

Kavşak Sinyalizasyon Sisteminin JACK Etmen Geliştirme Platformunun Kullanılarak Oluşturulması

Serap ERGÜN¹, Tuncay AYDOĞAN²

¹Elektronik Bilgisayar Eğitimi Bölümü, Süleyman Demirel Üniversitesi, Isparta, Türkiye

²Yazılım Mühendisliği Bölümü, Süleyman Demirel Üniversitesi, Isparta, Türkiye

serapbakioglu@sdu.edu.tr, tuncayaydogan@sdu.edu.tr

(Geliş/Received: 11.09.2012; Kabul/Accepted: 17.03.2013)

Özet- Bu çalışmada, dağıtık çoklu etmen tabanlı trafik yönetim sisteminin tasarımı ve uygulanması anlatılmıştır. Sistemde kararsız yapıda olan nesnelere etmen olarak tanımlanmıştır. Bunlar araç, şerit ve kavşak etmenleridir. Araç etmenleri bilgilerini sıklıkla şerit etmenlere gönderir. Gönderdiği bilgiler arasında hız, araçtaki kişi sayısı, acil durum bilgisi gibi parametreler vardır. Şerit etmen, üzerinde bulunan araç etmenlerinin bilgilerini toplayarak özet çıkarır. Oluşturulan bilgiyi bağlı olduğu kavşak etmene gönderir. Kavşak etmen şerit etmenlerden gelen şerit bilgilerini değerlendirerek sinyalizasyona karar verir ve şerit etmenleri bilgilendirir. Şerit etmenler de araç etmenlere sinyalizasyon bilgisini gönderir. Önerilen uyarlanabilen trafik yönetimi sayesinde, trafik kazaları ve sıkışıklık azalacak, trafik akışı hızlanacak, dolayısıyla enerji tüketimi de azalacaktır. Aynı zamanda araçlar arasında haberleşmenin yapılabilmesini sağlayacak bir altyapı elde edilecektir. Çalışma, İnanç-Niyet-İstek model tabanlı olan JACK etmen geliştirme platformu üzerinde yapılmıştır.

Anahtar Kelimeler- Çoklu etmen sistemi, kavşak sinyalizasyonu, JACK etmen geliştirme platformu

Building Interchange Signalling System Using JACK Agent Development Platform

Abstract- In this study, design and implementation of a distributed multi-agent-based traffic management system is described. objects that are unstable identified as an agent. These are vehicle, lane and interchange agents. Vehicle agents often sends details to lanes agents. The information consists of speed, number of people in the vehicle, such as emergency parameters. Lane agent, collecting information on factors summarize the vehicle, sends information on the factors that created the crossroads. Intersection factor evaluates the lane informations, decides signaling and informs lane agents. Lane factors sends signaling information to vehicle factors. Recommended adaptive traffic management, decrease traffic accidents and congestion, traffic flow will be accelerated, hence the energy consumption will be reduced. At the same time getting an infrastructure which could provide communication between vehicles. The work has been conducted on JACK agent development platform which is based on Belief-Desire-Intention model.

Keywords- Multi-agent system, intersection signage, JACK agent development platform

1. GİRİŞ (INTRODUCTION)

Amerika başta olmak üzere pek çok ülkede Zeki Ulaşım Sistemleri (ITS – Intelligent Transport System) ve yapay zeka teknikleri trafiğin kontrolünde önemli bir yere sahiptir. Trafiğin kontrolü, trafik akımlarının kontrolü olarak düşünülürse, kapasite kullanımının en iyi duruma getirilmesi, kent içi yollarda ana arterler ve arterlerdeki kavşakların bölgesel kontrolü ile kent içi yolları otoyollara birleştiren katılım ve ayrılım noktalarındaki denetim ile sağlanır [1].

Parametresi fazla olan problemlerin çözümü için kullanılan optimizasyon teknikleri, mevcut şartları en iyi şekilde değerlendirerek en kısa sürede sonuca ulaştırmaktadır. Yapay zeka tekniklerinin bu tür problemlerin çözümünde kullanılması, çözümlerin gerçek hayata geçirilmesi açısından kolay ve faydalı olabilmektedir.

Çoklu etmen sistemleri, trafik yönetim probleminin çözümü için çeşitli çalışmalarda kullanılmıştır. Oliveira ve Duarte [2] çalışmasında, trafiği oluşturan araçlar nesne, trafik ışıkları ve kavşaklar ise etmen olarak

tanımlandığı çoklu etmen sistemi gerçekleştirilmiştir. Bu çözüm önerisinde araçlar etmen olarak tanımlanmadığı için araç bazındaki farklılıklardan kaynaklanabilecek problemlerde kavşak etmen doğru bir çözüm önerisi sunamaz. Paruchuri [3] çalışmasında kavşak bazında düzensiz trafik için çoklu etmenli simülasyon gerçekleştirilmiştir. Dresner ve Stone [4] çalışmalarına çoklu etmen sistemlerini kullanarak rezervasyon tabanlı kavşak kontrolü gerçekleştirmişlerdir. Bir simülasyon sayesinde, yoğun trafik saatlerinde araçların geçişlerini trafik ışıklarının sürelerini değiştirerek kavşak kontrolünü sağlamışlardır. Tektaş ve arkadaşları [1] yapay zekâ tekniklerinin trafik kontrolünde kullanılması üzerine bir inceleme yapmışlardır. Çalışmanın amacı trafik kontrolünde yapay zeka tekniklerinin nerede ve nasıl uygulandığına dair sınıflandırma yapmak olarak belirlenmiştir. Yapay zekâ tekniklerinden uzman sistemler, bulanık mantık, yapay sinir ağları ve genetik algoritma yöntemleri tanıtılarak, bu tekniklerin trafik kontrol problemlerinde kullanımları üzerine araştırma yapılmıştır.

Önerilen çalışmanın bu çalışmalardan farkı, günlük hayatta uygulanabilirliğinin yüksek olması, iki taraflı haberleşme yapılması ile birlikte tüm nesnelerin kontrol altında olması, kullanılan etmen platformunun BDI (İnanç-İstek-Niyet) standartlarına uygun olması, birden fazla kavşak için çoğullanabilir özelliğinin olmasıdır.

Önerilen uyarlanabilen trafik yönetimi sayesinde, trafik kazaları ve sıkışıklıkların azalması, trafik akışının hızlanması, dolayısıyla enerji tüketiminin de azalması hedeflenmektedir. Aynı zamanda araçlar arasında haberleşmenin yapılabilmesini sağlayacak bir altyapı elde edilmiş olacaktır.

2. JACK ÇOKLU ETMEN GELİŞTİRME PLATFORMU (JACK MULTI-AGENT DEVELOPMENT PLATFORM)

JACK etmen geliştirme aracı; ticari bir etmen tabanlı geliştirme ortamıdır[5]. JACK, JAVA üzerine kurulmuş bir geliştirme ortamıdır. Dolayısıyla etmen davranışlarını uygulamak için JAVA sınıflarının bir uzantısı şeklinde hareket eder. Ayrıca yürütülecek proje içerisindeki etmenleri tanımlamak için de grafiksel kullanıcı ara yüzüne sahiptir. Bu ara yüz, kullanıcılara etmenlerin görüşlerinde, inanç setlerinde, kabiliyetlerinde ve planlarında değişiklik yapmasında kolaylık sağlamaktadır[6].

JACK; zeki etmenleri ulaşmak istedikleri açık hedeflere, yapay zekanın teorik İnanç-İstek-Niyet (BDI) modeline göre üstesinden gelecekleri olaylara sahip özerk yazılım bileşenleridir. Ayrıca farklı mekanizmalar ile uzak ya da yerel haberleşmeyi de desteklemektedir. JACK, FIPA standartlarına uymamaktadır [6].

2.1. JACK Etmen Dili Bileşenleri (JACK Agent Language Components)

Bir JACK uygulaması, bir ya da birden fazla etmen/takım, plan, olay ve inanç setleri ya da görüşler açısından tanımlanmaktadır. Uygulamalar seçmeli olarak yetenekleri (capability) de içerebilirler.

Etmenler ya da takımlar, bir uygulamanın özerk sayısal varlıklarını ifade etmek için kullanılır. Takım sınıfı ise, çoklu etmen davranışlarının koordine edilmiş görünümünü bütünleştirmek için kullanılır [4]. Takımlar, etmen fonksiyonlarının birçoğunu içermektedir. JACK etmen dilindeki programlama yapısı aşağıdaki bileşenleri içermektedir:

Olay: Olaylar, etmenler/takımlar da merkezi motivasyon faktörüdür. Olaylar olmadan bir etmen/takım, düşünmek ya da hareket etmek için uyusuk, motivasyonsuz bir haldedir. Olaylar, dışsal bir uyarıcıya karşılık olarak ya da içsel bir hesaplamının sonucu olarak yaratılabilirler. JACK, iki ana olay kategorisine sahiptir: Normal olaylar ve BDI olayları. Normal olaylar, çevresel anlayışlar gibi geçici beklenmeyen olayları temsil etmek için kullanılır. Örneğin, eğer etmen/takım ilk denemesinde olayı başarılı bir şekilde ele alamaz ise olay bırakılır çünkü bu ara periyotta etmenin içerisinde bulunmuş olduğu ortam değişmiş olacaktır. Buna zıt şekilde BDI olayları, geçici uyarılardan daha ziyade hedefleri temsil etmek için kullanılır. Bir etmen/takım, bir BDI olayına hizmet verdiği zaman olayı başarılı bir şekilde ele alır: Hedef sağlanana kadar birçok alternatif çözüm yolları denenir [6].

Plan: Planlar, olaylara nasıl karşılık verileceğini tanımlayan prosedürlerdir. Bir olay yaratıldığı zaman JACK, o olayla ilişki olan plan setlerini hesap eder (olayla uyuşan planlar). Her bir ilişkili plan, içerik duruma göre filtreden geçirilir. Bir planın, hangi şartlar altında uygulanabileceğini tanımlayan durumlar buna örnek olarak verilebilir. Daha sonra, içerik şartları mevcut durumları tarafından karşılanan ilgili planlar seti, etmenin bir sonraki amacını oluşturacak olan planın seçileceği tartışma sürecinin konusu haline gelir. Planlar, olaya karşılık verildiğinde çalıştırılacak adımların tanımlandığı bir gövdeye sahiptir. Bir plan adımı, ya bir JACK muhakeme ifadesi ya da bir JAVA ifadesidir. Muhakeme metodlarının kullanılması ile plan gövdesinin fonksiyonel ayrıştırılması için destek sağlanır. JACK operasyon zamanı altyapısı, plan adımlarının çalışmasının (muhakeme metodu çalışmasını da içerir) çok küçük olmasını garanti eder [6].

İnanç seti: İnanç setleri, ilk olarak etmeni tanıtan inançları temsil etmek için kullanılır. Bir inanç seti ilişkisinin değer alanları, eski JAVA tiplerini ve kullanıcı tanımlı sınıfları içeren herhangi bir veri tipi olabilir. Bir inanç seti açık kaynak ya da kapalı kaynak olabilir ve JACK çekirdeği, inanç setinin mantıksal tutarlılığını sağlar. İnanç setleri standart bilgi erişimi altında ya da üzerinde birçok yararlı fonksiyon sağlar. Örneğin, bir olay güncel bir inanç seti üzerinden otomatik olarak

yaratılabilir ve bu durum etmenin aktivitelerini değiştirip değiştirmemesi gerekliliği üzerine düşünmesine yol açar[6].

Görüş: Bir görüş, etmenlere kendi arabirimleri ile ilgilenmeden heterojen veri kaynaklarını kullanmalarına olanak sağlayan veri soyutlama mekanizmasıdır. Aslında görüşler, ara birimi bir inanç seti ile aynı olan dışsal veri kaynakları haline getirir [6].

Etmen: JACK, BDI modeli üzerine olduğu için bir JACK etmeni inançlara, isteklere ve niyetlere sahiptir. Bunlar etmenin içsel durumunun bir parçasıdır ve sistemdeki diğer etmenler tarafından direk olarak erişilemez. Bratman [7] tarafından tanımlandığı gibi inançlar, etmenin planları, inanç setleri ve görüşleri tarafından temsil edilir. Bunlar, planlar olması halinde etmenin sahip olduğu prosedürel bilgiler olan bilgiyi ve inanç setleri ve görüşler olması halinde sahip olduğu gerçeklikleri tanımlar. Aslında JACK isteklerin belirgin bir tanımlanmasına sahip olmamasına rağmen bir JACK etmenin istekleri, etmenin mevcut içsel durumuna uygulanabilir olan plan setleri içerisine dâhil edilmiştir. Her uygulanabilir plan, gevşek bir biçimde bir istek ile uyur. Diğer istekler, aynı sayısal kaynaklar için yarışırken etmenin yükleneceği bir aktivite buna örnek verilebilir. Uygulanabilir bir plan seçildiği zaman, istek niyet haline gelir. Örneğin, etmen seçilmiş bir planı kullanarak isteğin karşılanmasını sağlar [6].

Takım: Takımlar, sosyal yapıların ve koordine edilmiş davranışların modellenmesini kolaylaştıran BDI modelinin bir uzantısıdır. JACK, takım rotasyonunu ayrı muhakeme varlıkları olarak tanıtır (takım üyelerinden ayrı olarak). Bir takımın davranışı, özellikle takım üyelerinin koordine edilmiş aktivitesi, direk olarak takım varlığı tarafından tanımlanır. Böylelikle yazılım modelinde, her takım üyelerinin sahip oldukları inanç setlerinden farklı olan inanç setleri ile bir olarak yer alırlar. Bu üretken takım tabanlı yetenek, üzerinde birçok takım algoritmalarının tasarlanabileceği ve test edilebileceği esnek bir zemin sağlar. JACK, takım üyeleri arasındaki sosyal ilişkileri aynı zamanda koordine edilmiş aktiviteleri ifade etmek için kullanılan uygun yapıları ile takıma dayalı çözümlerin programlamasını destekler. JACK, bir uygulamadaki koordine edilmiş aktiviteleri çalıştırmak için ihtiyaç duyulan iletişim servislerini de bünyesinde barındırır [6].

Yetenek: Yetenekler, bileşenlerin etmenler arasında tekrar kullanılabilmesini sağlamak için bir etmenin fonksiyonel bileşenlerini (olaylar, planlar, inanç setleri ve diğer yetenekleri) organize etmede kullanılır. Yetenekler alt yetenekleri içerebildiği için bir etmenin becerisi, bir yetenekler hiyerarşisi olarak tanımlanır. Yetenekler, uygulamalar arasında tekrar kullanılabilen etmen tabanlı işlevsellik kütüphanelerinin gelişimini destekleyen yazılım ihtiyacına karşılık olarak JACK'e eklenmiştir [8].

2.2. JACK Geliştirme Ortamı (JACK Development Environment)

JACK, karmaşıklığı dolayısı ile sadece insanlar tarafından ele alınan görevler üzerinde etmenlerin rol almasına ihtiyaç duyulduğu birçok alanda uygulanmaktadır (Ör: meteorolojik değişimler). Ayrıca JACK geniş bir şekilde insan davranışlarını modellemede de kullanılmaktadır. Örneğin, askeri komutalar ve savaş pilotları. Bu tür uygulamalar, bilgi mühendisliği aşaması ve askeri simülasyon olması durumunda uzun ve kapsamlı bir model doğrulama aşamasını içerir. Konu içerik uzmanının alana özgü bilgisinin doğru bir şekilde uygulamasını sağlamak, usta bir analiz ve JACK programlamada deneyim gerektirir. Özellikle Konu İçerik Uzmanının, geri beslemelerinin bir sonucu olarak etmen ihtiyaçlarının modifiye edildiği yerlerde analizcinin JACK programlama yeteneğine sahip olduğu zamanlarda, geçmişte bu süreç çok verimli olarak geçmiştir. BDI modeli, konu içerik uzmanının problem çözme faaliyeti hakkındaki düşünme şeklini doğal olarak eşlemesine rağmen, bu uzmanlıklarını ancak bir programcı yardım ile JACK kodlarına çevirmenin yarattığı zorluk geçmişte JACK'in kabulünü sınırlandırmıştır [6].

Bu düşünce ile JACK, etmen uygulamalarının tasarımı, uygulaması ve takibi destekleyen grafiksel bir araç seti ile geliştirilmiştir. JACK Geliştirme Ortamı (JDE) etmene dayalı uygulamalar geliştirmeye olanak sağlayan grafiksel bir araç seti sunar. Bu grafiksel ara birimde etmenler, takımlar, takım yapıları ve bileşenler bir biri ile ilişkilerini gösteren çizgiler tarafından bağlanan ikonlar tarafından temsil edilirler. Bu kabataslak sunuş hedefleri, içerikleri, düşünme adımlarını ve etmenlerin/takımların eylemlerini tanımlamak için doğal bir ortam kullanır. Bu grafiksel ve doğal dil tanımlamaları daha sonra, Konu İçerik Uzmanları/Analist özelliklerine çok yakın haritasını çıkaran hesaplama yapısına sahip, uygulama davranış modellerinin üretilebilmesi için programcılar tarafından zenginleştirilmiştir. Bu bilgi kodlama sürecini kolaylaştırmış ve Konu İçerik Uzmanına ve analistlere, uygulamanın çalışma davranışlarını izleyebilme olanağı sağlamış ve dolayısı ile bir ayarlama gerekli olup olmadığını ya da ne zaman ayar yapılması gerektiğini belirleme olanağı sağlamıştır. Karar destek ve insan davranışı modelleme alanlarında JACK'in grafiksel araç seti, etmen/takım davranışları değiştirilmesinin, değerlendirmesinin, şifrelemesinin tekrarlı süreci üzerinde analiste kontrol sağlaması hayati bir özelliktir [8]. JACK hakkında daha geniş bilgi için JACK el kitabı incelenebilir [9]. Burada bileşenler şu şekilde temsil edilmektedir:

- Etmen-dikdörtgen içerisindeki çubuk insan,
- Yetenek-dikdörtgen,
- Plan-köşeleri yuvarlatılmış dikdörtgen,
- İnanç seti- silindir
- Olay-mektup

2.3. JACK' in Temel Plan ve Mimarileri (The Basic Plan and Architectures of JACK)

BDI mimarisi ve JACK geliştirme platformunun etmen plan ve takım planlarına olan yaklaşımların yazılımsal farklılıkları Şekil 1, 2, 3, 4' de gösterilmiştir.

```
public plan ProcessRequest extends Plan {
#handles event Request req;
#sends event Response resp;
context () {
req.isValid;
}
#reasoning method body() {
// Can contain Java code as well
//as JACK @-statements
...
@reply (req.resp.response(...));
}
}
```

Şekil 1. Basit Bir JACK Planı (A Simple Plan) [10]

- 1- Event posted.
- 2- Determine the set of relevant plans.
- 3- Determine the applicable plans.
- 4- Select an applicable plan and run it.
- 5- If plan fails, goto step4.

Şekil 2. BDI mimarisinde olay yaklaşımı
(Approach to event of BDI Architecture) [10]

- 1- Event posted.
- 2- Find plans that #handle it.
- 3- Determine the set of relevant plans using relevant() method.
(Relevant method can only Access event, not beliefs).
- 4- Determine applicable plans using context().
- 5- Select a plan and run its body().
(meta-reasoning can be used to make the selection).
- 6- If plan fails, goto step4.
(recompute applicable plan set)

Şekil 3. JACK platformunda olay yaklaşımı
(Approach to event of JACK platform) [10]

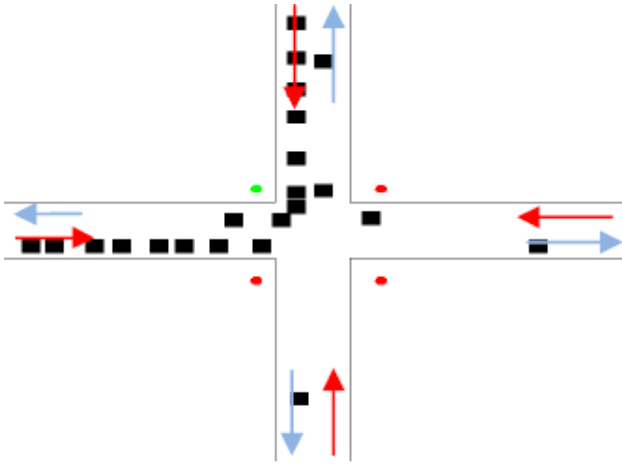
```
Teamplan FeedBaby extends TeamPlan {
#handles event BabyHungary pfv;
#uses role Parent as p1;
#uses role Parent parents as p2;
//establish the task team
#reasoning method establish () {...}
body() {
@parallel(ParallelFSM ALL false null) {
@team_achieve (p1,p1.prepareFood.pf());
@team_achieve (p2,p2.calmBaby.cb());
@team_achieve (p2,p2.feedBaby.fb());
} //body
} //FeedBaby team plan
```

Şekil 4. Basit bir takım planı (A simple team plan) [10]

3. TRAFİK KAVŞAK YÖNETİMİNDE ETMEN KULLANIMI (AGENT USE OF INTERCHANGE TRAFFIC MANAGEMENT)

3.1. Sistem Tasarımı (System Design)

Problemin çözümü için; trafik sisteminin en temel birimi olan kavşak ele alınmıştır. Her ne kadar bu çözüm bir kavşak üzerinde uygulanmış olsa da ardışık olarak diğer kavşaklara da uygulandığında; trafik akışını artıracaktır. Kavşağa bağlı gidiş dönüş yönlerine sahip dört şerit bulunmaktadır. Her şeridin sonunda trafik ışıkları mevcuttur ve araçlar bu trafik ışıklarına göre hareket edeceklerdir. Karar verme, iletişim kurma ihtiyacı olan yapılar etmen olarak tasarlanmıştır. Bunlar, kavşak, şerit ve araçlardır. Kavşak için kavşak etmen tipinde bir etmen yaratılarak; kavşaktaki trafik sinyalizasyonu hakkında karar vermesi ve verdiği kararı kendine bağlı olan şeritlere dağıtması sağlanmıştır. Her şerit, şerit etmen tipinden etmen ile yaratılmış ve araçlar ile kavşak arasında ara yüz oluşturmuştur. Araçlar ise; araç etmeni tipinden yaratılarak, şeritler üzerinde hareketi sağlanmış ve yol üzerinde yük yaratarak, kavşağın bu yükü dağıtmadaki yeteneği izlenmiştir. Sistemdeki şerit etmenleri birbirleri ile aynı özelliklere sahiptir. Kavşağa bağlı olarak tanımlanan şeritler üzerinde gidiş ve dönüş olmak üzere iki yön vardır [11]. Söz konusu kavşağın durumu Şekil 5' de gösterilmiştir.



Şekil 5. Kavşak Örneği (Intersection Sample)

Şekil 5’de araçların şeritler üzerindeki akış yönleri gösterilmiştir. Araçlar kavşağa geldiklerinde şerit değiştirerek istedikleri yöne hareket edebilirler.

3.1.1. Araç Etmeni (Vehicle Agent)

Sahadaki her araç için araç etmeni yaratılacaktır. Araç etmeninin sahip olduğu özellikler, aracın fiziksel durumundan ve aracı kullanan kişiden gelen özelliklerdir. Bu özellikler, araç ontolojisi tipinde ifade edilir ve her aracın özellikleri kendi ontoloji bilgisinde yer alır. Araç değişken V hızı ile hareket edebilir, kavşaklardan geçebilir, bir rotaya sahip olabilir. Araç etmeninin sahip olduğu davranışlar ise, şerit etmen bulmak, şerit etmene kayıt yaptırmak, araç bilgisini şerit etmene göndermek, şerit bilgisini beklemek, şerit etmene kayıt sil mesajı göndermektir. Sistemdeki araçların hızlanma algoritması şu şekildedir:

```
If (YeniAraç=true)
  Rasgele_Hiz_Hesapla()
Else If (Kavşak_Geçis_Mesafesinde)
  Kavşak_İçindeki_Hizi_Hesapla()
Else
  Kontrollü_Sürüş_Mesafesi_İçin_Hizi_Hesapla
  (Yol_Sonu_Mesafesi, AraçTipi, GeçişHakkı)
```

3.1.2. Şerit Etmeni (Lane agent)

Her şerit, şerit etmen tipinden etmen ile yaratılmış ve araçlar ile kavşak arasında mesajlaşmada ara yüz oluşturmuştur. Şeridin sahip olduğu özellikler; çift yönlü, N araç kapasitesine sahip ve kavşağa bağlı olmasıdır. Sahip olduğu bilgiler ise; üzerindeki her araca ilişkin araçların ontolojisiyle gelen bilgilerdir. Şerit, araçların hareket edebilmesi için fiziksel bir alan sunar, hiç bir eylem gerçekleştirmez. Dolayısıyla şerit etmeninin sahip

olduğu davranışlar artı bir özelliktir. Şerit etmeninin sahip olduğu davranışlar; kavşak etmeni bulmak, kavşak etmene kayıt yaptırmak, şerit bilgisini kavşak etmene göndermek, kavşak bilgisini beklemek, araç kayıt işlemlerini yapmak, araç bilgisi beklemek, araca şerit bilgisi göndermektir.

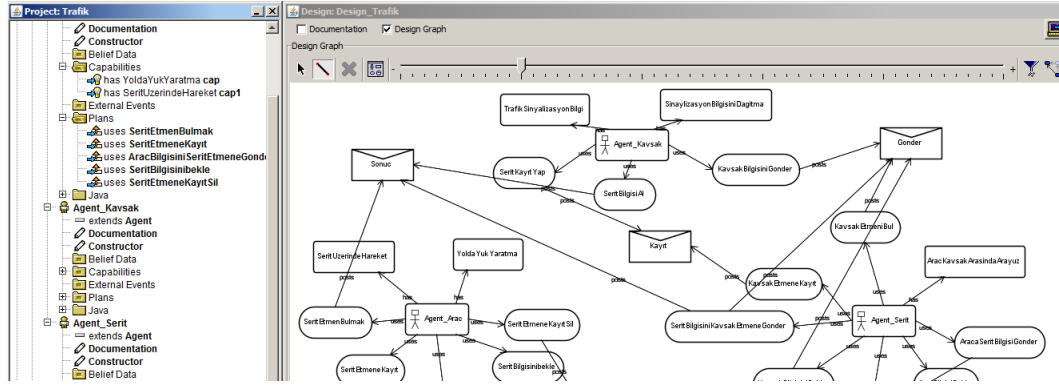
Şerit etmen her zaman mesaj bekleme ve mesaj gönderme halindedir. En önemli özelliği araçlardan gelen bilgi mesajlarını kavşak etmen için anlamlı hale getirerek kavşağa bilgilendirme mesajı göndermektir. Tasarım araç etmenlerle kavşak etmenler iletişim kuracak şekilde de yapılabilir. Ama sadece bir etmenin tüm etmenlerle sıklıkla konuşmasının performans düşürücü olabileceği düşünülmüştür. Dolayısıyla şerit etmenler yaratılmıştır. Şerit etmeni aynı şekilde kavşak etmeden mesaj bekleyip, gelen mesajları uygun formata sokarak kendi listesindeki tüm araçlara gönderir. Şerit etmenin kavşak etmene gönderdiği mesajın içeriği şerit ontolojisi tipindedir.

3.1.3. Kavşak Etmeni (Intersection Agent)

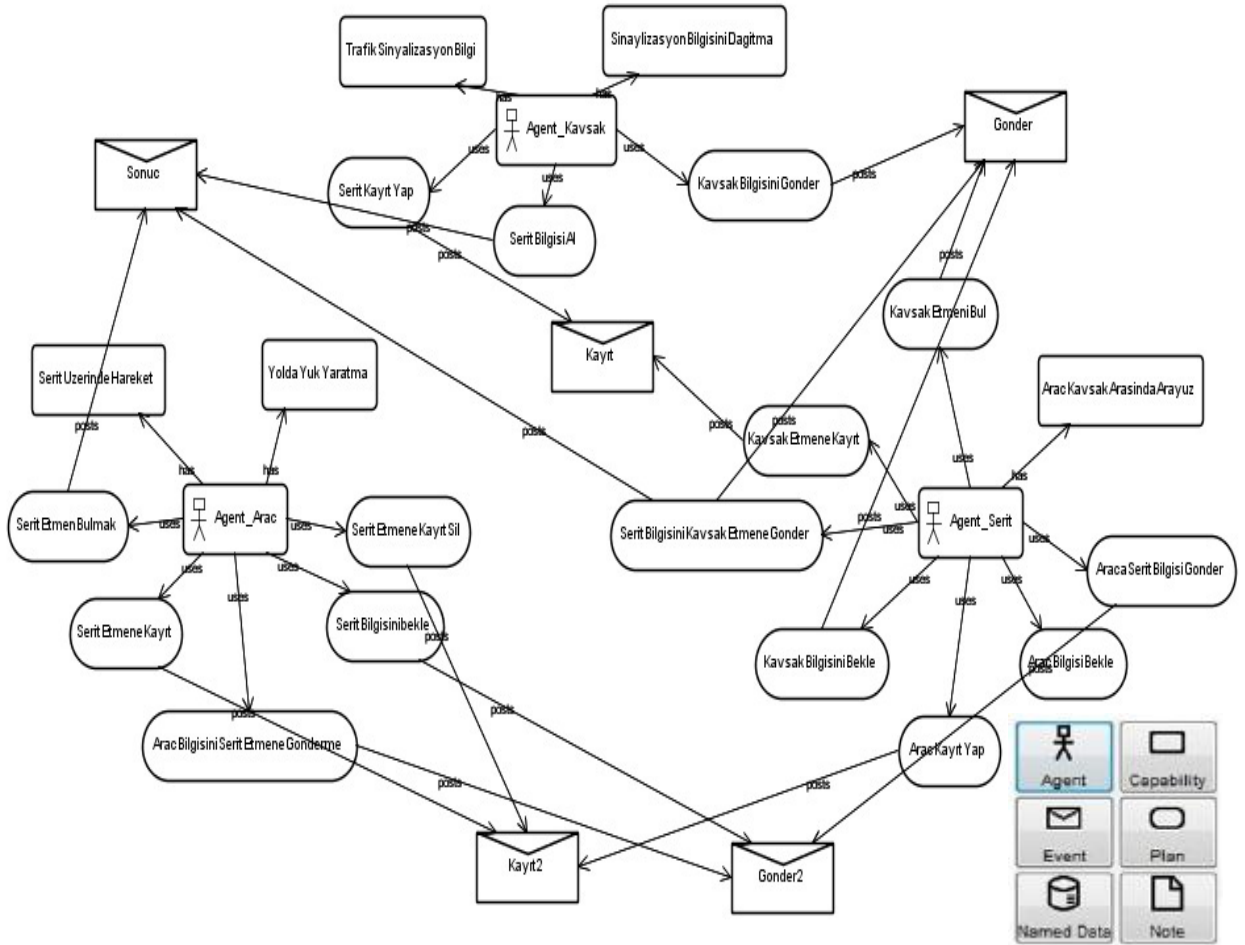
Trafik ışıklarını yönetecek ve sinyalizasyona karar verecek olan etmendir. Kavşak etmeni şerit etmenlerden gelen bilgilerle beslenir, kendine ait bilgileri oluşturarak Kavşak ontolojisinde taşır. Kavşak etmeninin sahip olduğu davranışlar, şerit kayıt işlemlerini yapmak, şerit etmeden şerit bilgisi beklemek, kavşak bilgisini göndermektir.

3.2. Sistemin JACK ile Oluşturulması (Creating a system with JACK)

Sistemde öncelikle etmenler oluşturulup yetenekleri, planları, olayları tanımlanmıştır. Üç etmen de birbirleriyle iletişim halinde oldukları için olaylarla birbirlerine bağlanmışlardır. Sistemin ara yüzü Şekil 6’ de görülmektedir. Ara yüzün sol tarafında etmenler, planlar vb. görülmektedir. Etmenler arası ilişkiler de Şekil 7’ de verilmiştir. Şekil 7’ de etmenlerin, planların, yeteneklerin gösterim şekilleri görülmektedir. Her bir etmenin özellikleri kendi başlıkları halinde önceki bölümlerde açıklanmıştır.



Şekil 6. Sistemin ara yüzü (The system interface)



Şekil 7. Etmenler arasındaki ilişkiler (Relationships Between Agents)

4. SONUÇLAR (RESULTS)

Günümüzde trafiğin kontrolü amacıyla geleneksel kontrol tekniklerine ilave olarak ileri trafik kontrol teknikleri de kullanılmaya başlanmıştır. Bu tekniklerinin önemli bir bölümünü yapay zekâ teknikleri (artificial intelligence, AI) oluşturmaktadır [1].

Akademik çalışmalarda görülmektedir ki gerçek hayat problemlerinin çözümü için yapay zekâ tekniklerinin

kullanımı tercih edilmektedir. Bu alanda başlıca kullanılan yöntemlerden uzman sistem ve bulanık mantık metodu transit ile kentçi yollardaki trafik akımlarının kontrolü problemlerine, yapay sinir ağları trafik akışı tahminine çözüm ararken genetik algoritmalar trafik optimizasyonunda kullanılmıştır.

Günümüz önemli problemlerinden biri olan trafik yönetim sistemi için önerilen çözüm çoklu etmen yapılarıyla gerçekleştirilmiştir. Geliştirilen çalışmada örnek senaryo için

nasıl bir tasarım oluşturulduğu anlatılarak JACK etmen tabanlı yazılım ortamı kısaca tanıtılmıştır. JACK ile örnek senaryonun gerçekleşmesi üzerinde durulmuştur.

Gerçekleştirilen bu uygulama kullanılarak, çoklu etmen yapısı ile gerçekleştirilen trafik yönetim sisteminin başarısı ölçülmüştür. Trafikçi oluşturan her kararsız yapı, sistemde etmen olarak tanımlanmıştır. Araçların anlık durumları, sürücülerine bağlıdır. Şeridin durumu o an şeritte bulunan araçlara bağlıdır. Kavşak durumu ise kendine bağlı olan şeritlerdeki araçlara bağlıdır. Bu kararsızlığın nedeni olan araçlar hakkında anlık bilgiye şeridin ve kavşağın ulaşabilmesi önemlidir.

5. KAYNAKLAR (REFERENCES)

- [1] Tektaş, M., Akbaş, A., Topuz, V., Yapay Zeka Tekniklerinin Trafik Kontrolünde Kullanılması Üzerine Bir İnceleme, **1. Uluslararası Trafik ve Yol Güvenliği Kongresi**, Ankara, 2009.
- [2] Oliveira E. ve Duarte N., A Multi-Agent System for Simulation of Traffic Control, Proceedings of the ESM 2005 (European Simulation and Modelling), 128 – 135, 2005.
- [3] Paruchuri P., Pullalarevu A.R., Karlapalem K., Multi Agent Simulation of Unorganized Traffic, **International Conference on Autonomous Agents**, 176 – 183, 2001
- [4] Dresder, K., Stone, P., Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism, **The Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 04)** pp. 530-537, New York, New York, USA, July, 2004.
- [5] İnternet: <http://aosgrp.com/index.html>, 11.06.2012.

Gerçekleştirilen sistemde de araçlar bilgilerini üzerinde oldukları şeritle paylaşır, şerit bilginin özetini çıkararak kavşağa iletir. Kavşak ise bu bilgiyi değerlendirerek sinyalizasyonu yönetir. Amaç araçların kavşaktan en kısa sürede ayrılmalarını sağlamaktır.

JACK' in temelde güçlü bir etmen geliştirme ortamı olarak kullanılabilmesini; fakat bütüncül bir etmen tabanlı yazılım geliştirme ortamı olabilmesi için sistem çalışırken oluşan bazı çalışma zamanı hatalarının giderilmesi ve görsel ara yüzün geliştirilmesi gerektiği söylenebilir [12].

- [6] Şahin, C., Kaplanoğlu, V., Baykasoğlu, A., Erol, R., 2011. Çoklu Etmen Geliştirme Platformu. **Endüstri Mühendisliği Yazılımları ve Uygulamaları Kongresi**, 30 Eylül-01/02 Ekim 2011.
- [7] BRATMAN, M. E., "Intention, plans, and practical reason", 1999.
- [8] EVERTSZ, R., FLETCHER, M., JONES, R., JARVIS, J., BRUSEY, J. and DANCE, S., "Implementing industrial multi-agent systems using JACK", **Prog. Multi-Agent Systems**:18-48, 2004.
- [9] Jack Agent Manual. Australia: Agent Oriented Software Pty. Ltd. ; 2008.
- [10] Winikoff, M., JACK™ Intelligent Agents: An Industrial Strength Platform. Multi-Agent Programming, Multiagent Systems, **Artificial Societies and Simulated Organizations**, Volume 15,11.175-193, 2005.
- [11] Soysal, K., Balçılar, M., Sönmez, A.,C., Multi Agent Based Traffic Management System, **Eleco'2008 Elektrik - Elektronik - Bilgisayar Mühendisliği Sempozyumu Ve Fuarı**, Bursa, 2008.
- [12] Y. Kaymak, K.D. Teket, S. Demirci, S. Getir, R.C. Erdur, G. Kardaş, "JACK Kanı-İstek-Hedef Modeline dayalı Çok-etmenli bir Kütüphane Yönetim Sisteminin Prometheus Metodolojisi ile Geliştirilmesi", **UYMS**, 2010.