

Etmen tabanlı dağıtılmış olay sistemi

Özgür Koray ŞAHİNGÖZ*, A. Coşkun SÖNMEZ

İTÜ Fen Bilimleri Enstitüsü, Kontrol ve Bilgisayar Mühendisliği Programı, 34469, Ayazağa, İstanbul

Özet

Geleneksel istemci-sunucu mimarisinin kullanıldığı küçük ölçekli dağıtılmış sistemler uçtan uca ve senkron haberleşmeleri sebebi ile statik uygulamaların geliştirilmesinin ötesine geçmemiştir. Bu haberleşmenin büyük ölçekli sistemlerde kullanılması ise uygulamaların hantal ve kullanışsız olmalarını sağlamıştır. Bu kısıtlamalar, dinamik olan ve bağlantısız uygulamaların çalıştığı bir ortamda, daha esnek haberleşme modellerini kullanan yazılım sistemlerinin geliştirilmesi gerekliliğini ortaya çıkarmıştır. Yapılan çalışmada sistem bileşenleri arasında haberleşme için kullanılan olay kavramının üretim niteliği geliştirilmesi amaçlanmış olup olaylar sadece ham veri üretiminin yanı sıra, eylemlerin de üretildiği ve saklandığı ileri bir düzeye yerleştirilmiştir. Bu düzeydeki bir olay, işlenebilir veriler üretebileceği gibi, kendine özgü bir kontrol akışına sahip olan bir eylemler dizisinin tetikleyicisi de olabilir. Söz konusu eylemler belirli bir amacı gerçeklemeye yönelik işlem dizisine, karar mekanizmalarına ve çoğunlukla birlikte üretildikleri bir veri kümesine sahip olmaktadır. Sistemde olay yapısı, içerdiği eylem dizisinin ve bu eylemleri yerine getirmede kullanacağı veri bloğunun yer aldığı bir “gezgin etmen” yapısıyla temsil edilmektedir. Bu yeni yapıdaki olaylara “agent-agent event” adı verilmiş olup sistem bir bütün olarak “Etmen Tabanlı Dağıtılmış Olay Sistemi” veya kısaca “ABDES Sistemi” olarak adlandırılmıştır. Geliştirilen sistem, kayıt/yayım haberleşme modelinin asenkron haberleşme, bir uçtan çok uca haberleşme ve zayıf bağlı haberleşme özellikleri ile gezgin-zeki etmenlerin amaca yönelik davranmalarını, sosyal yeteneklerini, tepkisel davranabilmelerini ve otonom olma özelliklerini esnek bir ortamda birleştirmiştir.

Anahtar Kelimeler: Dağıtılmış olay sistemi, gezgin etmenler, kayıt/yayım modeli.

*Yazışmaların yapılacağı yazar: Özgür Koray ŞAHİNGÖZ. o.sahingoz@hho.edu.tr; Tel: (212) 663 24 90 dahili: 4321. Bu makale, birinci yazar tarafından İTÜ Fen Bilimleri Enstitüsü, Kontrol ve Bilgisayar Mühendisliği Programı’nda tamamlanmış olan "Etmen tabanlı dağıtılmış olay sistemi" adlı doktora tezinden hazırlanmıştır. Makale metni 18.05.2006 tarihinde dergiye ulaşmış, 27.07.2006 tarihinde basım kararı alınmıştır. Makale ile ilgili tartışmalar 30.04.2008 tarihine kadar dergiye gönderilmelidir.

Agent based distributed event system

Extended abstract

In recent years, a growing attention has been paid to the publish/subscribe communication paradigm as a means for disseminating information, also called as events, through distributed systems on wide-area networks. As it allows events to be propagated in a way that is completely hidden to the component that has generated them as well as to its receivers, it is particularly interesting when easy reconfiguration and decoupling among components in a distributed system is required. The historical development of publish/subscribe systems has followed a line which has evolved from channel-based systems, to subject-based systems, next content-based systems and finally type (object) based systems. In this study, we propose a new model for agent based distributed event systems, called as ABDES system, which combines the advantages of distributed event systems (like loosely coupled, asynchronous and one to many communication) and the advantages of intelligent mobile agents (like autonomy, reactivity, social ability and goal driven execution) into a flexible and extensible distributed execution environment. The major novelty of the model is that an event is represented by a mobile agent, (called as **agvent-agent event**), which is treated as a first class citizen of the system and given autonomy features to travel between system components. ABDES system differs from other distributed event systems with its distinct characteristics that are described below.

Event Definition: In most event systems, events are defined as low-level messages, which consist of record like structures, list of strings, tuple based structures, etc. In type-based systems, events are defined as objects and viewed as main component of the system. Nevertheless, they are not autonomous. In ABDES system, events are represented as mobile agents that have their own goals, beliefs and behaviors that they acquire at their creation. When an agvent reaches to an agvent server, it examines the routing table of the server and selects its targets autonomously. This approach reduces the load and complexity of agvent servers as well.

Subscription Mechanism: In most distributed event systems, subscribers register on a channel, on a specific topic or on a specific content of an event message. In ABDES system, subscribers register on types of agvents. For example, a subscriber can register on an agvent, which is an instance of

“Agv_type1” class, specifying certain constraints based on its advertised attributes and behaviors.

Information hiding: In previously developed event systems, an event server can, actually has to, access the content of the published event data. If you have a valuable data and you want to keep it secret it is really a hard work. In ABDES system, a published agvent searches the knowledge base of the agvent server, selects the registered subscribers, clones itself and sends clones of agent to each subscriber on the selected list. Therefore, the agvent server has no access to the content of the published event data, which simplifies its role and consequently facilitates the server development process. Information hiding also meets requirements of certain applications where secrecy of event data is essential for them.

User defined agvent types: Distributed event systems generally use predefined event types. Therefore, to add a new event type you have to make changes in the dispatch service and also at publisher and subscriber sites. In ABDES system, a publisher can create its own agvent type and declare its properties and behaviors through an advertisement message. Once an agvent type is announced to the system, subscribers can register on agvents of that type.

Filtering Mechanism: Distributed event systems use client side filtering. A client defines its profile by a subscription message and system forwards incoming events to this client according to its profile. There is not any possibility for producers to put their dispatching criteria to select the clients. Because of the agvents autonomous and self routing structure we can achieve the producer side filtering also.

Faults and failures are inevitable in a distributed system built over a wide area network and they should be tolerated for continuing system transactions. Therefore we also developed a fault tolerance mechanism by dynamically reconfiguring the connections among agvent servers in order to create paths that increase the performance of message routing. We think the new model will serve as an effective choice for large scale distributed computing application, automatic virus updating and several information oriented applications, such as e-commerce and information retrieval.

Keywords: Distributed event system, mobile agent, publish/subscribe model.

Giriş

Internet genişliğindeki geniş alan ağlarında ölçeklenebilirlik sistemin her katmanında sağlanması gereken önemli bir özelliktir. Bu sistemlerdeki haberleşme altyapısı potansiyel olarak milyonlarca dinamik katılımcı içerebilir. Bu nedenle bu sistemlerin merkezi bir yapıda değil, dağıtılmış bir yapıda gerçekleşmesi gerekmektedir (Cugola vd., 2001). Olay tabanlı haberleşme modeli bu şekildeki büyük ölçekli sistemlerin geliştirilmesinde yeni bir paradigma olarak karşımıza çıkmaktadır (Meier, 2000). Bu modelin zayıf bağlı haberleşme yapısı sayesinde sistem bileşenleri ölçeklenebilir hale getirilebilmekte ve sistem geliştiricilerine basit bir programlama modeli sunulabilmektedir.

Olay tabanlı sistemlerde olaylar sistemin temel haberleşme mekanizmalarıdır. Olay, sistem katılımcılarından birinin ilgi duyduğu bir değişikliğin olması durumunda yapılan bildirim olarak düşünülebilir. Olay tabanlı haberleşme sistemleri genellikle kayıt-yayım haberleşme modeli kullanılarak gerçekleştirilir.

Kayıt-yayım haberleşme modeli özellikle dağıtılmış olay sistemlerinde, sistemin bileşenleri için sağlamış olduğu anonim haberleşme özelliği sayesinde fonksiyonel bir model olarak görülmektedir (Mühl, 2002). Bu modelde katılımcıların, diğer katılımcılar hakkında herhangi bir bilgiye sahip olmasına gerek yoktur, sadece haberleşecekleri veri formatı üzerinde fikir birliği içinde olmaları yeterlidir. Sistem katılımcıları iki ana kategori altında gruplandırılabilir: yayımcılar ve kayıtçılar. *Yayımcılar* belirli koşulların sağlanması durumunda olay mesajları üretmekte ve sisteme göndermekte, *kayıtçılar* ise haberdar edilmek istedikleri olaylar hakkındaki kayıt taleplerini sisteme bildirerek ilgili olayların olması durumunda haberdar edilmeyi beklemektedirler. Gönderilen olay verisinin yayımcılardan kayıtçılara ulaştırılması arada bulunan dağıtım sisteminin görevidir. Kayıt\yayım haberleşme modeli kullanarak sadece dağıtılmış olay sistemleri değil farklı alanlarda çok sayıda yazılım sistemi de gerçekleştirilmiştir. (Sahingoz ve Erdogan 2003; Bornhövd vd., 2000; Skarmas ve Clark, 1999; Cugola vd., 2001).

Günümüze kadar geliştirilen tüm olay sistemleri olay kavramını, veri kaynağı / veri üreticisi tarafından üretilen basit formatta bir veri olarak algılamışlar ve çalışmalarını üretilen bu verinin, uygun formatta gösterilimi, işlenmesi ve hızlı şekilde olay tüketicilerine yayılması üzerine yoğunlaştırmışlardır. Bu çalışmanın amacı ise: olay kavramının üretim niteliğini geliştirerek, alışlagelen sadece ham veri üretiminin yanı sıra, eylemlerin (action) de üretildiği ileri bir düzeye yerleşmekte ve bu nitelikteki olayların üretilmesine, dağıtılmasına ve işlenmesine olanak sağlayan bir dağıtılmış olay sistemi altyapısı geliştirmektir. Bu düzeydeki bir olay, işlenebilir veriler üretebileceği gibi, kendine özgü bir kontrol akışına sahip olan bir eylemler dizisinin tetikleyicisi de olabilir. Söz konusu eylemler belirli bir amacı gerçeklemeye yönelik işlem dizisine, karar mekanizmalarına ve çoğunlukla birlikte üretildikleri bir veri kümesine sahip olacaktırlar.

Konu ile ilgili çalışmalar

Dağıtılmış olay sistemleri ile ilgili günümüze kadar yapılan çalışmaları kullanmış oldukları kayıt mekanizmalarına göre tarihsel gelişim sürecinde incelediğimizde beş ana başlıkta sınıflandırabiliriz.

Kanal tabanlı kayıt mekanizması

Dağıtılmış olay sistemlerinin ilk örnekleri kanal tabanlı kayıt mekanizmaları üzerine geliştirilmiştir. Burada *kanal* kavramı sistemdeki yayımcıların üretecekleri ve kayıtçıların da kayıt olacakları, olay verilerinin yazılacağı veya alınacağı hattı ifade etmektedir. Bir kayıtçı bir kanala kayıt olduktan sonra o kanala gelen tüm mesajlardan haberdar olur. Yayımcıda olay verisini üretip amacına uygun kanala göndererek ilgili kayıtçılara ulaşmasını sağlar. Java AWT gibi sistemler tarafından desteklenen bu modelde kanal adresleri önceden tanımlanmış olup yayımcıların ve kayıtçıların bunları bilerek kayıt ve yayım işlemleri yapmalarına olanak sağlanmaktadır (Sun, 1997).

Konu tabanlı kayıt mekanizması

Konu tabanlı kayıt mekanizmasını destekleyen sistemler, kanal tabanlı sistemin daha esnek ad-

reslemeye olanak sağlayan modeli olarak görülmektedir. Sistem kayıtçıların ve yayımcıların ortak *Mesaj Transfer Arabirimi* üzerinden olay verilerini transfer etmelerine olanak sağlamaktadır. Jedi, SwiftMQ, JMS 1.0.2 ve Steam tarafından desteklenen bu modelde olay verisi, olay mesajına çevrilirken, verinin konusu, olay mesajının başına yerleştirilmiş olup kalan kısım ise detay olarak mesajın sonunda bırakılmıştır (Cugola vd.; 1998; SwiftMQ, 2005; Sun, 2001; Meier ve Cahil, 2003). Olay sunucuları ve kayıtçılar sadece bu konuyu görebilir ve ona göre işlem yapabilirler. Mesajın detay kısmı, veriyi işleyecek kayıtçı tarafından açılarak incelenir.

İçerik tabanlı kayıt mekanizması

İçerik tabanlı kayıt modeli sayesinde olay sunucuları olay mesajının içeriği tamamıyla görülebilmekte ve hangi kayıtçının hangi veriyi alabileceği bu içeriğe göre filtrelenebilmektedir. Kayıtçılar ilgilendikleri olayları ifade ederken artık olayın konusundan değil, olayın özelliklerinden bahsetmektedirler. Olaylar sabit kriterlere göre sınıflandırılmamış, çalışma zamanında değişebilen parametrik değerlere göre de kayıt olma imkanı sağlanmıştır. Elvin, Siena, LeSubscribe, Gryphon, Rebeca, Dadi, ve Fuego gibi, çoğu dağıtılmış olay sistemlerinde olaylar basit veri tiplerinden oluşan niteliklerin bir kümesi olarak görülmektedir, (Segall ve Arnold, 1997; Carzaniga vd, 2001; Fabret vd, 2001; Strom vd., 1998; Mühl vd., 2002; Cao ve Singh, 2005; Tarkoma ve Kangasharju, 2005)

Tip-nesne tabanlı kayıt mekanizması

Tip tabanlı kayıt/yayım sistemlerinde üretilen olaylar uygulamalar tarafından tanımlanmış sınıfların örneği olacak birer nesne olarak modellenmiştir (Pietzuch ve Bacon, 2002; Eugster vd., 2001). Bu modelde kayıtçılar sisteme, kendi ilgi duydukları olay sınıfları üzerinden kayıt olurlar ve bu nesnelerin genel değişkenleri üzerinde kolaylıkla içerik tabanlı sorgulama ve filtreleme yapabilirler. Tip tabanlı olay sistemi, sistemin bileşenlerinde kullanılması için herhangi bir sabit olay tipi geliştirmede için yayımcılar tarafından üretilecek olay verilerinin de belirli bir şablona uygun olmasına gerek yoktur. Genellikle geliştirilen sistemlerde olaylar alt seviyede

mesajlar olarak makineler arasında dolaşırlarken, bu sistem sayesinde nesneye yönelik bir yaklaşımla, makineler arasında nesnelerin dolaşmasına, böylece daha esnek bir yapı oluşturulmasına olanak sağlanmaktadır.

Hibrid kayıt mekanizması

Bazı dağıtılmış olay sistemleri sistemin ölçeklenebilirliğini artırmak için farklı kayıt mekanizmalarını birlikte desteklemektedir. Corba Bildirim Sisteminde ve Jecho sisteminde olduğu gibi öncelikle konu tabanlı kayıt mekanizmasını destekleyen sistemler daha sonra diğer kayıt mekanizmaları da destekleyecek şekilde genişletilmişlerdir (OMG, 1996; Chen vd., 2003).

Bu tarihsel gelişim izlendiğinde nesnelerin kullanıldığı olay sistemlerinin bir ileri aşamasının etmenlerin kullanıldıkları olay sistemi olduğu değerlendirilerek yapılan çalışmada, gezgin etmenlerin birer olay verisi olarak kullanıldığı *Etmen Tabanlı Dağıtılmış Olay Sisteminin* (sistem kısaca İngilizce kısaltması kullanılarak ABDES sistemi olarak adlandırılmaktadır) gerçekleştirilmiştir.

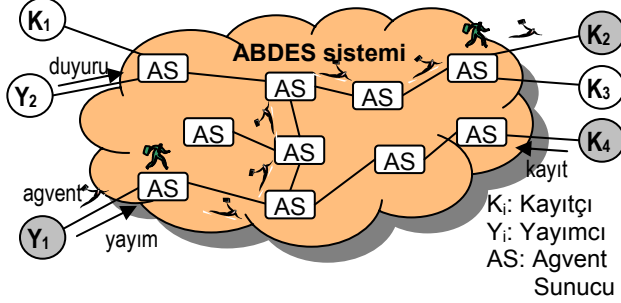
ABDES sistemi

ABDES sistemi, hem gezgin ve zeki etmen yapısının amaca yönelik davranma, sosyal yetenek ve tepkisellik özelliklerini, hem de kayıt/yayım modeliyle haberleşen olay tabanlı sistemlerin bir uçtan çok uca haberleşme, asenkron haberleşme ve gevşek bağlı sistemlerde kullanıma uygunluk özelliklerini birleştiren bir Etmen Tabanlı Dağıtılmış Olay Sistemidir. Sistem kayıtçı, yayımcı ve agent sunucu (AS) olarak adlandırılan üç ana bileşenden oluşmakta olup sistemin altyapısı Şekil 1’de gösterilmektedir.

ABDES sisteminin, daha önceleri geliştirilen dağıtılmış olay sistemlerinden üstün olan başlıca özellikleri şu ana başlıklar altında toplanabilir.

Otonomi: ABDES sisteminde olaylar basit bir olay verisi olarak görülmez ve sistemin temel elemanı, birinci sınıf üyesi olarak işlem yapılır. Üretilme aşamasında agentler birer gezgin etmen olarak üretilir ve bu aşamada, içine kendi amaçları, inançları (çevre bilgisi) ve davranışları

yüklenir. Bu gezgin etmen, bir AS'ya ulaştığı zaman, kontrolü yine AS'suna devretmemekte, AS'nun bilgi tabanını inceleyerek kendi hedef makinelerini otonom olarak seçmekte ve kendisini bu makinelere göç ettirmektedir.



Şekil 1. ABDES sisteminin altyapısı

Tip/agvent tabanlı kayıt: ABDES sisteminde sisteme dahil olan kayıtçılar ilgi duydukları olaylara, ilgili agventin tipini bildirerek kayıt olmaktadır. Olay bildirimlerinde aranan kriterler kayıtçının sisteme göndereceği kayıt mesajında ayrıntılı olarak yer almaktadır. Bu kriterler sayesinde sadece agventin özellikleri üzerinden değil, aynı zamanda davranışları üzerinden de filtreleme yapılmasına olanak sağlanmaktadır.

Bilgi saklama: ABDES sisteminde oluşan olaylar birer gezgin etmen yaratır ve bu etmen AS'lar arasında kendini yönlendirir. Bu nedenle bu etmenin taşıdığı değerlere dış kullanıcılar tarafından erişilmesi mümkün olmaz. Olay verisine ilgili etmenlerin kendi aralarında yürüttükleri kontrol işlemlerinden sonra erişilmesine olanak sağlanır (mesajlaşma ile). Bu sayede veri güvenliği sadece olay sunucusunun görevi olmaktan çıkar ve gönderilen agventin de güvenlik konusunda etkin bir şekilde rol alması sağlanır.

Kullanıcı tanımlamalı agvent tipleri: ABDES sisteminde yayımcılar, sistem tarafından önceden tanımlanmış olan agvent tiplerini kullanmak zorunda olmayıp, kendi olay-uygulama modellerine uygun yeni agvent tipleri tanımlayabilmektedir. Bu sayede sistemin esnekliği ve kullanılabilirliği artırılmaktadır.

Kendi kendini yönlendiren agventler: Agventin gezgin etmen yapısından dolayı her agvent ilgili

kayıtçılara ulaşması için izlemesi gereken yolu, AS'nun imkanlarından en alt seviyede faydalanacak şekilde belirler. Yönlendirme işlemlerinde fazla etkin olmayan AS sadece kendisine ulaşan agventlerin çalışması için uygun bir platform sağlamaktan ve kendine gelen kayıt bilgilerini saklayarak sunucu üzerinde çalışan agventlerin bu bilgi tabanından etkin şekilde faydalandırmaktan sorumlu olur.

Çift taraflı filtreleme: ABDES sistemi kayıtçı tarafından yapılan filtrelemeleri desteklemekle beraber, yayımcının ürettiği agventin de kendi filtrelemesini yapabilmesine olanak sağlar. Bu sayede yayımcı, ürettiği agventin belirli kriterleri sağlamayan kayıtçılara gönderilmesini engelleyebilir. Burada yönlendirme işlemi agventin kendisi tarafından yapıldığı için bu filtreleme işlemi de agvent tarafından kolaylıkla yapılmaktadır.

ABDES sisteminin bu özellikleri açısından incelendiğinde yukarıda belirtilmiş olan diğer dağıtılmış olay sistemleri ile karşılaştırılması Tablo 1'de gösterilmektedir. Bu temel özelliklerin yanı sıra geliştirilen sistem şu servis kalitelerine de sahiptir;

- Ölçeklenebilir yapısı sayesinde AS'ların sayısı artırılabilir ve daha fazla kayıtçıya ulaşılabilir.
- Kayıt kriterlerini kolay ve anlamlı olarak belirlemeye olanak sağlayan kayıt modeli sayesinde kayıtçıların kendi kriterlerini kolay ve anlaşılabilir olarak tanımlamasına olanak sağlanmaktadır.
- Kullanılan *Duyuru* ve *Kayıt* mesajları sayesinde sistem üzerinde dolaşan mesajların sayısı en aza indirilmektedir.
- Hata kotarma mekanizması sayesinde *Duyuru* ve *Kayıt* mesajlarının dağıtımı veya agventlerin yönlendirmeleri sırasında bir AS'nun çökmesi halinde alternatif yollar üzerinden hedef makineye ulaşım olanağı sağlanmaktadır.
- Bilgi tabanı yönetiminde nesne veri tabanları kullanılması sayesinde *Duyuru* ve *Kayıt* mesajlarının AS'lar üzerinde birer nesne olarak saklanmasına ve bu nesnelere hızlı şekilde erişilmesine olanak sağlanmaktadır.

Tablo 1. ABDES sisteminin diğer olay sistemleri ile karşılaştırılması

Olay Sistemi	Altyapı	Kayıt Modeli	Yönlendirme Mekanizması	Filtreleme	Bilgi Saklama	Kullanıcı Tanımlamalı Olay Tipleri	Otonomi
Yeast	Merkezi						
/Ready	/Dağıtılmış	İçerik	Sunucu	Tüketici	Yok	Yok	Yok
/Omninotify	/Dağıtılmış						
Java AWT	Merkezi	Kanal	Sunucu	Tüketici	Yok	Yok	Yok
Corba O.S.	Merkezi	Konu	Sunucu	Tüketici	Yok	Yok	Yok
Rebeca	Dağıtılmış	İçerik	Sunucu	Tüketici	Yok	Yok	Yok
LeSubscribe	Dağıtılmış	İçerik	Sunucu	Tüketici	Yok	Yok	Yok
Jedi	Dağıtılmış	Konu	Sunucu	Tüketici	Yok	Yok	Yok
Siena	Dağıtılmış	İçerik	Sunucu	Tüketici	Yok	Yok	Yok
Gryphon	Dağıtılmış	İçerik	Sunucu	Tüketici	Yok	Yok	Yok
Corba B.S	Merkezi	Konu + İçerik	Sunucu	Tüketici	Yok	Yok	Yok
Jecho	Dağıtılmış	Konu +Tip	Sunucu	Tüketici	Yok	Yok	Yok
Steam	Dağıtılmış	Konu	Sunucu	Tüketici	Yok	Yok	Yok
Fuego	Dağıtılmış	İçerik	Sunucu	Tüketici	Yok	Yok	Yok
Dadi	Dağıtılmış	İçerik	Sunucu	Tüketici	Yok	Yok	Yok
Hermes	Dağıtılmış	Tip	Sunucu	Tüketici	Yok	Yok	Yok
Obvent	Dağıtılmış	Tip	Sunucu	Tüketici	Var	Var	Yok
ABDES	Dağıtılmış	Tip (Agvent)	Agvent+ Sunucu	Üretici ve Tüketici	Var	Var	Var

Mesaj/agvent akışı

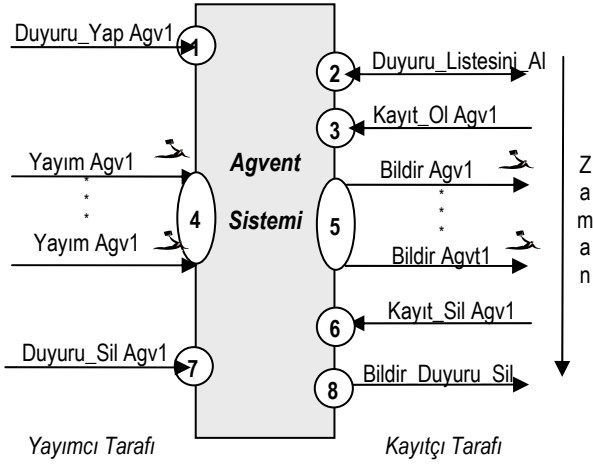
Sistemin bileşenleri (AS'lar, kayıtçılar ve yayımcılar) arasındaki haberleşme İnternet üzerinden Java RMI ile sağlanmaktadır. Yapısal tarafsızlık özelliğinden dolayı Java RMI, fiziksel olarak dağıtılmış olan heterojen makinelerde çalışabilmekte ve bileşenlere saydam bir ara yüz sağlayabilmektedir. Şekil 2'de gösterilen ABDES sisteminde kullanılan temel mesajlar ve bu mesajlara göre AS'sunun yaptığı işlemler şunlardır:

1. *Duyuru_Yap (advertise)*: ABDES sistemi, kullanıcı tanımlamalı agventlerin kullanımına da olanak sağlamaktadır. Yayımcı yeni bir agvent tipinin yayımına başlayacağı zaman, üreteceği agventin özelliklerini ve davranışlarını sisteme bildireceği zaman bunu sağlayan metod çağrısıdır. Bu sayede sisteme dahil olan kayıtçılar, ilgilendikleri agventin hangi özellikleri ve davranışları üzerine nasıl bir filtreleme yapabileceklerine karar verebilirler. AS, yayımcının *Duyuru_Yap* çağrısını alarak ilgili agventin bil-

gilerini, kayıt mesajlarının yönlendirilmelerinde kullanılmak üzere kendi duyuru tablosuna işler ve kendine doğrudan bağlı olan sunuculara da kendi duyuru tablolarını güncellemeleri için gönderir. Birden fazla yayımcının aynı agvent tipinde yayımı mümkündür.

2. *Duyuru_Listesini_Al (getAdvertisementList)*: Kayıtçı sisteme kayıt olurken hangi agventlerin yayımın yapıldığını öğrenmek için bu çağrıyı kullanır. *Duyuru_Listesini_Al* çağrısı ile kayıtçı kendi belirlediği kriterlere uygun olan duyuruları alır ve bu duyurulara göre agventlerin özelliklerini yorumlayabilir ve istediği agvent tipi üzerine kayıt olabilir.

3. *Kayıt_Ol (subscribe)*: Bir kayıtçı, bir agvent tipi ile ilgilenmesi durumunda üretilen bu agventin kendine yönlendirilmesini sağlamak için kendi kriterlerini *Kayıt_Ol* çağrısı ile sisteme gönderir. AS'su, kayıtçıdan gelen mesajı alır ve bu mesajı inceleyerek, kayıt olunmak istenen



Şekil 2. ABDES sisteminde kullanılan temel mesajlaşma modeli

agventin daha önce yayım yapılacağına duyurulup duyurulmadığını yani agvent kaydının var olup olmadığını kontrol eder. Eğer yayımı duyurulmuş bir agvent ise bu kayıtçının kriterlerini kendi kayıt tablosuna işler ve kendine bağlı ilgili AS'lara da işlemleri için iletir. Kayıt mesajı, duyuru mesajının dağıtım sırasında izlenen yolun tersi istikamette dağıtılır.

4. *Yayım (publish)*: Bir yayımcı daha önce duyurusunu yapmış olduğu bir agventi ürettiği zaman bunu olay sistemine *Yayım* çağrısı ile gönderir. Yayım çağrısı agventin dağıtımını başlatan ilk çağrıdır. ABDES sistemi kendine ulaşan agventin, istekçilerine kendisini yönlendirmesi için gerekli altyapıyı sağlamaktan sorumludur.

5. *Bildir (notify)*: Agvent, ABDES sistemi üzerinde hareket ettikten sonra, ulaşacağı kayıtçıya, kayıtçının *Bildir* metodu vasıtası ile gönderir. Bu agvent daha sonra görev yapısına uygun olarak kayıtçı tarafından aktif hale getirilerek çalışmasına uygun ortam sağlanır.

6. *Kayıt_Sil (unsubscribe)*: Kayıtçı daha önce ilgi duyduğunu belirttiği bir agvente artık ilgi duymuyor, üretilen bu agventten haberdar edilmek istemiyor ise, AS'suna *Kayıt_Sil* çağrısı ile bir mesaj gönderir. Sunucu bu mesajı alınca kendi kayıt tablosundan kayıt verisini siler ve kendine bağlı ilgili AS'larından da bu kaydın silinmesini talep eder.

7. *Duyuru_Sil (unadvertise)*: Daha önceden yayımı yapılacağı duyurulan bir agventin artık yayımının yapılmayacağına ABDES sistemine bildirilmesidir. Bu mesajı alan sunucu, ilgili agvent kaydını, kendi duyuru tablosundan çıkarır ve kendisi ile bağlantılı sunuculara da aynı işlemi yapması için gönderir. Böylece yayımı yapılmayacak agventler üzerine kayıtçıların kayıt olması engellenir.

8. *Bildir_Duyuru_Sil (notifyUnadvertise)*: Sisteme yayımlanacağı duyurulan bir agvent tipinin artık yayımının yapılmayacağı *Duyuru_Sil* ile bildirilmiş ve bu agventi yayımlayacağını duyuran başka yayımcı da kalmamış ise, bu agvent tipine ilgilerini bildirmiş olan kayıtçıların, bu durumdan haberdar edilmeleri gerekir. Kayıtçıya bu bilginin iletilmesi *Bildir_Duyuru_Sil* metodunun çağrılması ile yapılır.

Agventler

Agvent; yayımcılar tarafından üretilen olay verisini kayıtçılara ulaştırmak üzere ABDES sistemi üzerinde göç ederek hareket eden ve bağlantı düğümlerinde aktif hale getirilerek göç edeceği hedef makineyi/makineleri seçip kendini otonom olarak bu makinelere yönlendiren kod ve veri bütünüdür.

```

class PublicationAgvent extends Agvent
{
    private String Author;
    private String Name;
    private int Pages;
    private float[] Dimensions;
    private String Publisher;
    private Date Publish;
    private int ISBN;
    private float ListPrice;

    private boolean ReferenceContains(String AuthorName){...}
    private boolean TOCContains(String topic){...}
    private float WholesalePrice(int amount, String dest){...}
    .....
}
    
```

Şekil 3. Bir agvent örneği

Agventler yayımcılar tarafından üretilerek kayıtçılara ulaşmaları için herhangi bir topoloji ile birleştirilmiş AS'larından oluşan ABDES sistemine gönderilirler. Her düğümde yer alan bir AS'su, kendine ulaşan agventlerin çalışması,

kendi yaratılma amacına uygun işlevler yapabilmesi ve ihtiyaç duyulması durumunda kendisini (veya kopyasını) hedef bir makineye göndermesi için uygun bir çalışma platformu oluşturur. Geliştirilen bir agvent kodu örneği Şekil 3'te gösterilmektedir.

ABDES sisteminde filtreleme

ABDES sistemini diğer dağıtılmış olay sistemlerinden ayıran temel özelliklerden biri de kullanılmış olduğu filtreleme mekanizmasıdır. Dağıtılmış olay sistemlerinin bir amacında, sisteme gönderilen olay verisini, sadece bu olay verisi ile ilgilenen kayıtçılara ulaştırılmasını sağlayarak sistemdeki veri akışını en düşük seviyede tutmaktır. Bunun için de filtrelemeye olanak sağlayan kriterler/profiller oluşturularak sisteme gönderilmektedir. Günümüze kadar geliştirilen olay sistemleri, olay verisinin dağıtımında sadece kayıtçının kriter belirlemesine olanak sağlamaktadır. ABDES sisteminde ise, kullanılan gezgin etmen yapısından ve agvente yüklenen otonom özelliğinden dolayı olay verisini (agventi) üreten yayımcının da agvente belirli kriterler yüklemesi imkanı vardır. Bu nedenle ABDES sistemi hem *Kayıtçının* hem de *Yayımcının* filtreleme yapmasına olanak sağlamaktadır.

Kayıtçının filtrelemesi

Kayıtçı bir agvent ile ilgilendiği zaman, ilgilendiği agventin özelliklerini bir kayıt mesajı ile ABDES sistemine bildirir. ABDES sistemi gönderilen bu kayıt mesajını daha önceden sisteme ilan edilen duyuruların oluşturmuş olduğu dağıtım ağaç yapılarına göre AS'larına dağıtılmasını sağlar. Bir agvent sisteme yayımlandığı zaman, ulaştığı ilk AS'da kendisi ile ilgilendiğini beyan eden kayıtçıların bu kriterlerini inceleyerek filtreleme kriterlerine uyan kayıtçıları belirler ve kendisini bu kayıtçılara doğru yönlendirir.

Yayımcının filtrelemesi

Bir yayımcı ürettiği agvent üzerine yönlendireceği kayıtçıların seçimi için iki şekilde filtre koyabilmektedir.

- *Kayıtçının tanımlayıcısı (adı veya adresi) üzerinden filtreleme:* Yayımcı daha önce dağıt-

mış olduğu agventlerle ilgili verileri saklamış olduğu geçmiş tecrübelerini içeren bir bilgi tabanı vardır. Yayımcı bu bilgi tabanından faydalanarak belirli kayıtçılara bu agventin ulaştırılmasını istemiyorsa, bu kayıtçıları bir yasak listesine koyarak agvente tanımlarlar. Agvent kendi yönlendirmesini yaparken, kayıtçının kriterlerini sağlamış olsa dahi kendisini bu listedeki kayıtçılara doğru yönlendirmez. Bu filtreleme agventin ulaştığı ilk AS'su üzerinde yapılır.

- *Kayıtçının özellikleri üzerinden yapılan filtreleme:* Her kayıtçının bağlı olduğu AS'su üzerinde kendisi ile ilgili "çalışan sayısı, net gelir, kısa vadeli borç miktarı, uzun vadeli borç miktarı, toplam mal varlığı, yatırım miktarı, aktiflerinin değeri, pasiflerinin değeri" gibi istatistik bilgileri tutulmakla birlikte aynı zamanda geliştirilen agventlerin de yayımlanmasında faydalanılmak üzere kayıtçının bilgisayarının performansını gösteren "CPU hızı, bellek miktarı, hard diskteki boş alan" gibi değişik özellikleri de bulunmaktadır. Bu bilgiler kayıtçının özel bilgileri olduğu için tüm AS'lara dağıtılmaz, sadece bağlı bulunduğu AS üzerinde tutulur. Yayımcı, kayıtçının bu özelliklerini de kontrol ederek bir filtreleme yapmak istemesi durumunda kendi profilini tanımlayarak bir sorgu katarı olarak agventin içine yerleştirir. Agvent kayıtçının bağlı bulunduğu AS'ya ulaştığında bu kriterleri AS'nun bilgi tabanından kontrol eder ve kendisini bu kayıtçıya gönderip göndermemeye karar verir.

Test sonuçları

Dağıtılmış sistemlerin performans ölçümlerinin değerlendirilmesi geliştirilen uygulamanın geniş bir alan ağı üzerinde çalıştırılmasıyla veya bir simülasyon üzerinde testlerin yapılmasıyla sağlanabilir. Her ne kadar doğrudan uygulamanın çalıştırılmasıyla alınan ölçümler amaçlanan değerleri verse de büyük ölçekli dağıtılmış sistemlerin kurulması, gerekli sistem bileşenlerin eklenmesi ve ağ genişliklerinin ayarlanması çok zordur. Bu nedenle ABDES sisteminin performans testlerinin yapılması için bir simülasyon geliştirilmiştir. Büyük ölçekli bir sistemin simülasyonunun yapılmasında gerçekçi verileri al-

mak ve uygun ağ topolojilerini oluşturarak gerek duyuru ve kayıt mesajlarının gerekse agentlerin dağıtım sürelerinin hesaplanması amaçlanmıştır.

Dağıtım sürelerinin hesaplanmasında geliştirilen sistemdeki yazılımsal kriterlerin haricinde *Atlama Miktarı*, *Bant Genişliği* ve *Makine Performansı* kriterleri de önemlidir. ABDES sistemi için yapılan başarımlar ölçümleri özellikleri Windows NT işletim sisteminin çalıştığı 100 Mbit/s lik bant genişliğindeki bir ağ yapısı üzerinde gerçekleştirilmiştir. Sistemin geniş alan üzerine yayılacağı göz önüne alınarak üç farklı bilgisayar konfigürasyonu üzerindeki gerekli ölçümler alınarak hesaplamalar sırasında bunların ortalaması alınmıştır. Sistem üzerinde gönderilen duyuru mesajları, kayıt mesajları ve agentlerin dağıtım işlemlerinin test edilmesindeki temel işlemler AS'lar üzerinde yapılmaktadır. Bu nedenle test ortamındaki AS'ların sahip oldukları özellikler Tablo 2'de gösterilmektedir. Sistem 10 farklı topoloji üzerinde test edilmiştir. Bu topolojilerin özellikleri Tablo 3'te gösterilmektedir.

Tablo 2. Dağıtım sürelerinin hesaplanmasında kullanılacak kriterler

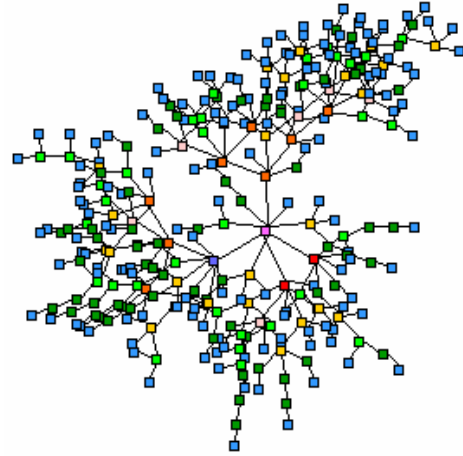
Nitelik	Değer aralığı
Duyuru Sayısı	5.000-15.000
Kayıt Sayısı	10.000-30.000
Komşu AS Sayısı	1-7
Bir AS'ya Kayıtlı Kayıtçı Sayısı	5-100
Sistemdeki AS Sayısı	20-100
Alternatif Yönlendirme Tablosu (AYT) Kayıt Sayısı	100-1000
Kayıtların Kriter Miktarları	
Özellik Sayısı	0-15
Metod (Davranış) Sayısı	0-10
Metoda Gönderilen Parametre Sayısı	0-5

Bu topolojileri oluşturmak amacıyla Boston Üniversitesi tarafından geliştirilen BRITE: Boston University Representative Internet Topology generator (Medina vd., 2001) sisteminden fay-

dalanılmıştır. Bu sistem ile üretilen 300 AS'ya sahip olan bir topolojinin örneği de Şekil 4'te gösterilmiştir.

Tablo 3. Test için kullanılan topolojiler

AS Sayısı	Ortalama Atlama Miktarı	Minimum Atlama Miktarı	Maksimum Atlama Miktarı
100	11.73	9.03	14.09
200	14.41	12.87	15.30
300	16.10	14.08	17.93
400	17.47	14.80	20.32
500	17.57	15.16	19.09
600	18.59	17.14	20.48
700	18.64	15.72	23.08
800	19.59	16.69	22.47
900	20.39	16.47	23.37
1000	20.58	17.75	23.86



Şekil 4. 300 AS'ya sahip bir topoloji örneği

Duyuru mesajlarının dağıtım işlemleri

ABDES sisteminde üretilen bir duyuru mesajının sistem üzerindeki tüm AS'lara dağıtılması gerekmektedir. Bu dağıtım yayın-broadcast mekanizması ile gerçekleştirilir. Duyuruların dağıtım sırasında bir AS üzerinde yapılan işlemler ve süreleri Tablo 4'te gösterilmektedir.

Kayıt mesajlarının dağıtım işlemleri

Kayıtçı tarafından üretilen bir kayıt mesajı, sistem üzerinde dağıtım sırasında ilgili duyuru mesajlarının çizdiği yoldan faydalanmaktadır. . Kayıt mesajları duyuru mesajlarının çizdiği ro-

tanın tam tersi istikamette sistem üzerinde dağıtılmaktadır. Duyuru mesajların kullanımı sayesinde kayıt mesajlarının dağıtımı hem hızlanmakta hem de gereksiz sayıda kayıt mesajı üretilmeyerek sistemdeki mesaj yoğunluğu azaltılmaktadır. Kayıt mesajlarının sistem üzerinde dağıtımı sırasında AS üzerinde yapılan işlemler ve süreleri Tablo 5’te gösterilmektedir.

Tablo 4. Duyuru mesajının dağıtımı sırasında agvent sunucu üzerinde yapılan işlemler

İşlem	Süre (ms)
Duyuru Sorgulama	4.95
Dosya Erişim İşlemleri	25.10
Duyuru Ekleme	27.32
Komşu AS Bilgilerin Erişim	4.60
RMI Bağlantısı Kurma	4.80
Duyurunun Gönderilmesi	4.31
AYT’nu Güncelleme	4.74

Tablo 5. Kayıt mesajının dağıtımı sırasında agvent sunucu üzerinde yapılan işlemler

İşlem	Süre (ms)
Kullanıcı Adı ve Şifre Kontrolü	0.92
Kayıt Tablosu Kontrolü	17.89
Kayıt Tablosuna Ekleme	55.53
Dağıtım Listesinin Seçimi için Duyuru Tablosundan Sorgulama	4.95
Dağıtım Listesini Oluşturma	2.00
RMI Bağlantısı Kurulumu	4.80
Kayıt Mesajının Gönderilmesi	8.91
Yönlendirme Tablosunu Güncelleme	4.74

Agvent dağıtım işlemleri

Agventler yapıları gereği içlerinde değerli verileri saklamaktadırlar. Kullanılan agvent modeli/örneğine göre agventin boyutu farklılıklar göstermektedir. Bu da agventin dağıtım süresini

etkilemektedir. Sistemin test edilmesi için geliştirilen örnek uygulamada (Şahingöz ve Sönmez, 2006) agvent yapısı Şekil 2’de gösterilmiş olup bu agvent tipi özellikle akademik yayınların dağıtımında kullanılmak üzere geliştirilmiştir. Sistem aynı zamanda normal köşe yazıları, dergi yazıları ve kitaplar için de kullanılabilir. Agventlerin boyutunu etkileyen temel faktör içerdiği veridir. PublicationAgvent örneğinin içerdiği yayındaki kelime ve karakter sayısına göre yaklaşık boyutu Tablo 6’da gösterilmektedir.

Tablo 6. PublicationAgvent boyutu tablosu

Kelime Sayısı	Karakter Sayısı	Sayfa Sayısı	Agvent Boyutu (Kb)
641	4301	Köşe Yazısı	4.60
1943	12658	Araştırma Yazısı(Dergi)	13.93
3116	17260	8 Sayfa Bildiri	22.35
4139	22706	10 Sayfa Bildiri	29.68
7504	40660	12 Sayfa Bildiri	53.81
44650	318920	Doktora Tezi	320.17
237300	1308750	50 Bildirili Sempozyum Kitabı	1700.00
325070	1600654	870 Sayfa Kitap	2330.00

Agventler yapıları gereği sistemin bağlantı düğümlerinde ve kayıtçılarda aktif hale gelmektedirler. Agventlerin her AS üzerinde yapmış oldukları işlemler farklılık göstermektedir. Agventlerin aktif hale geçemedikleri düğümlerde agventler paket olarak sunucu tarafından bir sonraki AS’ya doğru yönlendirilmektedir. Bu nedenle AS’ları ilk agvent sunucu (ias), aradaki agvent sunucu (aas) ve son agvent sunucu (sas) olarak tanımladığımızda bir agventin AS üzerinde yaptığı işlemler ve süreleri Tablo 7’de gösterilmiştir.

Bu işlem süreleri göz önüne alındığında Tablo 3’te gösterilen on farklı test topolojilerine sahip ABDES sistemleri üzerinde agventlerin ortalama

ma dağıtım süreleri hesaplanmıştır. Bu sürenin hesaplanmasında sitemdeki her AS'nun bir agvent üretmesi ve bunu sistemdeki AS'ların hepsine ulaşacak şekilde dağıtım süreleri hesaplanmış ve bu değerlerin ortalaması alınmıştır. Bu topolojilerdeki ortalama agvent dağıtım süreleri Şekil 5'te gösterilmektedir

Tablo 7. Agventin dağıtımı sırasında agvent sunucu üzerinde yapılan işlemler

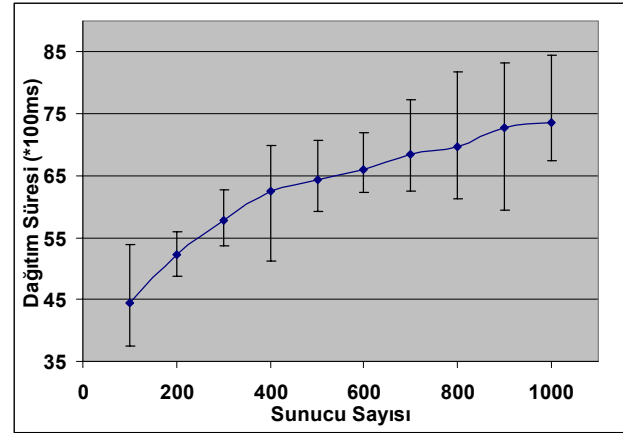
İşlemler	Dolaşan Agv. Sun.	Süre (ms)
Kuyruktan Agvent Alınması	ias,aas,sas	11.1
Agvent Aktif hale getirilmesi	ias,sas	2.3
runFirstAgventServer'in çalışması	ias	2.5
runLastAgventServer'in çalışması	sas	2.0
Kayıt Tablosundan sorgulama (ort 10000 Kayıt için)	ias	4.95
Bir kayıt'ın özellik kriterlerinin incelenmesi (ort 5 özellik)	ias	1.3
Bir kaydın davranış kriterlerinin incelenmesi (ort 4 metod)	ias	2.2
Kayıtçı Tablosunun İncelenmesi (ort 100 kayıtçı)	sas	0.92
AYT Güncelleme	ias,aas,sas	5.3
Dağıtım Kuyruğunda Bekleme	ias,aas,sas	29.2
RMI Bağlantısı Kurulması	ias,aas,sas	4.8
Agventin Gönderilimi (ort 750 Kb)	ias,aas,sas	85.9

Sonuç ve öneriler

ABDES sistemi, hem etmen yapısının özelliklerini, hem de kayıt/yayım modeliyle haberleşen olay tabanlı sistemlerin özelliklerini birleştiren bir *Etmen Tabanlı Dağıtılmış Olay Sistemidir*. Sistem nitelikli olay verilerinin üretilmesini ve dağıtılmasını amaçlamış olup, bu olay verileri (agventler) sistem üzerinde kendi otonomileri ile dolaşmalarına olanak sağlayan gezgin etmen yapısında geliştirilmiştir.

ABDES sistemi performansı açısından değerlendirildiğinde, önceden geliştirilen dağıtılmış

olay sistemleri olay verilerini basit formatta tanımlandıkları için sistem üzerinde hızlı şekilde dağıtılmalarına olanak sağlandığı ve bunun performanslarını daha da artırdığı görülmektedir. ABDES sisteminde bilgi saklamanın ön planda olduğu uygulamalarda yayımcı için gizlenmesi gereken ve büyük boyutlu veriler agventle beraber yayımlanmaktadır. Uygulama örneğine bağlı olarak agventin boyutu da büyümektedir. Agventin boyutunun artması ise üretilen agventin kayıtçılara ulaştırılma süresini kaçınılmaz şekilde artırmaktadır. Ancak sisteminin kullanım alanları açısından incelendiğinde hesaplanan dağıtım sürelerinin kabul edilebilir boyutlarda olduğu değerlendirilmektedir.



Şekil 5. Test topolojilerinde ortalama agvent dağıtım süreleri

Kaynaklar

- Bornhövd, C., Cilia, M., Liebig, C., ve Buchmann, A., (2000). An infrastructure for meta-auctions. *In Proceedings of 2nd International Workshop on Advance Issues of E-Commerce and Web-based Information Systems*, California, 21-30.
- Cao, F. ve Singh, J.P., (2005). MEDYM: Match-early and dynamic multicast for content based publish-subscribe service networks. *In Proceedings of Fourth International Workshop on Distributed Event-Based Systems (DEBS) (ICDCSW'05)* Ohio, 370-376.
- Carzaniga, A., Rosenblum, D. ve Wolf, A., (2001). Design and Evaluation of a Wide Area Event Notification Service. *ACM Transactions on Computer Systems*, **19**, 3, 332-383.
- Chen, Y., Schwan, K., ve Zhou, D., (2003). Opportunistic channels: Mobility aware event delivery,

- In Proceedings Middleware 2003, ACM/IFIP/USENIX International Middleware Conference*, Rio de Janeiro, 182–201,
- Cugola, G., Di Nitto, E. ve Fuggetta, A., (1998). Exploiting an event-based infrastructure to develop complex distributed systems, *In Proceedings of the 20th International Conference Software Engineering*, Kyoto, 261- 270.
- Cugola, G., Di Nitto, E., ve Fuggetta, A., (2001). The JEDI event-based infrastructure and its application to the development of the OPSS WFMS. *IEEE Transactions on Software Engineering*, **27**, 9, 827-850.
- Eugster, P.T., Guerraoui, R., ve Damm, C.H., (2001). On Objects and Events, *In Proceedings of the 16th ACM SIGPLAN Conference on Object Oriented Programming, Systems, Languages, and Applications*, Florida, 254-269.
- Medina, A., Lakhina, A., Matta, I., ve Byers, J., (2001). BRITE: An approach to universal topology generation, *In Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, Ohio, 346-356.
- Meier, R., (2000). State of the Art Review of Distributed Event Models, Teknik Rapor TCD-CS-2000-16, Trinity College, Dublin.
- Meier, R. ve Cahil, V., (2003). Exploiting proximity in event-based middleware for collaborative mobile applications, *In Proceedings of 4th International Conference on Distributed Applications and Interoperable Systems*, Paris, 285–296.
- Mühl, G., (2002). Large-scale content-based publish/subscribe systems, Doktora Tezi, *Darmstadt University of Technology*, Darmstadt, Almanya.
- Mühl, G., Fiege, L., ve Buchmann, A. P., (2002). Filter similarities in content-based publish/ subscribe systems, *In Proceedings of the ARCS International Conference on Architecture of Computing Systems*, Karlsruhe, Almanya, 244-240.
- Fabret, F., Jacobsen, H., Llibat, F., Pereira, J., Ross, K.A. ve Shasha, D., (2001). Filtering algorithms and implementation for very fast publish/subscribe. *In Proceedings of the ACM SIGMOD International Conference on Management of Data*. California, 115-126.
- Pietzuch, P. ve Bacon, J., (2002). Hermes: A distributed event based middleware architecture. *In Proceedings of the 1st International Workshop on Distributed Event-Based Systems*, Viyana, 611-618.
- Sahingoz, O.K. ve Erdogan, N., (2003). A two-leveled mobile agent system for electronic commerce, *The Journal of Aeronautics and Space Technologies Institute (ASTIN)*, **2**, 1, 21-32.
- Sahingoz, O.K. ve Sonmez, A.C., (2006). Mobile agent based publication alerting system, Springer Verlag **3993**, 903 – 907.
- Segall, B., ve Arnold, D., (1997). Elvin has left the building: A publish/subscribe notification service with quenching, *In Proceedings of the AUUG Australian UNIX and Open Systems User Group Conference*, Avustralya, 243-255.
- Skarmas, N. ve Clark, K., (1999). Content-based routing as the basis for intra-agent communication. *Springer Verlag*, **1555**, 345-362.
- Strom, R., Banavar, G., ve Chandra, T., (1998). Gryphon: An information flow based approach to message brokering. *In Proceedings of Symposium on Software Reliability Engineering*, Almanya, 10-21.
- Tarkoma S. ve Kangasharju, J., (2005). Filter merging for efficient information dissemination, *Springer-Verlag*, **3760**, 274-291.
-
- OMG Notification Service, (1996). CORBA services: Notification Service Specification -Request For Proposal. Object Management Group. http://www.omg.org/technology/documents/formal/notification_service.htm, (28.5.2006).
- SwiftMQ, (2005). JMS enterprise messaging platform, <http://www.swiftmq.com/>, (28.5.2006).
- Sun Microsystems Inc., (1997). Java AWT: Delegation Event Model. <http://java.sun.com/j2se/1.3/docs/guide/awt/designspec/events.html>, (28.5.2006).
- Sun Microsystems Inc., (2005). Java Message Service, <http://java.sun.com/products/jms/docs.html>, (28.5.2006).