahiptir:

- (Parlak) Kırmızı sözdizimi denetimi sırasında ve ağların simulasyonu zamanında hatalı nesneleri gösterir.
- Karanlık kırmızııtrak auralar sözdizimi denetiminde geçişlerin ve yerlerin benzersiz olmayan adını belirtir.
- Yeşil ağlarının simülasyonunda etkin geçişleri gösterir.
- Koyu mavi yerler ,geçişler ve sayfalar gibi tanımlamalar ve diğer öğeler arasında bağımlılığı gösterir.
- Aqua bir nesnenin hangi nesneye ait olduğunu gösterir.
- Turuncu bir nesnenin sözdizimi denetiminin henüz başlamamış olduğunu gösterir.
- Sarı, bir nesnenin sözdizimi denetiminin şu anda gerçekleştirildiğini gösterir.
- Pembe hangi füzyon yerinin bir füzyon kümesine ait olduğunu gösterir.
- Aqua hiyerarşik ağlar ile çalışıldığında zaman bağlantı noktası(port)/soket atamaları ve altsayfa(subpage) ilişkilerini gösterir.
- Hangi eylemlerin olabilirliğini veya gerçekleşiyor olduğunu belirtmek için imleç simgesi değişir. Örneğin:
 - Standart imleç ok ya da sadece ok ucudur.
 - El imleci bir öğenin taşınabileceğini gösterir.
 - Çubuk imleci metni düzenlemenin mümkün olduğunu gösterir.
 - Çift başlı oklu imleç bir öğenin yeniden boyutlandırılabilir olduğunu gösterir. Ok başlarının yönleri hangi yönde öğenin yeniden boyutlandırılabilir olduğunu yatay mı, dikey mi ya da her ikisini de aynı anda gösterir.
 - Paletlerin birinden bir araç aldıktan sonra hangi aracın seçilmiş olduğunu belirtmek için imleç değişecektir.
 - Çoklu fazlı araçlar, yani birden fazla nesne üzerinde tıklatarak uygulanan araçlar için imleç aracın hangi fazının sonra uygulanacağını gösterir. Devralan port(bağılantı noktası) –soket(yuva) aracı ve set altsayfa aracı çoklu fazlı araçların

1.4 CPN Tools Araç Kutusu

Araç kutusu aşağıdaki araç paletlerini sağlar.



Şekil 1.15:Araç Kutusunda Tanımlı Paletler.

- Ağ araçları: Tüm ağ işlemleri için
- Araç oluşturmak: Petri ağların çizimi ve düzenlemesi için
- Araçların benzetimi: Ağın davranışının simülasyonu(benzetimi) için
- Durum uzay araçları:Durum uzayının oluşturma ve çözümlemesi
- Hiyerarşi araçları: çok düzeyli ağlar oluşturmak için
- Stil araçları: Ağların Görünüm Özellikleri için
- Görüntüleme araçları: Ölçek seçimi ve grupları vurgulama için
- Yardımcı araçlar: Ağların okunabilirliğinin geliştirilmesi için

1.4.1 Ağ araçları



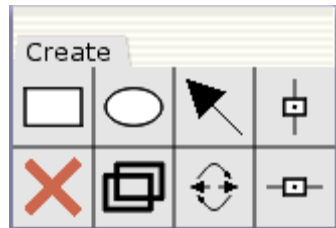
Şekil 1.16:Ağ İşlemleri.

Palet öğelerini (soldan sağa ve yukarıdan aşağı) şu anlamı vardır:

- Yeni bir ağ oluşturur
- Yeni bir sayfa oluşturur
- Ağ kapatır
- Bir ağda yüklenir
- Bir ağı kaydeder
- Yeni bir adla bir ağı kaydeder
- Bir ağı yazdırır

Yeni bir ağ oluşturmak için "Yeni Ağ Oluşturur" maddesi ile başlamalı ve "Ağ Kaydeder" öğesi ile bitirmelisiniz. Mevcut bir ağı açmak için "Bir Ağdaki Yükler" ile başlamalıdır. Ağlar .eps dosya (Genişletilmiş Post Komut dosyası) biçiminde yazdırılır ve örneğin ms Word belgeleri içine resim gibi eklenebilir. Yeni sayfalar çoğunlukla hiyerarşik ağlar için oluşturulur.

1.4.2 Oluşturma araçları



Şekil 1.17:Oluşturma Paleti.

Paletin öğelerinin aşağıdaki anlamları vardır.

- Geçiş oluşturur
- Yer oluşturur
- Ok oluşturur
- Dikey manyetik kılavuz oluşturur
- Elemanı siler
- Bir elemanı klonlar

- Okun mümkün yönleri arasında döngü yapar
- Yatay manyetik kılavuz oluşturur

Her düğümün kendi özellikler kümesi vardır. Bir ögeyi işaret ettikten sonra klavyenin Tab sekme tuşu kullanarak öznitelikleri arasında geçiş yapılabilir:



Şekil 1.18:Yer ve Geçişlerin Ayrıntıları.

Yukarıdaki örnekte, geçiş özniteliklerini kullanılmaz. Yerin geçerli işaretlemesi yeşil renkte CPN araçlar tarafından otomatik olarak yazılır.

Manyetik kılavuzlar sırayla ağı elemanlarının düzenlemek için çok kullanışlıdır. Elemanlar otomatik olarak en yakın kılavuza taşınır.

1.4.3 Benzetim(Simülasyon) araçları



Şekil 1.19: Benzetim(Simulasyon) Paleti.

Paletin öğelerinin şu anlamı vardır :

- Başlangıç durumuna gider
- Süren benzetimi(simulasyon) durdurur.
- Seçilen bir bağlayıcı ile bir geçiş yürütür(uygulanır).
- Bir geçiş yürütür.
- Ara işaretleri gösteren belirtilen sayıdaki geçişleri yürütür.
- Belirtilen sayıda geçişleri ara işaretleri görüntülenmeden çalıştırır;
- ML kodu gibi bir metin değerlendirir

Ateşlenebilir bir geçişi seçmek için üzerine veya CPN Tools'a geçişe izin vermek gerekir.

1.4.4 Diğer araçlara(tools) genel bakış

Diğer araçlar yardımcı(Auxiliary),stil(Style), görünüm(View) olarak aksesuar(Eklenti) ya da hiyerarşi(Hierarchy), durum uzayı (State Space) olarak

daha karmaşıktır. Yardımcı(Auxiliary) araçlar kutuları, daireler ve herhangi bir semantik(anlambilimsel) anlamı olmayan ama ağın(net) okunmasını kolaylaştırabilen metin etiketleri oluşturulmasına olanak sağlar.

Stil (Style) araçları renklerle önemli ağ yapıları altını çizmek , çizgi kalınlığı, okunucunun büyüklüğü-boyu (ok) , Okunabilirliği geliştirmek için elemanları doldurma için kullanılır. Bu araçların (tools) hiçbirinin ağ(net) üzerinde herhangi bir semantik (anlamsal) etkisi yoktur.

Görüntüleme (View) araçları bir sayfanın ve elemanlarının gruplandırma ve yakınlaştırma yoluyla görünümünü değiştirmek için kullanılır.

Hiyerarşi araçları, ağın hiyerarşik yapısını düzenlemek için kullanılır. Palet aşağıdan yukarıya ve yukarıdan aşağıya ağ yapılanması için araçlar içerir.

Durum Uzay (State Space) araçları, ağın durum uzaylarını hesaplamak için ,simülatör(benzetici) ve durum uzay aracı arasında durumları aktarmak için ve durum uzaylarının raporlarını üretmek için kullanılır.

1.5 CPN ML Temelleri

CPN Tools , bildirimler ve ağ yazıtları CPN ML dili kullanır. CPN ML renk kümeleri bildirimleri (veri türleri), değişkenler, işlevler, değerleri (sabitler) sağlar. Renkli Petri Ağı'nın her yerinin kendi özniteliği gereğince kesin bir renk kümesi olmalıdır; belirtilen renk kümesinin yalnızca jetonlarını içerebilir. Değişkenler ve işlevler geçişler ve okların yazıtları olarak kullanılır.

Bildirimleri, dizinde (indeks) ağın bir parçası olarak bulunmaktadır. Böyle renk kümelerinin standart önceden belirlenmiş tanımlamaları vardır: E – başlangıç(temel), INT - tamsayı, BOOL - Boolean, STRING - dizgi. İçerik duyarlı menü kullanarak standart bildirimden(açıklama) sonra kullanıcının bildirimleri eklenebilir. Ayrıca, karmaşık ağlar için CPN Tools bir dosyadan yüklenmiş olabilen dıştan gelen bildirimleri sağlar.

CPN Tools otomatik olarak, ağlarınızı onları bir ağda yüklediğinizde ya da oluştururken sözdizimi kontrol eder. Renklerin göstergeleri tarafından ne kadar uzak kontrol almış olduğunu görebilirsiniz. Renklerin göstergeleri rengin ait olduğu sayfanın adının altı çizilerek dizinde(indeks) gösterilir. Bir cilt içinde sayfa açıksa,

rengi de sayfanın ve CPN ögesinin rengin ait olduğu üst sayfa sekmesinde görüntülenir. Turuncu aura bir elemanın mevcut durumda kontrol edilmediğini işaret eder. Neti (ağı) yüklediğinizde, sözdizimi denetiminin tamamlanması birkaç dakika sürer. Bu evre sırasında elemanlar aurayı turuncudan sarıya aura yok durumuna (veya kırmızı, bir hata varsa) değişecektir. Turuncu aura kalırsa, muhtemelen eksik bir şey vardır ya da ilgili ağ ögesinde bir hata olduğundandır.

Bildirimler, üstten başlayarak kontrol edilir. Bir bildiri bir sonraki bildirimine bağımlıysa, diğer bildirim tanımlı değil diyen bir hata alırsınız. Herhangi bir bildiriminde değişiklik yapıldığında hatalı bildirimler yeniden denetlenir. Bildirimlerde hata olması durumunda hatalı bildirimin kırmızıyla altı çizilir. Ağ girişi ve tüm etkilenen sayfaların da kırmızıyla altı çizilecektir.

Kırmızı bir aura elemanın denetlendiği ama bir hatasının olduğu anlamına gelir. Kesin hata iletişiyle bir konuşma balonu görünmelidir. Hata düzeltilene kadar hatalı ögesine bağlı ögeler denetlenmez.

1.5.1 Basit renk kümeleri

CPN ML basit renk kümelerini sağlar: birim, Boolean, tamsayı ,dizgi ,numaralandırılmış, indeks(dizin). Birim renk kümesi, tek bir ögeyi kapsar. Bildirimin sözdizimi:

colset adı = **unit(birim)** [**with** (yeni birim)];

Boole değerleri Doğru ve yanlıştır . Bildirimin sözdizimi :

colset name(adı) = **bool** [**with** (new_false(yeni_yanlış), new_true(yeni_doğru))];

Seçenek doğru ve yanlış için yeni adlara izin verir.Örneğin, Evet ve hayır.

Colset Answer(Cevap) = **bool with** (no,yes);

Boolean değişkenlerine aşağıdaki işlemler uygulanabilir:

not b (b değil) Boole değeri b'nin değillenmesi(zıddı)

b1 **andalso** b2 (b1 vede b2) Boolean bağlaç(birleşim) ve

b1 **orelse** b2 (b1 yada b2) Boolean ayrılma, kapsayan(içermeli) yada

Tamsayılar, bir ondalık noktası olmayan sayılardır. Bildirimin sözdizimi:

colset name (adı) = **int** [**with** int-exp1... int-exp2];

Seçenek İnt-exp1 ve int-exp2 şeklindeki iki ifade tarafından belirlenen bir zaman aralığına tamsayı renk kümesininin kısıtlamasına imkan sağlar:

colset Dozen(düzine) = int 1..12 ile;

Aşağıdaki işlemler tam değişkenlere uygulanabilir: +, -, div, mod, abs, Int.min, Int.max.

Dizgeler(strings) çift tırnak ile çevrili yazdırılabilir ASCII karakter dizisi tarafından belirtilir. Bildirimin sözdizimi:

colset name(adı) = **string** [**with** string -exp1... string -exp2 [**and** int-exp1...int-exp2]];

Seçenek geçerli karakter aralıklarını belirtir:

Colset lowerstring(altdizge) = with "a".. "z";

Dizge değişkenleri için aşağıdaki işlemler uygulanabilir: ^-birleştirme(birbirine bağlama), String.size(dizge.ebat), alt dizge(substring)

Numaralandırılmış değerler bildirimde açıkça tanıtıcılar olarak adlandırılır. Bildirimin sözdizimi :

colset name(adı) = **with** id0 | id1 | ... | idn;

Dizinlenmiş(indekslenmiş) değerler bir tanımlayıcı ve dizin belirticisinden oluşan değerlerin dizileridir. Bildirimin sözdizimi:

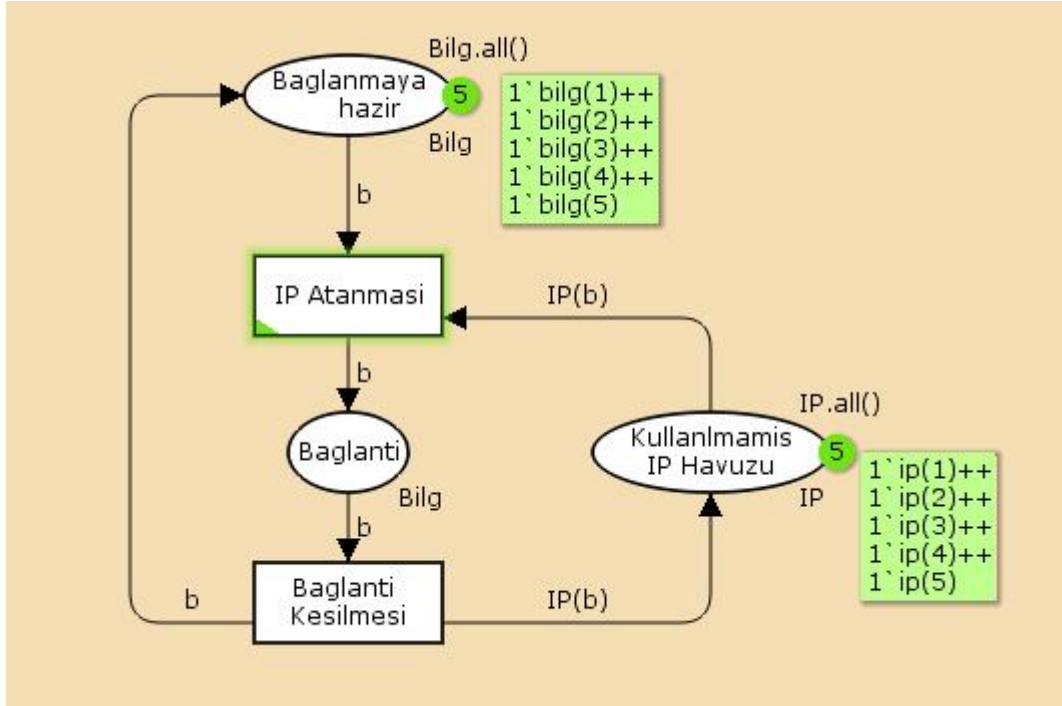
Colset name (adı) = **index** id **with** int-exp1..int-exp2;

Dizin oluşturulmuş değerlerin olan formu:.. id i veya id(i) burada i bir tamsayıdır ve int-exp1 <= i <= int-exp2.

Bilgisayarların İnternete bağlanma olayını modelleyen aşağıdaki örneği göz önüne alalım. Bilgisayarları ve IP'leri şu şekilde belirtiriz:

colset Bilg = index bilg with 1..n ;

colset IP = index ip with 1..n;



Şekil 1.20: İnternete Bağlanma CPN Tools Modeli Başlangıç Durumu.

Modelimizde IP üzerinden internete bağlanma olayı gösterilmiştir. Burada aşağıdaki bildirimler tanımlanmıştır.

```
Standard declarations
▼ colset UNIT = unit;
▼ colset INT = int;
▼ colset BOOL = bool;
▼ colset STRING = string;
▼ val n=5;
▼ colset Bilg=index bilg with 1..n;
▼ colset IP=index ip with 1..n;
▼ var b:Bilg;
▼ fun IP(bilg(i))=1` ip(i);
```

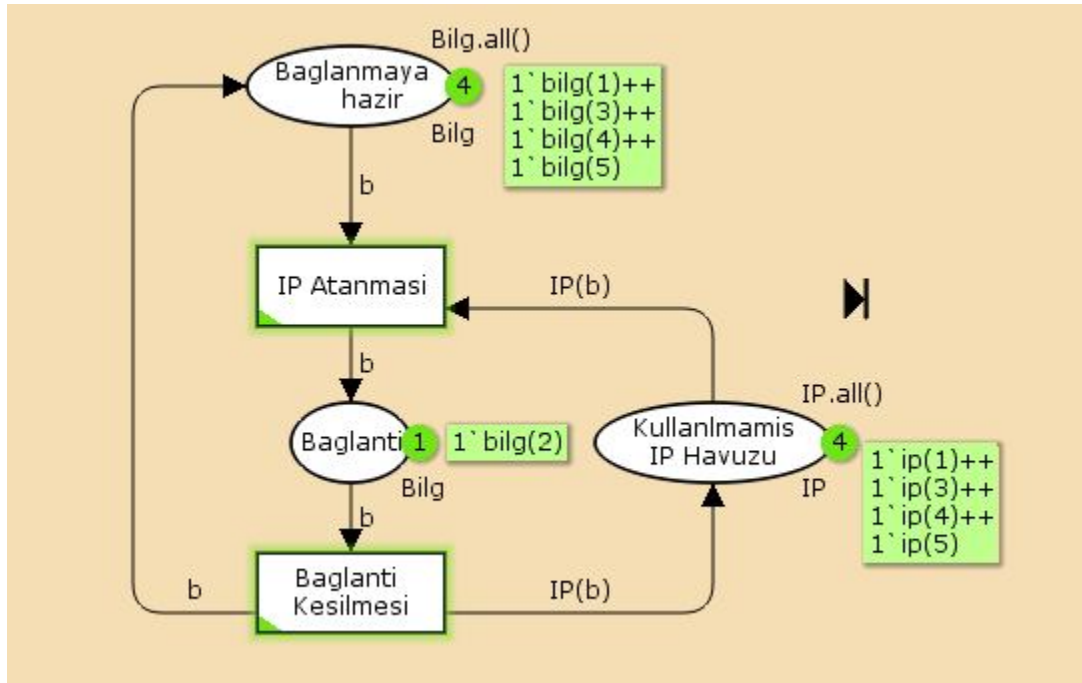
Şekil 1.21: İnternete Bağlanma Modeline Ait Tanımlamalar.

Modelde 3 yer ve 2 geçiş kullanılmıştır. Renk kümesi olarak Bağlanmaya hazır yerinde tanımlı Bilg renk kümesi ve Kullanılmamış IP Havuzu yerinde tanımlı IP renk kümesi vardır. IP(b) fonksiyonu ise bilgisayarları aynı numaralı IP'lerle eşleştirmektedir.

Çizelge 1.1 :İnternete Bağlanma Modelinde Kullanılan Düğümlerin Anlamları.

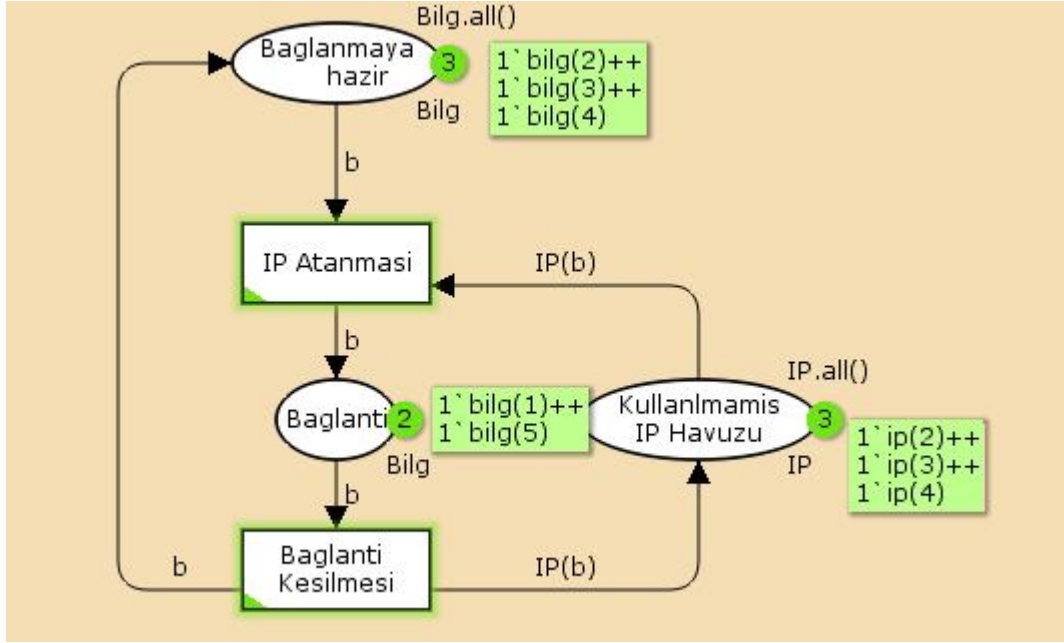
Yerler	Anlamları
Bağlanmaya Hazır	Bağlanmaya hazır bilgisayarları modeller.
Bağlantı	Bağlantı durumunu modeller.
Kullanılmamış IP Havuzu	Kullanılmamış IP'leri modeller.
Geçişler	Anlamları
IP Atanması	IP atanma olayını modeller
Bağlantı Kesilmesi	Bağlantı Kesilmesi olayını modeller

Farklı ateşleme anlarındaki durumlar :



Şekil 1.22:İnternete Bağlanma Modeli 1.Örnek Ateşleme Sonucu.

Bağlanmaya hazır yerinden ilgili ok üzerinde tanımlı b değişkeni aracılığıyla 2.Bilgisayar **IP Atanması** geçişinin ateşlenmesi ile **Kullanılmamış IP Havuzu** yerinden ilgili ok üzerinde tanımlanmış $IP(b)$ fonksiyonu gereği IP olarak **Bağlantı** yerinde IP atanarak 2.Bilgisayar görünerek bağlantı gerçekleşmesi modellenmektedir. Sonraki ateşlemede ise **Bağlantı Kesilmesi** geçişi aktif olarak **Bağlanmaya hazır** yerinde 2.Bilgisayar görünmekte ve **Kullanılmamış IP Havuzu** yerine atanan IP tekrar yerleşmektedir.



Şekil 1.23: İnternete Bağlanma Modeli 2. Örnek Ateşleme Sonucu.

2. Örnek İşaretlemede Bağlanmaya hazır yerindeki 1. ve 5. Bilgisayarlara IP atanması geçişinin ateşlenmesiyle Kullanılmamış IP Havuzu yerindeki aynı numaralı IP'ler $IP(b)$ fonksiyonu gereği atanarak Bağlantı yerinde 1. ve 5. Bilgisayarlar görünmektedir.

1.5.2 Bileşik renk kümeleri

Bileşik renk kümeleri basit renk kümelerinin bir kombinasyonunu oluşturmaktadır. CPN ML şu şekilde birleşik renk kümeleri sağlar :products(Ürünler),records(kayıtlar), unions(birleşimler), lists(listeler), subsets and aliases (alt kümeleri ve takma adları),lists (Listeler) ve unions(birleşimler) . Ürünler ve kayıtlar bileşenlerin renk kümelerinin kartezyen ürünleri tarafından şekillenen veri kafilesini gösterir. Pascal programlama dili veya C dilindeki yapılarıdaki kayıt veri (data) türü ile yakın bir benzerlik vardır.

Product(Ürün) renk kümesi belirtilmesinin sözdizimi :

colset name (adı) = product (ürün) name1 * name2 * ... * namen;

Bu renk kümesinin değerlerinin formu :

(v1, v2, ..., vn) burada v_i için $1 \leq i \leq n$ için name i (ad i) tipi vardır.

Bir ürünün i' inci ögesini ayıklamak için aşağıdaki işlem kullanılır:

#i name (ad)

Record(Kayıt) renk kümesi bildiriminin sözdizimi:

```
colset name (ad) = record id1: name1 * id2:name2 * ... *  
idn:namen;
```

Bu renk kümesinin değerlerinin formu:

{id1=v1, id2=v2, ..., idn=vn} burada v_i $1 \leq i \leq n$ için namei tipinde değerlerdir.

Bir alias(takma-diğer ad)renk kümesi tam olarak aynı değerleri ve özellikleri önceden bildirilen renk kümesi olarak vardır. Renk kümesinin farklı bir adını kullanmak için tanıtılmıştır. Bildirimin sözdizimi:

```
colset name = name0;
```

1.5.3 Değişkenler ve sabitler bildirgesi

Bir değişken modelin yürütülmesi sırasında değeri değiştirilebilir bir tanımlayıcıdır. Değişkenler, Petri Ağ elemanlarının yazıtlarında kullanılır. Değişkenin bildiriminin sözdizimi:

```
var id1, id2, ..., idn : colset_name;
```

Burada idi tanımlayıcıdır , colset_name önceden tanımlanmış renk kümesi adıdır.

Bir değer (Value) bildirimi bir tanımlayıcıya(identifier) (ki daha sonra sabit olarak çalışır) bir değeri bağlar. Değer bildiriminin(deklerasyon) sözdizimi:

```
val id = exp;
```

Burada id bir tanımlayıcı ve exp bir CPN ML ifadesidir. İfade tanımlayıcısıyla ilişkili değeri gösterir. Örneğin: val n=5;

1.5.4 Fonksiyonlar

CPN ML fonksiyonları (işlevleri) "if" ve "case" operatörleri gibi bir programlama dilinin standart kontrol yapılarını uygularlar. Ama ML özde bir işlevsel programlama dili oluştururken bunun en etkilişi özyinelemeli(tekrarlı) fonksiyonlar ile ortaya çıkar.

fonksiyonun bildiriminin sözdizimi:

```
fun id pat1 = exp1  
| id pat2 = exp2  
| ...  
| id patn = expn;
```

Burada pat1, pat2,..., patn şablonlarıdır(model) ve exp1, exp2,..., expn hepsinin aynı tipi vardır. Bildirim gerçek argümanların pat i şablonunu yerine getirdiği(karşıladığı) durumda daha sonra fonksiyonun değeri exp i olarak hesaplandığı anlamına gelir.

If-then-else ve case kontrol yapıları işlevlerin(fonksiyonlar) tanımlaması için uygundur:

```
if bool-exp then exp1 else exp2;
```

Burada exp1 ve exp2 aynı türdür.

```
case exp of
```

```
pat1 => exp1
```

```
| pat2 => exp2
```

```
| ...
```

```
| patn => expn;
```

Burada exp1, exp2,..., expn tümünün aynı türü vardır. Anlamları diğer programlama dillerine benzer olarak olağandır. Örneğin, bir sayının işaretini hesaplayan fonksiyonu yazılabilir:

```
fun sign (x) = if x>0 then 1 else if x<0 then ~1 else 0;
```

Sayının adını döndüren(tekrarlayan) bir işlev aşağıdaki şekilde yazılabilir:

```
fun nname (x) =
```

```
  case sign(x) of
```

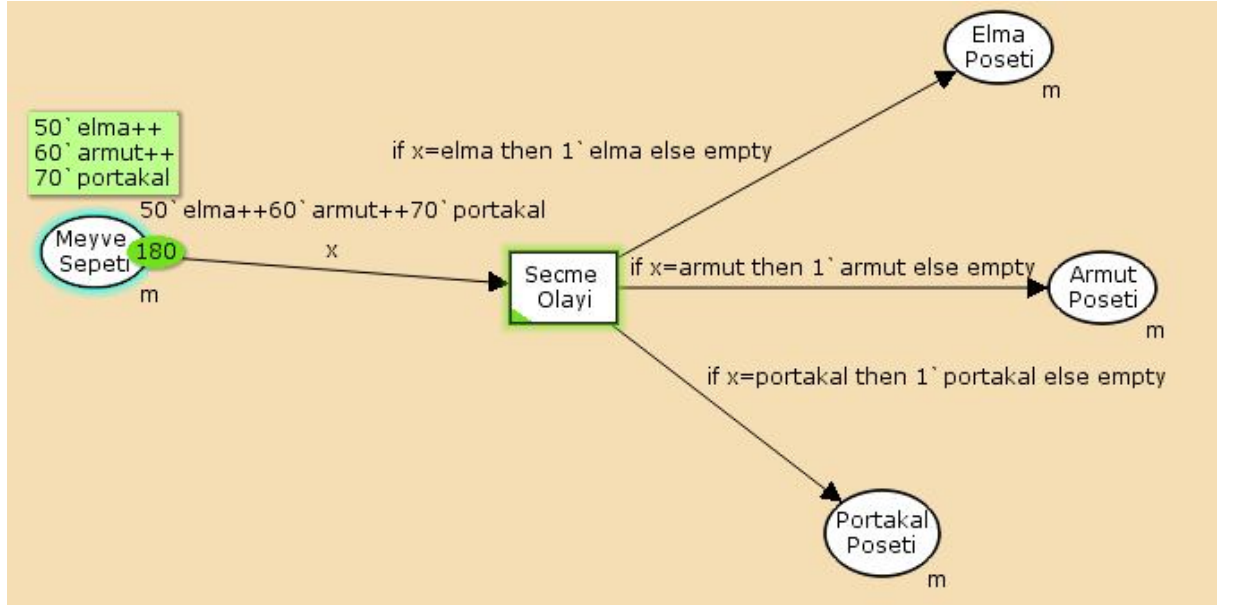
```
    1 => "positive"    (pozitif)
```

```
  | ~1 => "negative"  (negatif)
```

```
  | _ => "zero";      (sıfır)
```

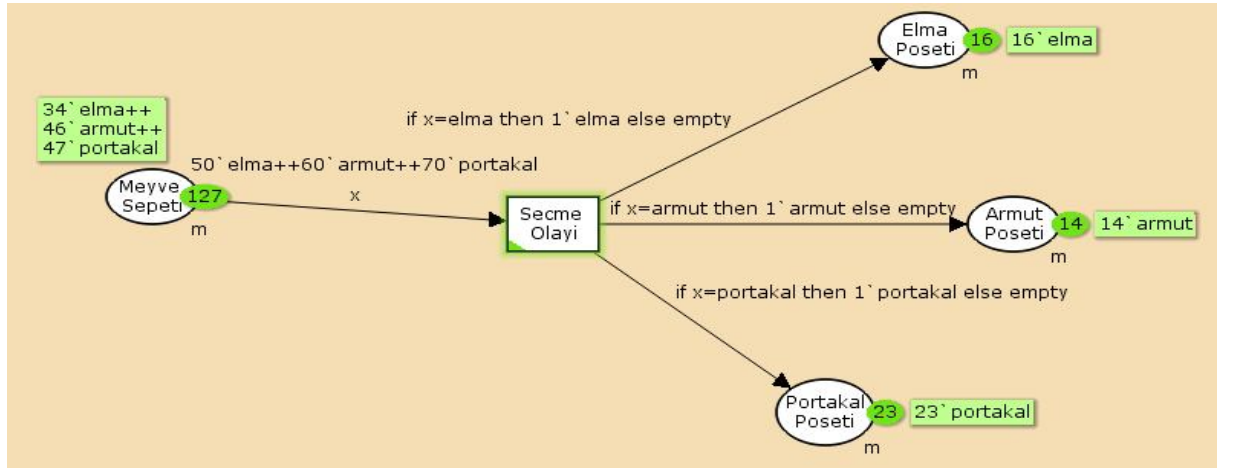
Son satırında işaretli alt çizgi _ "Sıfır" dizgesinin(string) denetlenmiş sign(x) ifadesinin tüm diğer değerleri için seçilecek olduğunu gösterir.

Burada manavın meyve seçmesi ile ilgili aşağıdaki örnek verilebilir.



Şekil 1.24: if Yapısına Ait Uygulama Örneği.

Seçme olayı geçişi ateşlendikten sonra Meyve Sepeti yerindeki karışık meyve jetonları(colset m=with elma|armut|portakal; ile tanımlı) okların yazıtlarındaki if'li ifadeler gereğince elma poşeti,armut poşeti,portakal poşeti yerlerine ayrıştırılarak görünmektedir.



Şekil 1.25:Seçme Olayı Geçişinin Ateşlendiği Örnek Bir Görünüm.

1.5.5 Rastgele fonksiyonlar

Rastgele fonksiyonlar istatistiksel özellikleri modelleme için İmkanlar sağlar. Örneğin, telekomünikasyon sistemlerinde trafik yoğunluğu veya hataların frekans tanımlamasını sağlarlar. CPN Tools'da rastgele seçim tanımlanması için birkaç yol vardır:

- Serbest değişkenler;
- Fonksiyon ran;
- Özel rastgele dağılım fonksiyonu;

Serbest değişkenler girdi ok veya korumada bağlı olmayan çıkış ok değişkenleridir. Bir CPN yürütülürken rastgele atanan değerlerdir. Serbest değişken türü küçük renk kümeleri olmalıdır. Renk kümeleri, büyük veya küçük olarak sınıflandırılabilir. Bu ayırım, önceden tanımlı hangi işlevlerin (fonksiyon) belirli renk kümesi için anlamlı olduğunu belirler. Bir renk kümesi numaralandırmak için çok fazla (varsayılan 100) öğeleri içeriyorsa büyüktür, aksi takdirde küçüktür. Birim renk kümeleri, Boolean(mantıksal) renk kümeleri, dizin renk kümeleri ve numaralandırılmış renk kümeleri küçüktür.

Ran fonksiyonu büyük renk kümeleri için rastgele bir değer üretir.

CPN Tools Bernoulli, Binom(iki terimli), Erlang, Üstel,Poisson, Üniiform(düzgün-tek biçimli) şeklinde iyi bilinen dağılımlar için bir dizi özel rastgele dağılım fonksiyonu da sağlar.

1.5.6 Multi-sets

Multi-sets CPN Tools'da yerlerin işaretleme gösterimi ve diğer amaçlar için yaygın olarak kullanılır. Her zamanki sete rağmen belirli bir çoklukla diğer bir deyişle kesin kopya sayısında her öğeyi içerir.

Multi-set kurucu geri-tırnak (') işlecidir. Örneğin, 3'5 multi-set renk 5 in üç görünümlü multi-setidir. Bir multi-set tanımlamasının sözdizimi:

i'c

Tamsayı i Negatif olmayan olması gerekir. Bu durumda değilse boş multi-set geri gelecektir. Multi-set operatörü , multi-set ek (++) ve çıkarma (--) ile birlikte multi-sets belirtmek için özlü bir yöntem sağlar. Örneğin

50`elma++ 60`armut++ 70`portakal

Bu karışımın 50 elma,60 armut,70 portakal meyvesi içerdiği anlamına gelir. Multi-set yapılandırıcının işareti geri-tırnak tır ama kesme işareti değildir.

Multisets için aşağıdaki sabitler, operasyonlar ve işlevler kullanılabilir:

empty Boş sabiti multisetlerin tüm türleri için özdeş olan boş bir multi-set oluşturur.

<code>ms1 == ms2</code>	multi-set eşitliği
<code>ms1 <><> ms2</code>	multi-set eşitsizliği
<code>ms1 >> ms2</code>	multi-set daha büyük olma durumu
<code>ms1 >>== ms2</code>	multi-set daha büyük ya da eşit olma
<code>ms1 << ms2</code>	multi-set daha az
<code>ms1 <<== ms2</code>	multi-set daha az ya da eşit
<code>ms1 ++ ms2</code>	multi-set toplama
<code>ms1 -- ms2</code>	Multi-Set çıkarma (ms2 ms1 eşit veya daha az olmalıdır), ms2
<code>ms1 'den daha</code>	az veya eşit değilse, özel durum (istisna) çıkarma durumunu yükseltir.
<code>i ** ms</code>	skaler çarpım
<code>size ms</code>	multi-set ms büyüklüğü
<code>random ms</code>	msden rasgele bir sözde(sahte) renk döndürür(geri verir)
<code>cf(c,ms)</code>	ms'de c renginin görünümünün sayısını döndürür.
<code>filter p ms</code>	bir p yüklemine ve bir ms multi-setini alır ve yüklemi karşılayan(sağlayan) MS'in tüm görünümünün multi-setini üretir.

CPN Tools'da ilk ve geçerli bir yer işaretlemesi yerin multiset renk kümesi ile simgelenir. Yerin çıkış oku yazıtındaki bir değişken aracılığıyla bir jeton seçiminde CPN tools rastgele(random) fonksiyonla benzer şekilde bir rastgele seçim sağlar.

1.5.7 Zamanlı Multi-setler

Zamanlanmış multi-setler CPN Tools'nda modelindeki süreli gecikmeleri temsil etmek için kullanılır. Karşılık gelen renk kümesinin bildirgesi zamanlanmış(*timed*)değiştirici ile başarılmış olmalıdır. @, @+ ve @@+ operatörleri, zaman damgalarını(pullarını) renklere eklemek için kullanılır. Bir c rengine bir x zaman gecikmesini (time delay) eklemek bir c rengi için geçerli model zamanı +x 'e eşit olan bir değerle bir zaman damgasını(pulunu) iliştiirecektir. Aşağıdaki işlemler zamanlanmış multi-setler için geçerlidir:

<code>c @ t</code>	zaman damgası(pulu) t yi (zaman) c rengine iliştirir.
<code>ms @+i</code>	ms multi-setindeki her bir renge tamsayı zaman gecikmesi i 'yi ekler, zamanlanmış bir multisete geri döner
<code>tms1 +++ tms2</code>	zamanlanmış multi-set ekleme

Örneğin bildirim:

```
colset tint = int timed;
```

var t : tint;

t = 1`2@100 ++ 1`3@200 ++ 1`4@300;

2@100 jetonu yalnızca model zaman örneği 100'e eşitlendikten sonra , 3@200 jetonu - yalnızca model zaman örneği 200'e eşitlendikten sonra CPN Tools tarafından alınmış olabileceği anlamına gelir. Etkinleştirme(aktivasyon) saatinden önce bir jeton bir modelin herhangi bir geçişi tarafından alınmış olamaz.

1.6 Modellerin Dilinin Tanımlanması

CPN tools'da Petri Ağı'nın her elemanının CPN ML dilinde tanımlanan öznitelikleri vardır. Biz Create paletini kullanarak modelinin bir sayfası üzerinde bir eleman yerleştiririz.Sonra elemanın öznitelikleri eklenmelidir. Bu amaçla fareyle karşılık gelen öge tıklanmalı ve nitelikleri arasında geçiş(anahtarlamak) için Tab tuşu kullanılmalıdır. ESC tuşuna basmanız seçili öğeden çıkmanızı sağlar; modelin diğer noktasında fareyi tıklatarak aynı sonuç alınabilir. Her ögesinin özniteliklerini ayrı olarak göz önüne alalım.

1.6.1 Yer yazıtları

Bir yer ile ilişkili olabilecek üç yazıt vardır. İkiisi isteğe bağlıdır ve biri gereklidir:

- Renk kümesi yazıtı - gerekli
- ilk işaretleme yazıtı - isteğe bağlı
- yer adı yazıtı - isteğe bağlı

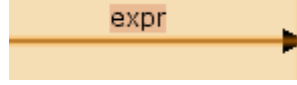


Şekil 1.26:Yer Ayrıntıları.

İlk(başlangıç) işaretlemede CPN Tools otomatik olarak yeşil renkte yazılan ve jetonların toplam sayısını ve işaretlemenin ayrıntıları gösteren geçerli işaretlemeyi oluşturur.

1.6.2 Ok yazıtları

Yazıtın,aşağıdaki formu vardır:



Şekil 1.27: Ok Yazıtı Gösterimi.

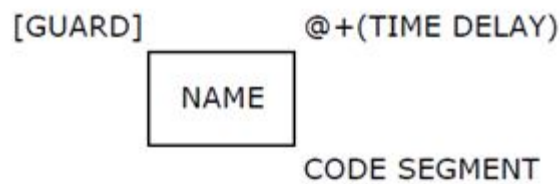
Ok `expr` ifadesinin renk kümesi oka bağlı yer renk kümesiyle eşleşmelidir. Eğer bir ok ifadesinin renk kümesi, oka bağlı yerin renk kümesiyle eşleşmiyorsa , sözdizimi denetimi sırasında okun yanında bir hata iletisi görünecektir. Bir geçişin giriş ve çıkış oklarının yazıtları arasında temel fark vardır.

Geçişin giriş ok ifadesi jetonun seçimi için bir örnek(şablon) meydana getirmektedir. Bu şablon karşılık gelen renk kümesi için filtre fonksiyonunda uygulanabilen bir yüklem tarafından tanımlanır. En basit durumda bu yüklem yukarıdaki örneklerdeki gibi ilgili renk kümesinin tek bir değişkeninden oluşur.

1.6.3 Geçiş yazıtları

Bir geçiş ile ilişkili olabilecek dört yazıt vardır. Tümü isteğe bağlıdır:

- Geçiş adı yazıtı
- koruma(bekçi) yazıtı (guard)
- Zaman yazıtı
- Kod segmenti(bölümü-kesiti) yazıtı



Şekil 1.28:Geçiş Ayrıntıları.

Geçiş gecikmesi bir pozitif tamsayı ifadesi olmalıdır. İfade `@+` ile önce gelir ve bu zaman yazıtının `@+delay-expr` formu olduğu anlamına gelir. Bir zaman yazıtı eklenmeden önce yazıt için varsayılan metin `@+` olur. Gecikme süresi her zaman geçerli saate göre eklenir. Örneğin, geçerli saati 10 ve gecikme süresi `@+2`, ardından çıkış yerlerine gönderilen jetonların zaman damgası(pulu) 12 olacaktır. Eksik(kayıp) bir zaman yazıtı bir sıfır gecikmeye eşdeğerdir. Okların gecikmesine rağmen geçişin gecikmesi tüm geçiş çıkış jetonlarına uygulanır.

Bir bekçi (guard) doğru veya yanlış olarak değerlendirilen CPN ML Boole ifadesidir. Bir bekçi eklenmeden önce yazıt için varsayılan metin [] olur. Bekçi tek bir Boolean ifadesi olabilir veya Boolean ifadeler listesi [b-exp1, b-exp2,..., b-expn] olabilir. Geçiş sadece guardı doğru olduğu durumda ateşler ve guard giriş jetonlarının seçimini kısıtlar.

Ayrıca, guard ifadelerindeki kombinasyonlarını kullanarak farklı yerlerden jetonların parametrelerinin karşılaştırılmasına imkan sağlar.

Her geçişin ml kodunu içeren bağlı kod bölümü vardır. Ana geçişler ortaya çıktığı zaman kod bölümleri yürütülür(çalıştırılır). Kod segmentleri(bölümleri) CPN değişkenlerini kullanabilir ve başka bir yere bağlı olmayan çıkış okları üzerinde bulunan CPN değişkenlerini bağlayabilir. Her kod segmenti(bölümü) aşağıdakileri içerebilir:

- Giriş şablonu(örnek) (isteğe bağlı)
- Çıkış şablonu (isteğe bağlı)
- Kod eylemi (zorunlu)

Bir giriş şablonu CPN değişkenlerinin sıralı listesidir , input(girdi) anahtar kelimesiyle önce gelen bir kalıptır. Giriş paterni(şablonu) kod eyleminde kullanılabilecek CPN değişkenlerini listeler. Kod eylemi bu CPN değişkenlerin değerlerini kullanabilir, ancak onları değiştiremez. Bildirim düğümünde aynı adla bir ML tanımlayıcı bildirilse bile giriş paterninde(şablon) listelenen CPN değişkenleri kod eyleminde kullanılabilir. Giriş yan tümcesi atlanırsa, kod eyleminde CPN değişkenlerinin hiçbirinin kullanılmayacağı anlamına gelir.

Bir çıkış paterni (şablonu) output (çıkış) anahtar kelimesiyle önceden gelen CPN değişkenlerinin bir sıralı listesidir. Çıkış paterni (şablonu) kod eylemi çalıştırılmasının (yürütülmesinin) bir sonucu olarak değiştirilecek CPN değişkenlerini listeler. Bir çıkış paterni bir CPN değişkeni veya tekrarlama olmadan CPN değişkenlerin bir sıralı listesi olmalıdır. Output yan tümcesi atlanırsa, CPN değişkenlerinin hiçbirinin hesaplanmadığı anlamına gelir.

Bir kod eylemi action (eylem) anahtar kelimesi ile önceden gelen bir ML ifadesidir. Kod eylemi renk kümeleri, CPN değişkenleri veya referans(başvuru) değişkenlerinin herhangi bir bildirimini içerebilir. Ancak, kullanıcı tanımlı

(bildirimli) ve ön bildirimli sabitler, operasyonlar ve işlevleri uygulayabilir. Kod eylemi, giriş şablonunda belirtilen CPN değişkenlerini içeren bir ortamda yerel bir deklarasyon olarak yürütülür. Bu, kod eyleminin herhangi bir CPN değişkenlerini ama onların sadece yerel kopyalarını doğrudan değiştiremeyeceğini garanti eder. Kod eylemi yerine getirildiği zaman, sonucu çıkış şablonunda (paterninde) CPN değişkenlerini bağlamak için uygulanır. Kod eylemi, giriş paterni değişkenlerinin bulunduğu bir ortamda değerlendirildiğinde çıkış şablonundaki gibi aynı türde bir sonuç verim sağlaması gerekir. Hiçbir çıkış şablonu verilmezse, türünün birim olduğu varsayılır.

Kod segmentleri (bölümleri) giriş jetonlarının daha karmaşık süreci için kullanılır (Url 2).

2. SMTP (BASİT POSTA AKTARIM PROTOKOLU)

Günümüzde yaygın olarak kullanılan e-posta, sistem olarak postaları internet üzerinden SMTP (Basit Posta Aktarım Protokolü) kullanarak sunucudan sunucuya iletir. Bu mesajlar e-posta kullanıcısı tarafından ya POP(post office protocol) ya da IMAP (internet message access protocol) kullanılarak alınır. SMTP aynı zamanda genellikle e-posta istemcisi tarafından, POP ve IMAP kullanmadan basit mbox özelliğinin aynı sistem üzerinde olabileceği “host bazlı” posta sistemlerine mesaj göndermek amacıyla da kullanılır.

2.1 SMTP Temelleri

SMTP, e-postaları farklı hostlar arasında TCP/IP çatısı altında iletmek amaçlı kullanılan yaygın mekanizmadır. Uygulama katmanlı protokoldür. SMTP altında istemci(client) SMTP süreci, uzak anasistem(host) üzerindeki sunucu SMTP sürecine bir TCP bağlantısı açar ve bu bağlantı ile e-posta gönderme işlemini gerçekleştirir. SMTP sunucusu TCP bağlantısını spesifik olarak port (25)’ten dinler, istemci(client) SMTP süreci ise bu porttan bağlantıyı başlatır. TCP bağlantısı başarıyla gerçekleştikten sonra bu iki süreç SMTP tarafından tanımlanan basit bir talep-cevap diyalogu başlatır. İstemci(client) SMTP sürecinin ilettiği verici ve alıcı e-posta adresleri, sunucu süreci tarafından kabul edildiğinde, istemci süreci e-posta anlık mesajını iletir.

POP3 ya da IMAP kullanıcılara, yerel sunucunun posta kutusuna gelen e-postaları indirip kaydetmeyi sağlar.

2.2 SMTP Model Ve Protokol

SMTP modeli hem end-to-end hem de store-and-forward e-posta teslim metotlarını destekler. End-to-end metodu organizasyonlar arasında kullanılırken, store-and-forward metodu ise TCP/IP ve SMTP bazlı ağlarda organizasyonların çalıştırılmasında kullanılır.

E-posta başarıyla alıcıya iletilene kadar e-posta parçası istemci tarafında da saklanır. Diğer e-posta sistemlerinden farklı olarak ileti aktarımı, özgün SMTP göndericisi ile nihai SMTP alıcısı arasındaki tek bir bağlantıda oluşabileceği gibi bir sürü ara bağlantı üzerinden atlaya atlaya da gerçekleşebilir.

RFC 821 standartı istemci-sunucu protokolünü tanımlar. SMTP komutları SMTP istemcisi tarafından üretilir ve SMTP sunucusuna gönderilir. SMTP yanıtları ise SMTP sunucusundan SMTP istemcisine komutlara yanıt olarak gönderilir.

Bir SMTP sunucusu ya bir nihai hedef ya bir ara "röle" (iletiyi aldıktan sonra aktaran bir SMTP istemcisinin rölesi olarak da kabul edilebilir) ya da bir "ağ geçidi" olabilir (iletiyi SMTP dışında bir protokol kullanarak aktaran anlamında ele alınabilir).

E posta sistemi terminolojisi RFC 821 yayınlandıktan sonra yaygın duruma geldi ve uygun düştükçe de bu belirtimde kullanıldı. Kısmen, SMTP sunucuları ve istemcileri bir posta aktarım hizmeti sağlamaları nedeniyle "Posta Aktarım Aracıları" (MTA – Mail Transfer Agent) gibi davranırlar.

SMTP istemcisi ve SMTP sunucusunun iki temel bileşeni vardır: UA ve MTA. Ağlar arasında e-posta gönderimi konusunda birkaç durum örneği vardır.

Gönderenin UA'ı mesajı hazırlar, zarfı yaratır ve mesajı bu zarfa yerleştirir. MTA e-postayı ağ üzerinde alıcı MTA'nın port 25'ine denk gelecek şekilde iletir.

İkinci durumda ise sunucu ile istemci arasındaki e-posta iletimi esnasında rölenin devreye girmesi mevcuttur.

Üçüncü senaryoda ise gönderen ve alıcı arasındaki iletişim SMTP protokolü dışındaki protokoller ile hazırlanmış e-postaları alabilen (sonradan bu formata dönüştürülür) bir MTA rölesi olan e-posta ağ geçidi ile sağlanır.

Bazı uygulamalarda ise TCP/IP SMTP e-posta sistemi ile yerel e-posta sistemleri arasında dönüşüm gerçekleştirilebilir.

2.3 User Agent(Kullanıcı Ajanı) (UA)

UA e-posta gönderimi ve alımı için kullanılan bir programdır. RFC 876da SMTP implementasyonu ile ilgili bilgiler bulunmaktadır.

2.3.1 E-posta gönderimi

E-posta gönderimi, istemci-sunucu talep-cevap işlemlerinin seriler halinde gerçekleşmesi sonucu gerçekleşir.

SMTP iletişim işlemleri, zarf ve mesajın birleşimiyle oluşur. Bu birliktelik başlık (gönderen ve alıcı bilgileri) ve body kısmının (mesajı içeren text kısmı) birleşimi ile oluşur. Zarf MAIL FROM ve RCPT TO komutlarıyla mesajdan ayrı olarak iletilir.

2.3.2 E-posta başlığı formatı

E-posta başlığı gönderim tarihi, gönderenin adresi, e-postanın nereye gideceği... gibi bazı bilgileri anahtar kelimeler halinde içerir. Kullanılan bazı önemli anahtar kelimeler (ingilizce olarak): “to, cc, from, reply-to, return-path, subject”

2.3.3 E-posta alımı

UA periyodik olarak posta kutusu içeriğini kontrol eder. Yeni e-posta alındığı zaman özel olarak kullanıcıyı uyarır. Kullanıcı e-postayı okumak istediği zaman, gelen e-posta paketleri listesi kullanıcıya gösterilir. Listedeki her satır gelen e-posta içeriği ile ilgili özet bilgi içerir. Bu özetler aynı zamanda e-posta başlığı anahtar kelimelerinden bazıları (from gibi) içerir.

2.3.4 Ertelenmiş iletim

SMTP protokolü e-postaların ertelenmiş şekilde gönderilmesine ve/veya ulaşmasına olanak sağlar.

Gönderenin e-posta gönderimini erteleyebilmesi için e-postaları biriktiren ve saklayan bir sistemi (spool) bulundurması gerekmektedir. UA mesajı oluşturduktan sonra, mesaj biriktirme işlemini yapan sisteme (spool) aktarılır. E-posta, alıcının hazır olduğu zaman, sunucunun IP adresi DNS üzerinden alındığında gönderilir. Mesaj yaklaşık 3-5 gün arasında iletilmediğinde gönderene geri döner. Bu süre zarfında MTA periyodik olarak (10-30 dk gibi bir süre) spoolda bekleyen hangi e-postaların gönderildiğini kontrol eder.

E-posta alımı için sunucu MTA’ı, mesajı alıcının posta kutusunda saklar. Bu durumda, alıcı posta kutusuna uygun olduğunda erişip e-postayı alır.

Ayrıca SMTP standart prosedürleri, ortadaki MTA’lerin istemci ve sunucu gibi hareket edip e-posta erteleme sürecinde görev üstlenmelerine olanak sağlar.

2.3.5 Takma adlar

SMTP mekanizması takma ad tanımlanarak tek bir isimle birden fazla e-posta adresinin temsil edilmesine olanak sağlar. Sistem bunu bir takma ad veri bankası oluşturarak gerçekleştirir.

2.4 MTA(Mail Transfer Agent)

MTA gerçek e-postaları transfer eder. Sistem e-posta göndermek için istemci MTA'sına ve e-posta alabilmek için sunucu MTA'sına sahip olmalıdır. SMTP'ye bağlı RFCler spesifik MTA tanımlamazlar. Windows'un kullandığı en yaygın MTA, Microsoft Exchange 2000/2003'tür.

“mta-name-type” ve “address-type” parameterleri SMTP gönderim durum raporunda kullanılması için tanımlanır. Diğer e-posta sistemlerinin kimlik tanıma sistemleri de kullanılabilir.

2.4.1 SMTP e-posta alışverişi akışı

SMTP protokolü (RFC 821) MTA'lara komut ve cevapların nasıl iletilmesi gerektiğini tanımlar. İstemci komutu sunucuya gönderir ve sunucu sayısal kodlar ve insanların okuyabileceği opsiyonel diziler ile karşılık verir. İstemci tarafından sunucuya neredeyse bir düzineden az olacak kadar komut gönderilebilir.

1.İstemci hedef SMTP ile TCP bağlantısı kurar ve sunucunun başarılı bağlantı için 220 Service ready mesajı ya da başarısız bağlantı için 421 Service not available mesajı göndermesini bekler.

2.HELO komutu gönderilir ve alıcının kendisini, domain ismini göndererek, tanıtması zorlanır. İstemci bu bilgiyi doğru SMTP ile bağlantı kurduğunu teyit etmek amacıyla kullanır. Eğer gönderen-SMTP, SMTP servis uzantılarını RFC 1651de belirtildiği gibi destekliyorsa; EHLO komutu HELO komutunun yerini alır. Servis uzantılarını desteklemeyen alıcı-SMTP, 500 Syntax error, command unrecognized mesajı ile geri dönüş sağlar. İstemci, bu durumda HELO komutunu yeniden denemeli veya mesaj servis uzantılarıyla iletilmiyorsa QUIT mesajını göndermelidir. Eğer alıcı SMTP servis uzantılarını destekliyorsa, bu uzantı listesini içeren çoklusatırlı 250 OK mesajları ile cevap vermelidir.

3. İstemci, artık e-posta alışverişini MAIL komutunu alıcıya göndererek başlatabilir. Bu komut hata raporu gönderimi sağlayan ters yönlü yol içerir. Ek olarak yol gösteren hostların listesini de içerebilir.

Eğer e-posta, e-posta köprüsünü geçerse ve kabul edilirse, sunucu(alıcı) 250 OK mesajı ile cevap atar.

4. Gerçek e-posta alışverişine ait ikinci adım, sunucu SMTP'sine mesajın hedefini sağlamaktır. Bu işlem RCPT TO:<forwardpath> komutunu bir ya da daha fazla göndererek gerçekleştirilir. Hedef bilindiği takdirde sunucu 250 OK cevabını alır aksi takdirde ise 550 No such user here cevabı ulaşır.

5. Tüm RCPT komutları gönderildiğinde istemci, sunucuyu mesajın içeriğinin gönderileceği konusunda bilgilendirmek amacıyla DATA komutunu yayınlar. Sunucu bunun üzerine 354 Start mail input, end with <CRLF>.<CRLF> mesajıyla cevaplar.

6. İstemci, artık verileri seriyi <CRLF>.<CRLF> satırıyla sonlandırarak şekilde satır satır gönderir. Bunu da alıcı 250 OK ile anlamlandırır. Bir sorun var ise de hata mesajı gönderir.

7. Bundan sonraki adımlar aşağıdaki gibi gerçekleşebilir:

- Gönderenin göndereceği mesaj kalmadığından bağlantıyı QUIT komutu ile koparır. Cevap olarak da 221 Service closing transmission channel mesajı döner.

- İstemcinin göndermesi gereken mesajlar kalmıştır ve bunun için 3.adıma döner ve yeni bir MAIL komutu gönderir.

Yukarıda sadece en önemli temel komutlar verilmiştir.

2.4.2 SMTP komutları

Komutlar istemciden sunucuya ASCII (text) formunda gönderilir. Komutların basit yapısı tüm platformlarda e-posta istemci ve sunucularının kurulmasına izin verir.

Bir komut, sıfır veya daha fazla argümanla takip edilen anahtar kelimedenden oluşur.

Tüm implementasyonların zorunlu olarak içerdiği beş komut (HELO, MAIL FROM, RCPT TO, DATA, ve QUIT) şeklindedir. Diğer 3 komut (RSET, VRFY ve NOOP) sıklıkla kullanılır. Sonraki 6 program ise (TURN, EXPN, HELP, SEND FROM, SOML FROM, ve SAML FROM) nadiren kullanılır.

2.4.3 E-posta servisi çeşitleri

E-posta sunucularından istenen servis setleri bazen “hello” anahtar kelimesi ile karakterize edilir. Çeşitli e-posta servisleri:

HELO for Simple Mail (Basit E-posta) (RFC821)

EHLO for Mail Service Extensions (E-posta Servisi Uzantısı) (RFC1869)

LHLO for Local Mail (Yerel E-posta) (RFC2033).

2.4.4 SMTP yanıtları

Cevaplar sunucudan istemciye gönderilir. Cevap 3 digitli kod şeklindedir. Bazen ek olarak text formunda bilgi de eklenebilir. İlk digitin anlamı:

2bc—pozitif tamamlama cevabı: istenen komut başarıyla gerçekleştirilmiş ve yeni komut başlatılabilir.

3bc— pozitif ara cevap: istenen komut başarıyla kabul edilmiş ama sunucu tamamlama işlemini gerçekleştirmek için daha fazla bilgiye ihtiyaç duyuyor.

4ab—geçici negatif tamamlama cevabı: istenen komut reddedilmiş ama hata durumu geçici ve komut tekrar gönderilebilir.

5ab—kalıcı negatif tamamlama cevabı: istenen komut reddedilmiş ve komut tekrar gönderilemez.

2.5 SMTP Sunucu

SMTP sunucusu diğer internet anasistemlerinden SMTP’yi kullanarak e-posta gönderip alır. SMTP sunucusu işlemleri hem içeri hem dışarı yönlüdür. Dışarı yönlü e-postalar SMTP sunucusu hedefe ulaştığını onaylayana kadar biriktirilir.(Spool) İçeri yönlüler ise kullanıcı POP3 veya IMAP4 e-posta istemcisi ile e-postaya ulaşana kadar spoolda bekletilir. Bu spooling işlemi istemciden sunucuya transferi arka planda yapmaya olanak sağlar.

2.6 On-Demand Mail Relay(Talep Üzerine E-posta Rölesi)

Doğrulanmış TURN (ATRN) olarak da bilinen On-demand mail relay (Talep üzerine e-posta rölesi) (ODMR), kullanıcıların herhangi bir internet bağlantısından; internet servis sağlayıcısına (ISP) bağlanmasını, doğrulamayı geçmesini ve dinamik IP adresi kullanarak e-posta talep etmesini sağlayan e-posta servisi.

İlk istemci ve sunucunun rolü kısa sürelidir; çünkü servis sağlayıcı tarafından talep e-postayı göndermek amaçlı sunucuya aralıklı olarak bağlanma izni verilir. Kullanıcı sağlayıcıyla olan bağlantıyı başlatır, doğrular ve e-postayı talep eder. İstemci ve sunucunun rolü tersine çevrilir ve normal SMTP senaryoları işler. Sağlayıcı ODMR port 366'daki bağlantıları dinlemek için ODMR prosesine sahiptir.

Sunucuda, bu işlemler EHLO, AUTH, ATRN, ve QUIT komutlarını implemente eder. Ayrıca dışarı yönlü e-postalara ulaşan SMTP istemcisi olmalıdır. MTA normalde; çıkan e-postaları işleyen, belirli domainler için e-posta gönderimini deneyen, zamana ya da olaya bağlı e-posta istemci bileşeni içerir.

2.7 MIME(Multipurpose Internet Mail Extensions)

RFC 821/ STD 10 standartı yüksek anlamlı bit sıfırlanarak 7 bit ASCII verisi olan SMTP ile data gönderimini tanımlar. Bu durum konuşma dili mesajlarının iletiminde çoğu örnekte yeterlidir. Ama farklı dillerdeki text ve text olmayanlar için yeterli değildir.

Bu durumun üstünden gelmek için iki yaklaşım mevcuttur. 1.si MIME'nin text ve binary datanın çözülmesini sağlayan mekanizmayı belirten RFC 1521 ve 1522 içinde tanımlanmasıdır.

2. yaklaşımda, SMTP servis uzantıları (RFC 1651, RFC 1652 ve RFC 1653), RFC 821'in getirdiği sınırlamaları aşmak amaçlı SMTP'nin kapasitesini arttıran mekanizmayı tanımlar. RFC 1651 dökümanı alıcı SMTP'sini bilgilendirmek amaçlı dökümandır. Yine bazı spesifik uzantılar ise RFC 1652 ve RFC 1653de tanımlanır. SMTP ve MIME servis uzantılar birbirini tamamlayıcı özelliktedir.

SMTP servis uzantılarının maksimum satır uzunluğu sınırlaması vardır. (RFC 821e göre 1000 karakterdir.) Uzantılar ayrıca e-posta başlığında ASCII olmayan karakter kullanımına da sınır getirir.

MIME protokolü orjinal başlığa eklenebilen 5 başlık tanımlar: "MIME-version, contenttype,content-transfer-encoding, content-id, and contentdescription."

2.7.1 MIME-version

Header satırı MIME-Version: 1.1, mesajın MIME protokolünün versiyon 1.1'i kullanılarak oluşturulduğunu ifade etmektedir.

2.7.2 Content-type

Header satırı Content-Type:<type/subtype; parameters> , mesajın ana kısmında kullanılan data tipini ifade eder. İçerik tipinin ve içerik alt tipinin tanıtıcıları bölme işareti ile ayrılır. Alt tipe bağlı olarak, header farklı parametreler içerebilir.

2.7.3 Content-transfer-encoding

Content-Transfer-Encoding:<type> header satırı, taşıma sırasında mesajları 0'lar ve 1'lerden oluşan bit-stream'lere kodlamak için kullanılan metodu tanımlar. 5 tip kodlama çeşidi aşağıdaki gibidir:

- 7 bit-NVT ASCII karakterleri ve 1000 karakterden daha kısa satırlar için
- 8 bit-ASCII olmayan karakterler ve 1000 karakterden daha kısa satırlar için; arka plandaki SMTP protokolü, 8-bit ASCII olmayan karakterleri iletebiliyor olmalı (bu tip tavsiye edilmez.)
- İkili-limitsiz uzunluktaki satırlara sahip ASCII olmayan karakterler için; 8-bit kodlamadır. Arka plandaki SMTP protokolü, 8-bit ASCII olmayan karakterleri iletebiliyor olmalı (bu tip tavsiye edilmez.)
- base64-en yüksek bit'in 0 olma zorunluluğu olmadığı durumlarda, baytlardan oluşan datanın gönderilmesi için; 6-bit blok datası 8-bit yazdırılabilir ASCII karakterlerine kodlanır , böylece altta yatan e-posta transfer mekanizmaları tarafından desteklenen herhangi bir karakter seti olarak gönderilebilirler.
- quoted-printable-çoğunlukla ASCII karakterlerinden oluşan, ASCII olmayan karakterlerin oranı az olan datayı göndermek için; eğer karakter ASCII değilse 3 karakter olarak gönderilir: ilk karakter eşittir (=) işareti, sonraki iki karakter ise baytın onaltılı gösterimidir.

İçerik tipi ve kodlama birbirinden bağımsız olmasına rağmen, RFC 1521 dökümanı, ASCII olmayan data içeren metin için quoted-printable; şekil, ses, görüntü ve sekizli

dize uygulama datası için base64 kullanılmasını önerir. Bu yöntemle, RFC 821 ile en yüksek seviyede birlikte çalışabilirlik sağlanır .

2.7.4 Content-Id

Content-Id: id=<content-id> header satırı, çoklu mesaj ortamlarında bütün mesajı tespit eder.

2.7.5 Content-Description

Content-Description: <description> header satırı, ana metnin şekil, ses ya da görüntü olup olmadığını tanımlar.

2.7.6 MIME güvenlik şeması

S/MIME, MIME protokolü için bir güvenlik sistemidir. RSA Güvenlik tarafından geliştirilmiştir ve PGP (oldukça iyi gizlilik) şifrelemeye ve public-key cryptography kullanan dijital imza sistemine alternatiftir. S/MIME sistemi IETF tarafından standardize edilmiştir. IAB Güvenlik Mimarisi Seminer Raporuna göre (RFC 2316), MIME-kapsüllenen e-posta'ya güvenli bölümler eklemek için belirlenmiş güvenlik mekanizması, “MIME için Güvenlik çokluparçaları: Multipart/Signed ve Multipart/Encrypted (RFC 1847)” de açıklandığı üzere Security/Multipart'tır.

S/MIME, şirket içi ve şirket dışı e-postalarda e-posta güvenliğini standartlaştırma ihtiyacı duyan büyük şirketler tarafından oldukça yaygın bir şekilde kullanılmaktadır. Şirket içinde geliştirilen ya da diğer sertifikalı uzmanları kullanarak public-key altyapısı kurmayı gerektirir .

2.8 E-Posta İletim Tipleri

SMTP (RFC821) ve Standard for the Format of Advanced Research Project Agency (ARPA) Internet Text Messages (RFC822), internette taşınırken, bir dizi “alınan” satırın elektronik e-posta mesajlarının headerlarına ekleneceğini açıkça belirtir. Alınan satır opsiyonel olarak “yoluyla” ya da “ile” kalıbı içerebilir (ya da her ikisini de). “Yoluyla” kalıbını kullanmanın amacı mesajın iletildiği bağlantıyı ya da fiziksel ortamı belirtmektir (Örnek olarak, Unix-to-Unix Copy Program için UUCP bağlantı tipi belirtilmelidir). “İle” kalıbı ise mesajı iletmek için kullanılan protokol ya da mantıksal süreci belirtmek için kullanılır (SMTP ya da ESMTP parametreleri,

sırasıyla SMTP [RFC821] ya da servis uzantılı SMTP [RFC1869] protokol tipleri için kullanılır.).

2.9 E-Posta Erişim Modları

Bir e-posta mesajı , hedef varış yerine ulaşabilmesi için, e-posta sunucusu, e-posta erişim protokolü ve e-posta istemcisi tarafından işlenmelidir. İnternet e-posta sunucusu (MTA = Posta iletim aracı olarak da bilinir), e-postayı iletmek ve almaktan sorumlu bir yazılımdır. MTA yazılımı, internet erişimi olan bir bilgisayarda koşturulur ve ISP'ler ya da şirketin bilgi servisleri görevlileri tarafından yönetilir, gözlemlenir ve yedeklenir. Bazı e-posta sunucuları e-postayı yalnızca kullanıcı e-postaya erişene kadar saklarken; diğerleri kullanıcı e-postasını kalıcı olarak saklar. Bir e-posta kullanıcısı çoğunlukla e-posta sunucusuyla etkileşimde bulunmak için bir e-posta istemci programı kullanır .

E-posta istemcisi (e-posta kullanıcı aracı olarak da bilinir) kullanıcının e-postasını okumak, göndermek, dosyalamak ve proses etmek için kullandığı yazılımdır. Genellikle kullanıcının masaüstü bilgisayarında çalıştırılan e-posta istemcisi, aynı zamanda ilgili e-posta datasını da (adres defteri, yazım hatalarını düzeltme vs) yönetir. E-posta istemcisi yeni e-postaya ulaşmak için e-posta sunucusuna bağlanır. Bazı e-posta istemcileri tüm e-postaları saklamak için de e-posta sunucusunu kullanır.

E-posta istemcisi ve e-posta sunucusu arasındaki iletişim, birçok farklı tipteki network bağlantısı üzerinden gönderilen standartlaştırılmış bir dizi komut ve yanıtta oluşan e-posta erişim protokolü ile düzenlenir. Protokol komutları (sadece İnternet e-posta erişimini yönetmek için oluşturulmuştur) e-posta istemcisi ve e-posta sunucusu arasındaki iletişimin biçimini, tarzını, karakteristiğini ve yeterliliğini de etkileyebilecek bir tasarım yaklaşımına bağlıdır . SMTP protokolü, internet üzerinde gönderilen bir e-postanın fiili olarak gönderme işlemiyle ilgilenir.

Bir e-posta erişim protokolü kullanıcının e-postasını proses etme şekline göre (nerede ve nasıl) 3 genel modda çalışır:

- Offline mod-e-posta, e-posta sunucusu üzerindeki geçici bellekten kullanıcının bilgisayarına indirilir. E-posta indirildikten sonra, sunucudan silinir.

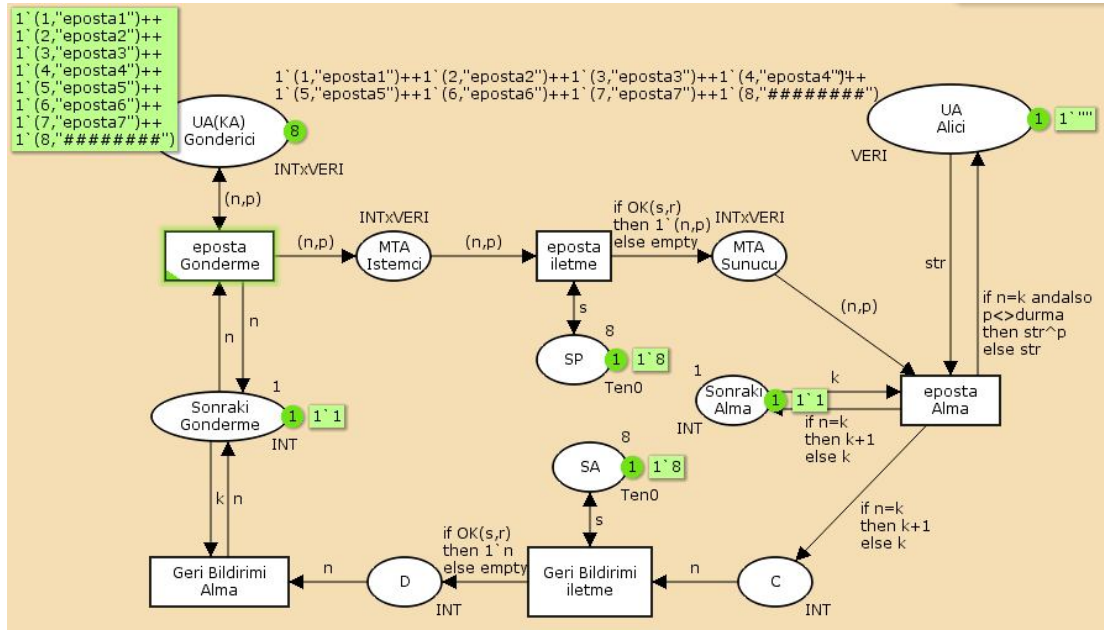
- Online mod-kullanıcının e-postası, gelen kutusu, dosyalanmış tüm e-postaları kalıcı olarak e-posta sunucusunda saklanır. Kullanıcı sunucuya bağlanarak, e-postasının geçici bir kopyasını indirir, okur ya da e-posta gönderir. Bağlantı sona erdiğinde, kopya kullanıcının bilgisayarından silinir ve sadece orijinali sunucu üzerinde kalmaya devam eder.

- Disconnected/resynchronization mod-Offline ve online modların birleşimidir. E-posta bilgisayara kaydedilir ve orijinal mesaj e-posta sunucusunda kalmaya devam eder. Kullanıcı, herhangi bir bilgisayara aldığı kopyayı değiştirebilir, daha sonra sunucudaki orijinal e-postayı ve diğer bilgisayarlardaki kopyaları da içeren tüm kopyaları senkronize eder.

Yukarıdaki 3 mod da, çoklu platform desteği sunar (Url 3).

3. SMTP CPN TOOLS MODELLERİ

Bu bölümde e-postaların geçebileceği veya kaybolabileceği bir ağ aracılığıyla e-posta dizisinin bir bölümden diğer bölüme gönderildiği protokol modellenmeye çalışılmıştır.



Şekil 3.1: SMTP Protokolünün CPN Tools Modeli.

```

▼ colset VERI=string;
▼ colset INT = int;
▼ colset BOOL = bool;
▼ colset STRING = string;
▼ var n,k:INT;
▼ var p,str:VERI;
▼ val durma="#####"
▼ colset Ten0=int with 0..10;
▼ colset Ten1=int with 1..10;
▼ var s:Ten0;
▼ var r:Ten1;
▼ fun OK(s:Ten0,r:Ten1)=(r<=s);
▼ colset INTxVERI=product INT*VERI;

```

Şekil 3.2: SMTP Protokolünün CPN Tools Modeline Ait Tanımlamalar.

Protokol 3 katmandan oluşmaktadır:

- * Gönderici
- * Alıcı
- * Ağ

Gönderici: eposta Gonderme ve Geri Bildirimi Alma şeklinde 2 geçişe sahiptir. Gönderilecek epostalar UA(KA) gönderici yerinde görünmektedir. Bu yerdeki her bir jeton bir paket numarası ve veri içerir. Sonraki gönderme yeri sonra gönderilecek eposta sayısını içerir. Başlangıç olarak bu sayı 1'dir ve her onay alınması zamanında güncellenir.

Alıcı: Alıcı kısmı eposta Alma şeklinde bir tek geçişe sahiptir. Gönderici ve Ağ arasındaki arayüz MTA istemci ile D yerinden oluşurken ağ ile alıcı arasındaki arayüz MTA sunucu ve C yerinden oluşur. Alınan mesajın içeriği UA(KA) Alıcı yerinde saklanır. Bu yer alınan epostalarda bulunan metin dizilerinin birleşimi olan bir metin dizisine sahip bir tek jeton içerir. Başlangıç olarak UA(KA) Alıcı'daki Metin dizisi boşdur," "şeklinde.İletimin sonunda UA(KA) Alıcı 'da eposta1eposta2eposta3eposta4eposta5eposta6eposta7 metin dizisinin olmasını bekleriz. Sonraki alma yeri alınacak sonraki epostaların sayısını içerir.Başlangıç değeri olarak 1'dir. Epostanın başarıyla her alınışında güncellenir.

Ağ: Ağ kısmı eposta iletimi ve Geri Bildirimi iletmek şeklinde 2 geçişe sahiptir.

Protokolde 5 farklı geçiş vardır:

Eposta gönderme; MTA istemci yerinde epostanın bir kopyasını ileterek epostayı ağa gönderir. Sonraki gönderme'deki sayı hangi epostanın gönderileceğini belirtir. E-postanın UA(KA) Gönderici yerinden kaldırılmadığına dikkat edilmelidir. Sonraki gönderme'deki sayı da artmamıştır. Bunun nedeni eposta kaybolmuş ve bundan dolayı yeniden gönderilme ihtiyacı olabilir. Protokolümüz de aynı epostayı tekrar etmeye devam eder-epostanın başarıyla alındığına dair onayı alıncaya kadar.

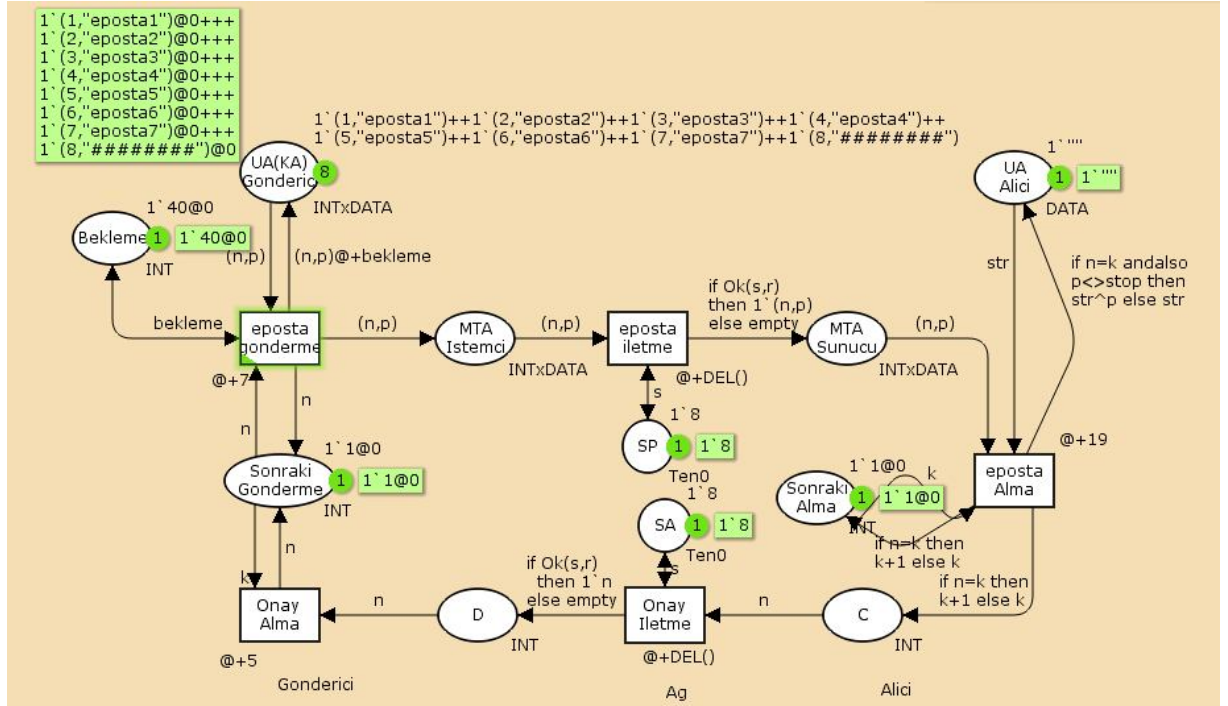
Eposta iletmek; e-postayı ağın gönderici tarafından alıcı tarafına karşı gelen jetonu MTA istemci 'den MTA sunucu'ya hareket ettirerek iletir. $OK(s,r)$ mantıksal ifadesi epostanın başarıyla iletilmiş veya kaybolmuş olduğunu belirler. r değişkeni renk kümesinde keyfi bir değere bağlanmaktadır(1 ile 10 arasında keyfi bir tamsayı).CPN Tools 10 değer arasında uygun bir seçim yapmaktadır. OK fonksiyonu r 'nin değeri

s'nin değerinden küçük veya eşitse doğru gelmektedir. Bu başarılı iletimin olasılığının SP yerindeki jeton tarafından belirlendiği anlamına gelir. SP'de 8 değerinde bir jeton verilmiştir. Bundan dolayı başarılı iletim için %80 şansımız bulunmaktadır. Ancak , başarı yüzdesini SP'deki jeton değerini değiştirerek kolaylıkla düzenleyebiliriz.

Eposta alma; bir epostayı alır ve sonraki alma'daki k sayısı ile n paket sayısının aynı olup olmadığını kontrol eder. İki sayı uyduğu zaman ,sonraki alma 'daki sayı 1 artar ve paketteki metin dizgesi UA(KA)Alıcı 'daki metin dizgesi ile birleştirilir.- mesajın sonuna işaret eden stop =”#####” oluncaya kadar. Aksi halde paket önemsenmez(yok sayılır) ve sonraki alma'daki sayı değişmez. Her iki durumda göndericinin göndereceği bir sonraki paketin numarasını içeren onay gönderilmektedir.

Geri bildirim iletilme; ağıın alıcı tarafından gönderici tarafına C'den D'ye karşı gelen jetonu hareket ettirerek onay iletir. Geçiş eposta iletimi 'ne benzer şekilde çalışır. Bu onayın SA yerindeki jeton tarafından belirlenen bir olasılıkla kaybedilebileceği anlamına gelir.

Geri bildirim alma; bir onay alır ve sonraki gönderme 'deki sayıyı günceller.



Şekil 3.3: SMTP Protokolünün Zamanlanmış CPN Tools Modeli.

```

▼ Standard declarations
▼ colset UNIT = unit;
▼ colset BOOL = bool;
▼ colset INT=int timed;
▼ colset DATA=string;
▼ colset INTxDATA = product INT * DATA timed;
▼ var n,k,bekleme:INT;
▼ var p,str:DATA;
▼ val stop="#####";
▼ colset Ten0=int with 0..10;
▼ colset Ten1=int with 1..10;
▼ var s:Ten0;
▼ fun Ok(s:Ten0,r:Ten1)=(r<=s);
▼ colset NetDelay=int with 25..27;
▼ var r:Ten1;
▼ fun DEL()=NetDelay.ran();

```

Şekil 3.4: SMTP CPN Tools Zamanlanmış Modeline Ait Tanımlamalar.

Bekleme yeri bir epostayı tekrar iletmeden önce **eposta gönderme** geçişinin ne kadar zaman beklemesi gerektiğini belirtmek için kullanılır.

Deklerasyonlardan INT renk setinin zamanlanmış olduğu görülmektedir. DATA,Ten0 ve Ten1 ise zamanlanmamıştır. INTxDATA renk kümesi INT zamanlanmış bileşenini içerdiğinden zamanlanmıştır.

Beş geçişin her birine @+ ile başlayan bir zaman alanı eklenmiştir. Zaman alanı karşılık gelen işlemin ne kadar zaman aldığını belirtmektedir. Şimdi protokol sistemindeki 5 farklı geçişi daha yakından inceleyelim.

Eposta gönderme @+7 zaman alanına sahiptir. Bu durum tüm çıkış jetonlarının zaman etiketlerine 7 zaman birimi ortak gecikmesinin eklendiği anlamına gelmektedir. **MTA İstemci** ve **Sonraki Gönderme** 'de oluşturulan jetonlar geçişin aktif olduğu r^* geçerli zamanı + 7 olan bir zaman etiketi alır. **UA(KA)Gonderici** yeri çıkış oku **UA(KA)Gonderici** 'ye eklenen jetonlar için kullanılmak üzere ek bir zaman gecikmesini belirtir. Bu jeton r^*+7+40 şeklinde (**bekleme** değişkeni 40'a bağlandığı için) bir zaman etiketi alacaktır. 7 gecikmesi bir epostanın gönderilmesi için kullanılan zamanı göstermektedir bununla birlikte 40 gecikmesi bir yeniden iletimden önce (**eposta gönderme** aynı eposta için bir kez daha aktif olmasından önce) geçmesi gereken zamanı gösterir. Yeniden iletim sadece **Sonraki Gönderme** 'deki sayı 7+40 zaman birimi için değişmemiş olarak kalırsa(eğer eposta için bu zaman diliminde hiçbir onay alınmadıysa) gerçekleşecektir.

Eposta iletme @+DEL() şeklinde bir zaman alanına sahiptir. Burada DEL NetDelay renk kümesinden rastgele bir eleman dönen (25 ve 75 arasında rastgele bir tamsayı) bir ML fonksiyonu olarak tanımlanmıştır. Bu bir iletim operasyonunun süresinin bu aralıkta değişeceği anlamına gelir.

Eposta Alma ve **Onay alma** sabit bir süreyi (sırasıyla 19 ve 5) belirten zaman alanlarına sahiptir. Bununla beraber **Onay İletme** 25 ve 75 arasında değişken bir zamana sahiptir. Tüm zaman gecikmeleri bir ML ifadesi aracılığıyla belirtilmektedir. Bu nedenle daha karmaşık gecikme türlerini belirten istatistiksel fonksiyonları kullanmak kolaydır.

Zamanlanmış CPN Tools modelimiz aracılığıyla protokolün performansını inceleyebiliriz (**bekleme** ile belirtilen yeniden iletim gecikmesi için farklı değerler denenebilir). Kısa bir gecikme gereksiz yeniden iletim yapılması ihtimalini artırır. Aynı zamanda bir **Onay Alma** operasyonunun ertelenme ihtimalini artırır çünkü **Gonderici** prosesi bir yeniden iletimde meşguldür. Uzun bir gecikme **Gönderici** 'nin bir eposta ya da onayın kaybedilmiş olduğunu fark etmesinden önce uzun zaman alabilir anlamına gelir. Belirli sayıda simulasyon (benzetim) yapılarak **bekleme** 'deki farklı değerlerle yeniden iletim gecikmesi için optimal değeri belirlenebilir (Url 4).

4. SONUÇ VE ÖNERİLER

Renkli Petri Ağları(CPN) haberleşme sistemlerin modellenmesinde kullanışlı bir yaklaşım önermektedir . Bir sistemin özellikle eşzamanlı bir sistemin çalışmasının ve performansının incelenmesi amacıyla CPN Tools programı kullanışlı modellerin tasarlanmasına imkan vermektedir. İnternet kullanımındaki yaygınlaşmaya paralel olarak SMTP protokolü kullanılarak e-posta aktarımı artmaktadır.

Bu çalışmada, Renkli Petri Ağları(CPN) temel bileşenleri örnekler verilerek anlatılmıştır. CPN Tools programı kullanılarak SMTP protokolü zamansız ve zamanlı olarak modellenmiştir.

CPN Tools programı kullanılarak haberleşme protokollerinin modelleri oluşturulup karşılaştırmalı performans analizleri yapılabilir.

KAYNAKLAR

Jensen, Kurt., Kristensen, Lars Michael ve Wells, Lisa, 2007, Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems, International Journal on Software Tools for Technology Transfer(STTT).VOL.9,no. 3S,pp.213-254.

Jensen, K. Ve Kristensen, L. M., 2009.Coloured Petri Nets:Modeling and Validation of Concurrent Systems, Springer Verlag.

Kızmaz, S., 2010 Zamanlanmış Renkli Petri Ağları ile Oturum Başlatma Protokolü(OBP)'nün Modellenmesi,Yüksek Lisans Tezi,İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü.

Zaitsev, D.A. ve Shmeleva, T. R.(2006) Simulating of Telecommunication Systems with CPN Tools Students' book on the course "Mathematical Modelling of Information Systems" for teaching of masters in communications.

Url 1 <http://webdiis.unizar.es/~lrecalde/doctorado/bibliografia/coloreadas.pdf>

alındığı tarih 30.11.2013

Url 2 <http://daze.ho.ua/cpnmp2.pdf> alındığı tarih:30.11.2013

Url 3 [http://www.researchgate.net/publication/236855917_Simple_Mail_Transfer_Protocol_\(SMTP\)/file/e0b4951983fbecf238.pdf](http://www.researchgate.net/publication/236855917_Simple_Mail_Transfer_Protocol_(SMTP)/file/e0b4951983fbecf238.pdf) alındığı tarih:30.11.2013

Url 4 [http://cpntools.org/CPN tools programı Sample CPN Models SimpleProtocol ve TimedProtokol](http://cpntools.org/CPN_tools_programı_Sample_CPN_Models_SimpleProtocol_ve_TimedProtokol)

ÖZGEÇMİŞ

Ad Soyad: Mehmet Fikri OTMAN

Doğum Yeri ve Tarihi: Çaykara / 30.07.1980

Adres: Alanaldı cad. Doğanay Sok. No:20 D:2

İçerenköy/ATAŞEHİR/İSTANBUL

E-Posta: mfotman@yahoo.com

Lisans: Yıldız Teknik Üniversitesi

Elektronik ve Haberleşme Mühendisliği, 2004