

Proses Otomasyonunda OPC Teknolojisi

Serdar Büyük¹, Salih Gök²

ÖZET

Endüstriyel otomasyonda kullanılan haberleşme sistemlerinin ve protokollerinin markadan markaya farklılık göstermesi, bu alanda bir standartlaşmaya gitme ihtiyacı doğurmuştur. Bu nedenle OLE for Process Control (OPC) - Süreç Yönetiminde Nesnelerin Bağdaştırılması ve İlişkilendirilmesi standartları ortaya çıkmış ve farklı kontrolörlerin haberleşmesi için sunucu-istemci mimarisi temel alınarak geliştirilmiştir. OPC teknolojisi günümüzde birçok işletmede çok hassas proseslerde dahi kullanılabilir.

1. ENDÜSTRİYEL OTOMASYONDA HABERLEŞME

Endüstriyel otomasyon alanında çalışanlar için farklı marka cihazlardan tek bir ortama veri aktarılması ya da cihazların birbiriyle haberleşmesi çoğu zaman bir engel olmuştur. Cihaz/sistem üreticisi firmaların endüstriyel haberleşmede kendilerine özgü protokoller kullanmaları bu durumun başlıca sebebidir ve hem üreticiler hem de son kullanıcılar bundan olumsuz etkilenmektedir. OPC (OLE for Process Control) kavramı bu alandaki engelleri ortadan kaldırmak için endüstriyel otomasyon sistemleri ve endüstriyel haberleşme sistemleri üreticileri ile otomasyon firmalarının bir araya gelmesiyle ortaya çıkmıştır.

OPC, Microsoft'un Object Linking & Embedding (OLE)/ Component Object Model (COM) standardına dayanır. OLE/COM, Microsoft'un farklı uygulamaları arası bütünleşmeyi hedefleyen nesne tabanlı bir teknolojisidir. OPC ise OLE tabanlı bir haberleşme standardıdır ve farklı otomasyon seviyeleri arasında hızlı ve güvenilir bir bağlantı sağlar. OLE teknolojisinin kullanımıyla OPC, ister yönetim katındaki bir uygulama olsun isterse proses denetiminde kullanılan bir uygulama olsun, farklı uygulamalar arası veri alışverişinin, tanımladığı yöntem ve nesnelerle, standart bir şekilde gerçekleşmesini sağlar [1]. OLE haberleşmenin altyapısını hazırlarken, OPC ile bu temellerin üzerine veri akışının hızının belirlenmesi ve zaman bilgisinin de veriye eklenmesi gibi otomasyon odaklı özellikler eklenir [2].

Endüstriden örnekler vererek OPC'nin çalışma şeklini açıklamak daha anlaşılır olacaktır:

Örneğin bir Programmable Logic Controller (PLC), Distributed Control Systems (DCS) ya da diğer türde bir kontrolörün içindeki verinin, cihaz kontrolü ve veri izleme için ara yüz işlevi gören Supervisory Control and Data Acquisition (SCADA) yazılımlarında okunması ve kontrolöre verinin yazılması için kontrolörü SCADA'ya tanıtan bir sürücü gereklidir. Her kontrolör üreticisinin haberleşme protokolü farklı olduğundan her SCADA yazılımına özel, her bir kontrolör markasının

sürücüsü tasarlanmalıdır. Bu ise emek, zaman ve maddi külfet açısından hem çok kaynak tüketir hem de uygulama safhasında birçok zorluk çıkarır.

Ürüne özel sürücü geliştirmeye dayanan bu tür çözümler şu sorunları ortaya çıkarmıştır:

- Aynı çalışmanın defalarca tekrarı: Her yazılım geliştirici kendi yazılımı için aynı cihazla haberleşmek için bir sürücü yazmak zorundadır.
- Sürücüler arası uyumsuzluk: Son kullanıcı, bir programda sahip olduğu bir özelliği aynı donanımla haberleşen diğer bir programda bulamayabilir. Zira sürücü geliştirenler kendi programlarıyla direkt olarak ilgili olmayan bazı donanım özelliklerini desteklemeyebilirler.
- Gelecekte donanımda olacak değişikliklere destek: Geliştirilen sürücüler donanım firmalarından bağımsız olarak üretildiklerinden, aynı markanın yeni nesil bir ürünü önceki donanım için yazılmış sürücüyle haberleşmeyebilir.
- Erişim problemi: Farklı iki uygulama farklı sürücüler kullandıklarından dolayı çoğu zaman aynı anda aynı cihaza erişemezler.

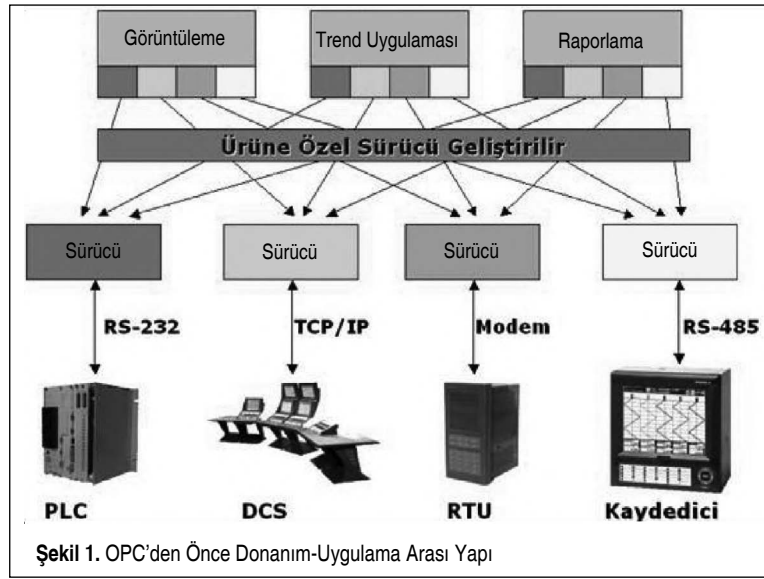
Şekil 1'de gösterilen örnek bir otomasyon sistemi uygulamasında, OPC'den önce her uygulama yazılımı ürüne özel sürücü geliştirilmesi ve bunun yol açtığı sorunlar ele alınmıştır.

Bunun yerine tüm kontrolörlerin stan-

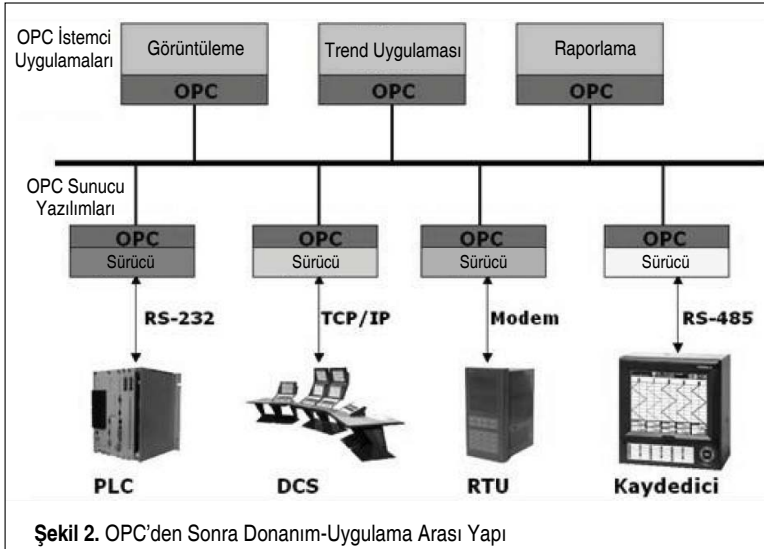
¹ ASP Otomasyon San. ve Tic. Ltd. Şti., İstanbul - serdar@aspltd.net

² ASP Otomasyon San. ve Tic. Ltd. Şti., İstanbul - salih@aspltd.net

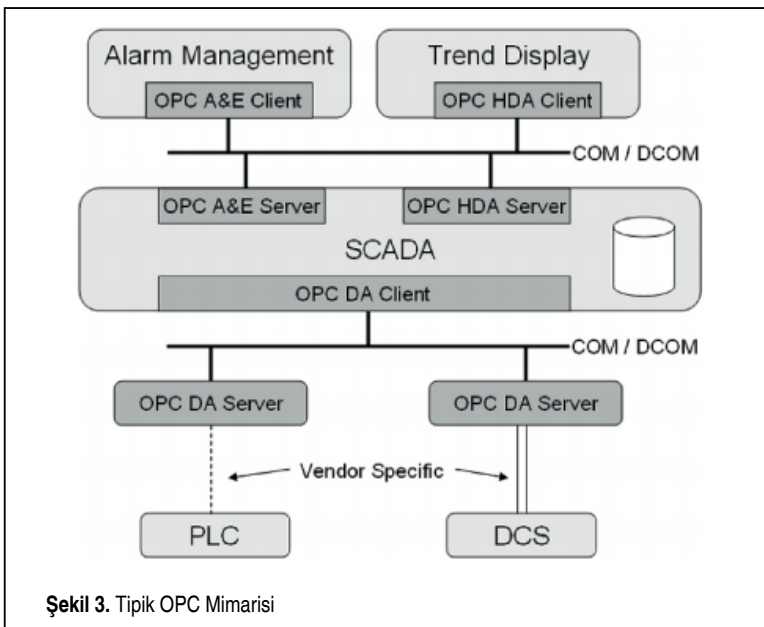
* Bu yazı, 21-22 Ekim 2011 tarihlerinde Makina Mühendisleri Odası tarafından İstanbul'da düzenlenen Endüstriyel Otomasyon Sempozyumu'nda bildiri olarak sunulmuştur.



Şekil 1. OPC'den Önce Donanım-Uygulama Arası Yapı



Şekil 2. OPC'den Sonra Donanım-Uygulama Arası Yapı



Şekil 3. Tipik OPC Mimarisi

dart bir haberleşme uygulamasıyla veri alışverişi yapması daha uygundur. OPC tanımladığı standart arayüzle donanım üreticileri ve yazılım sağlayıcıları arasında belirgin bir çizgi çizmiştir. Artık donanım üreticileri sadece tek bir sürücü geliştirerek –ki bu OPC Sunucu yazılımıdır, piyasada bulunan tüm OPC istemci özelliğine sahip yazılımlarla haberleşebilme özelliğine kavuşmuş olmaktadır. Böylece kaynak ve zamandan tasarruf ve uygulamada birçok kolaylık sağlanabilir.

Şekil 2’de OPC kullanımıyla sistemde kullanılan tüm yazılım ve donanımlar için tek ve standart bir arayüz olan OPC arayüzü kullanılarak, ürüne özel sürücülerin tüm dezavantajlarından kurtulmuş olunmaktadır.

2. OPC MİMARİSİ

2.1 Sunucu – İstemci Yapısı

OPC, veri alışverişinde istemci-sunucu yapısını kullanır. Bir OPC sunucusu proses bilgisinin kaynağı olan cihazdan verileri alır ve istemci arayüzü üzerinden kullanıcıya iletir. OPC uyumlu bir istemci, sunucuya bağlanır ve talep ettiği veriye ulaşır [3]. Sistem yapısı genel olarak aşağıdaki gibidir.

OPC’nin sağladığı standart arayüz dışında, İstemci-Sunucu (Client-Server) yapısındaki çalışma prensibi donanıma olan erişim problemlerinin önüne geçmiştir. OPC istemcisi olan bir yazılım sahip olduğu standart ara-yüzle aynı anda aynı arayüzü kullanarak farklı markalara ait OPC Sunucu programlarına bağlanabilir ve o programlar aracılığıyla fiziksel donanımla veri haberleşmesinde bulunabilir. OPC sunucusu, donanıma bağlanarak kullanıcı tarafından belirlenebilecek zaman aralıklarıyla, son değerlerin sorgulamasını yapar ve elindeki verileri günceller. OPC uyumlu istemci ise, sunucuya sorgulamalar göndererek verileri sunucudan istediği sıklıkla çeker.

2.2 OPC Standartları

Zamanla, OPC bir standartlar bütünü haline almıştır. Bu standartlardan başlıca olanlar şöyle sıralanabilir:

OPC Data Access (OPC DA), OPC Historical Data Access (OPC HDA), OPC Alarm & Events (OPC A&E), OPC Extensible Markup Language (OPC XML) ve OPC Unified Architecture (OPC UA).

OPC DA gerçek zamanlı veri akışını, OPC HDA geçmişe yönelik veri kaydının aktarımını, OPC A&E alarm ve olaya yönelik standardı, OPC XML. NET gibi ortamların OPC ile uyumlu arayüzler geliştirmesini düzenler. OPC UA ise son yayımlanan standart olup, OPC teknolojisinin daha güvenli kullanılabilir olmasını düzenleyen standarttır [4].

3. ENDÜSTRİDE OPC UYGULAMALARI

3.1 Kağıt Fabrikasında OPC Uygulaması

OPC teknolojisinin endüstride en yaygın örneği veri izlemedeki kolaylık olarak öne çıkar. Örneğin, Yalova’da bulunan İpek Kâğıt Fabrikası’nın Biyolojik Arıtma tesisinde gerçekleştirilen uygulamada ABB Advant Operatör Kontrol İstasyonunda (OCS) Siemens PLC’den gelen verilerin de izlenmesi istendi. Talebin klasik yöntemlerle çözülmesi Siemens PLC’den ABB operatör istasyonuna Profibus bağlantısı yapılmasıyla mümkündür. Üstelik ABB kontrolörünün DCS olması nedeniyle, Profibus hattına yapılacak bir ilave için DCS’in, durdurulup başlatılması, dolayısıyla kâğıt makinesinin durması ve hem ABB hem de Siemens kontrolör yazılımlarına müdahale edilmesi gerekliydi. Bu ise kâğıt prosesi açısından önemli bir maliyet ve zaman kaybı demektir. Bu sorunlardan kaçınmak için Kepware OPC sunucusu ile Siemens PLC’lerdeki veriler Ethernet üzerinden ABB kontrolörünün kurulu

olduğu bilgisayara aktarıldı ve Link-Master OPC köprüleme yazılımıyla ABB ve Kepware OPC sunucuları arasında veri alışverişi sağlandı. Böylece hem tesis durmamış oldu ve hem de iki kontrolörün yazılımlarına müdahaleye gerek kalmadı. Üstelik proje maliyeti klasik yöntemle orana yarıya indi. Sonuçta ABB Advant OCS üzerinde hem ABB kontrolöründen hem de Siemens PLC’lerden gelen veriler izlenebilir hale geldi.

3.2 Otomotivde OPC Uygulaması

Bir başka OPC uygulaması örneği ABD’de bir otomobil üreticisinin tedarik zincirini daha pratik bir hale getirme çabası sonucunda geliştirildi. Bu uygulamada üreticinin PLC’lerine hangi malzemeden ne kadar gerektiği bilgisi yazılıyor ve bu bilgi birbirine köprülenmiş iki OPC sunucu ile SQL server veri tabanına iletilerek buradan ilgili malzemenin tedarikçisinin veri tabanına, oradan da yine OPC server ile tedarikçi PLC’lerine iletiliyor. Bu bilgiyi alan tedarikçi PLC’leri gerekli üretimi yapıp otomotiv üreticisine aktarmaktadır. Böylece tedarik zinciri bir otomasyon sistemine bağlanmış olarak daha hızlı ve güvenilir biçimde işlemektedir [5].

4. OPC UA STANDARDI

OPC UA’nın tüm standartları bir araya toplamasının yanında, güvenli bağlantı çözümünü sunması da önemli bir özelliğidir. Bilgisayarlar arası veri alışverişine de imkân tanıyan OPC standartları, OPC UA öncesinde Microsoft’un Distributed Component Object Model (DCOM) altyapısını kullanarak uzak bilgisayarların uygulamaları arasında veri alışverişini sağlıyordu. Ancak bunun için, veri alışverişi yapacak bilgisayarlarda bazı Windows kullanıcı izinleri değiştirilir. Bu da bir takım güvenlik açıklarına sebep olabilir. Bu nedenle, OPC UA’da güvenlik mekanizması

olarak, veri alışverişi yapacak bilgisayarlarda sertifika modeli uygulanmış, böylece her tür bağlantıya izin verilmesi yerine, belirli kimliğe sahip olan uygulamaların ancak belirli portlardan erişimine izin verilmiştir.

5. SONUÇ

OPC, günümüzde kurumsallaşmış tesislerin endüstriyel otomasyon yapılarında hayati öneme sahiptir. Artık OPC sunucu yazılımları, saha verilerinin OPC sunucularına çıktıktan sonra bir veri tabanına kaydedilmesi, ya da bir SCADA yazılımıyla bağdaştırılması gibi işlevleri yapabilmektedir. Otomotiv örneğinde de görüldüğü gibi, artık sadece saha cihazlarından değil, Open Database Connectivity (ODBC) sürücüsü ile bir veri tabanından hatta Dynamic Data Exchange (DDE) sunucusu veya bir Excel çalışma sayfasından veri alışverişi yapılabilir. Bu sayede en alt seviyeden en üst seviyeye kadar bütünleşmiş bir otomasyon sistemi ve bilgi akışı mümkündür. OPC teknolojisinin bilinirliği ve uygulamaları ülkemizde son yıllarda giderek artmaktadır.

KAYNAKÇA

1. Ünlü, Y. 2007. OPC Standardı ve Uygulaması Yüksek Lisans Tezi, İstanbul, 22
2. Winchester, C. “What is OPC?”, http://www.opcactivex.com/What_is_opc/what_is_opc2.html
3. Mahnke, W., Leitner, S. H., Damm, M. 2009. OPC Unified Architecture, Berlin, 3
4. OPC Foundation, http://www.opcfoundation.org/Default.aspx/01_about/01_what_is_opc?MID=AboutOPC
5. http://www.kepware.com/Solution_Chronicles/OPC_Products_Secure_Data_Exchange.asp