

GPS Mesajlarının Nesneye Yönelik Yazılım Tasarım Prensiplerinden Genişlemeye Açık, Değişikliğe Kapalı Prensibine Göre Ele Alınması

Mehmet Mustafa GÜRSUL*¹, Muhammed Ali ASAN², Tahsin Alp YANAR³

¹Savunma Teknolojileri Mühendislik ve Tic. A.Ş. , Ankara

²Savunma Teknolojileri Mühendislik ve Tic. A.Ş. , Ankara

³Savunma Teknolojileri Mühendislik ve Tic. A.Ş. , Ankara

(Alınış / Received: 06.06.2016, Kabul / Accepted: 12.08.2016,
Online Yayınlanma / Published Online: 09.01.2017)

Anahtar Kelimeler
Strateji Tasarım
Kalıbı,
Fabrika Metot
Tasarım Kalıbı,
Açık Kapalı
Tasarım Prensibi,
NMEA 0183
Standardı,
GPS Mesajları

Özet: NMEA 0183; deniz sistemlerindeki elektronik cihazlar arasındaki iletişimde kullanılan bir standarttır. Bu uygulama bildirisinde; ilk önce NMEA 0183 standardı ile ilgili genel bilgiler verilecektir. Sonra ise; GPS alıcısından gönderilen NMEA 0183 standardındaki mesajların nesneye yönelik programlamanın temel prensiplerinden olan “genişlemeye açık fakat değişikliğe kapalı” prensibine uygun olarak strateji ve fabrika tasarım kalıpları ile yazılım tasarımının nasıl yapıldığı anlatılacaktır.

Handling GPS Messages According To The Object Oriented Software Design Principles of Open for Extension, Closed for Modification Principle

Keywords
Strategy Design
Pattern,
Factory Method
Design Pattern,
Open Closed
Design Principle,
NMEA 0183
Standard,
GPS Messages

Abstract: NMEA 0183 is a standard for interfacing marine electronic devices. In this application article, general information about the NMEA 0183 standard will be given first. Then, design of the software, which handles NMEA 0183 messages coming from GPS receiver, by using strategy and factory method design patterns will be discussed according to the one of the object oriented programming principle “open for extension, close for the modification”.

*Sorumlu yazar: mehmetgursul@gmail.com

1. Giriş

NMEA 0183; deniz sistemlerinde kullanılan derinlikölçer, sonar, pusula, otomatik pilot, GPS alıcıları gibi elektronik cihazlar arasındaki iletişimde kullanılan bir standarttır. Amerika'da kurulan Ulusal Deniz Elektronik Birliği

(National Marine Electronics Association) tarafından tanımlanmıştır ve kontrol edilmektedir. Deniz uygulamalarında NMEA 0183 standardının yerini yavaş yavaş NMEA 2000 standardı almaya başlamıştır [1].

Tipik bir NMEA 0183 standardındaki GPS cümlesi aşağıdaki gibi bir formata sahiptir:

- <başlangıç karakteri> <başlık>,<veri...>*<kontrol numarası><CR><LF>
- Tüm NMEA 0183 standardındaki mesajlar başlangıç karakteri (starting delimiter) olarak "\$" ya da "!" karakteri ile başlar ve her bir veri alanı "," karakteri ile ayrılır.
- Kontrol numarası (checksum) değeri: \$ ve * karakterleri arasındaki karakterlerin 8 bitlik exclusive OR değeridir ('\$' ve '*' karakterleri hesaplamaya dâhil edilmez)[2].

GPS alıcısından gönderilen ve GP ile başlayan temel mesajlardan bazıları aşağıda verilmektedir:

- GPRMC: Zaman, konum, yön ve hız bilgilerini içeren mesajdır.
- GPGLL: Zaman, konum ve durum bilgisi içeren mesajdır.
- GPGLL: Zaman, konum, kullanılan uydu sayısı, anten irtifası vb bilgileri içeren mesajdır [3].

Bu mesajlardan \$GPGLL mesajının formatı detaylı bir şekilde aşağıda verilmiştir.

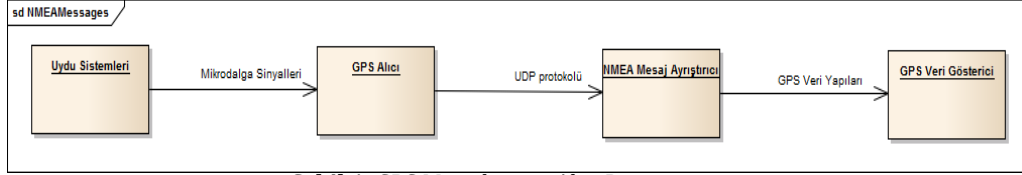
\$GPGLL,4101.38311,N,02834.97812,E,065932.18,A,A*6F

- GP, bu mesajın kaynağının GPS uydu sistemi olduğunu göstermektedir. Diğer

seçenekler GLONASS uydu sistemi için GL, GNSS sistemi kullanan aygıtlar için ise GN'dir.

- 4101.38311; DDMM.MMMMM formatındaki enlem değeri.
- N; kuzey enlemi olduğunu belirtir.
- 02834.97812; DDDMM.MMMMM formatındaki boylam değeri.
- E; doğu boylamı olduğunu belirtir.
- *6F, kontrol toplamı (checksum) değerini gösterir.
- İlk A, durum (status) değeridir. A ise geçerli (valid), diğer durumlarda geçersiz (invalid) dir.
- İkinci A, mode indicator değeridir. Aşağıdaki değerleri almaktadır:

- A: Auto
- D: Differential
- E: Estimated
- M: Manual Input
- S: Simulator
- N: Data Invalid



Şekil 1: GPS Mesajlarının Akış Diyagramı

Bu uygulama bildirisinde; yukarıda adları verilen NMEA 0183 standardındaki GPS mesajlarının nesneye yönelik programlamanın temel prensiplerinden olan “open for extension, close for modification” prensibine uygun olarak nasıl bir mimari ile ele alınacağına değinilecektir.

2. Problem

Uydu sistemlerinden GPS alıcısına gelen mikro dalgalar, GPS alıcısı tarafından NMEA 0183 standardına uygun olarak çeşitli mesajlar halinde UDP/IP protokolü ile Ethernet kartı üzerinden gönderilmektedir (bknz. Şekil 1). Sürekli gönderilen bu mesajların Ethernet kartı üzerinden okunup, genel geçerlilik kontrolleri yapıldıktan sonra ilgili mesajın nesnesindeki verileri doldurulmak istenmektedir.

GPS alıcısı tarafından GP ile başlayan birçok türde mesaj gönderilmektedir. GPS mesajlarını ele alan bir yazılım projesinde ihtiyaç duyulan mesajlar ele alındıktan sonra yeni ihtiyaçlara uygun olarak başka GP ile başlayan mesajların da işlenmesi ihtiyacı doğabilmektedir. Bu yeni mesajları ele alırken de var olan mesajları çözümleyen kod parçalarında en az değişiklik ile yazılımın genişleyebilir olması gereklidir.

Bu ihtiyaçlara uygun şekilde Çözüm ve Tasarım bölümünde anlatılan şekilde uygulama geliştirilebilir.

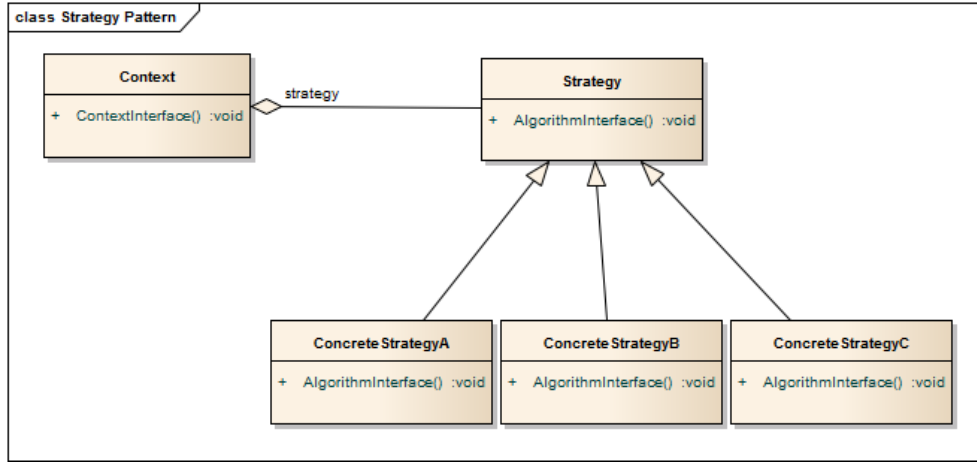
3. Çözüm ve Tasarım

Nesneye yönelik tasarım prensipleri; kodu tasarlarken ya da yazarken uygulanan ve böylece kodun daha kolay bakımı yapılabilir, esnek ve genişleyebilir olmasını sağlayan basit temel araç ya da tekniklerdir [4]. Nesneye yönelik tasarımın beş temel prensibinin baş harflerinden oluşan SOLID [5] kelimesinin içerisindeki “O” harfi, “open/closed” prensibini belirtmektedir. “Open/closed” prensibi yazılım varlıklarının (sınıflar, modüller, fonksiyonlar, vb.) genişlemeye açık fakat değişikliğe kapalı olması olarak tanımlanmaktadır [4].

GPS’ten gelen NMEA 0183 standardındaki mesajların ele alınması ile ilgili tasarımda open/closed prensibine uygun olarak strateji ve fabrika metot tasarım kalıpları birlikte kullanılmıştır. Strateji ve fabrika metot tasarım kalıpları ile ilgili kısa bilgi aşağıda verilmektedir.

Strateji tasarım kalıbı, algoritmanın davranışının çalışma zamanında (runtime) belirlenmesini sağlayan bir yazılım tasarım kalıbıdır. Strateji tasarım kalıbı;

- Bir algoritma ailesi tanımlar,
- Her bir algoritmayı sarmalar (encapsulates),
- Algoritmanın bu algoritma ailesi içinde değiştirilebilir olmasını sağlar [6].



Şekil 2: Strateji Tasarım Kalıbı Sınıf Diyagramı

Strateji tasarım kalıbı sınıf diyagramı aşağıda Şekil 2'de gösterilmektedir.

Fabrika metot tasarım kalıbı; nesnelerin yaratılmasını başka bir sınıfa delege eden bir yazılım tasarım kalıbıdır. Fabrika metot tasarım kalıbı ile;

- Nesne yaratılması fabrika sınıfı içinde yer alır,
- Böylece nesnenin yaratılması ve nesnenin kullanılması ayrıştırılmış olur,
- Fabrika sınıfı içindeki nesne yaratılması mantığında yapılacak değişiklik, nesne kullanımı tarafında herhangi bir değişikliğe gerek duymadan yapılabilir [6].

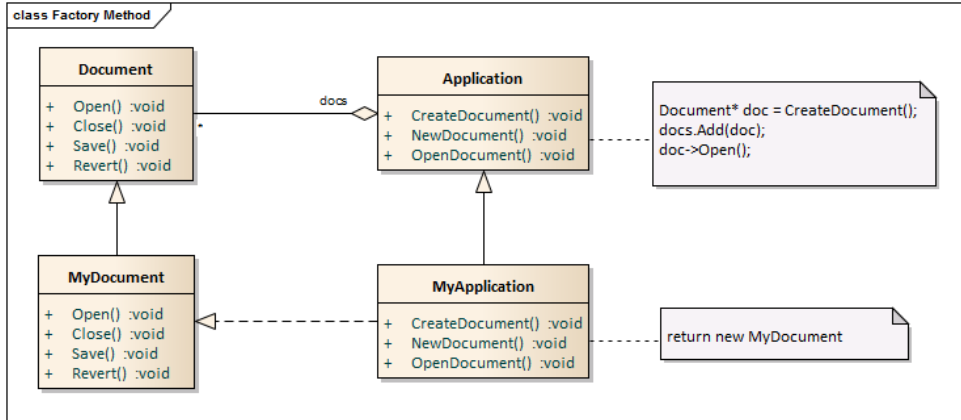
Fabrika metot tasarım kalıbı sınıf diyagramı aşağıda Şekil 3'te gösterilmektedir.

Strateji ve fabrika metot tasarım kalıpları Şekil 4'te gösterildiği şekilde sisteme uyarlanmıştır. Bu tasarım ile yeni bir mesaj yazılımda ele almak

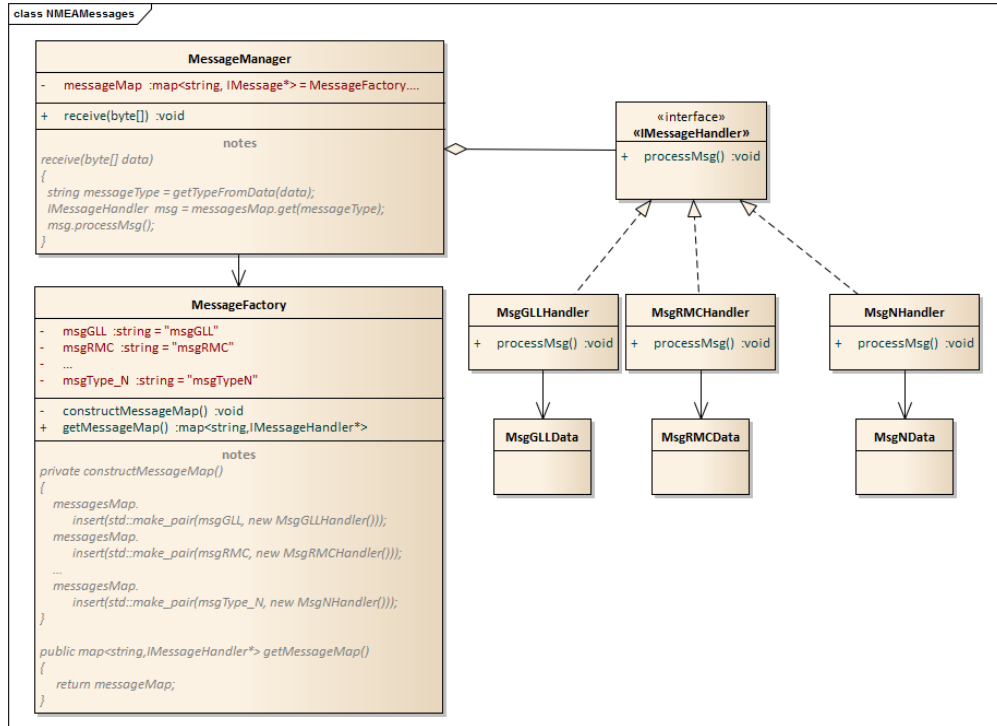
ihtiyacı doğduğunda; strateji tasarım kalıbı sayesinde, var olan mesajlarda herhangi bir değişiklik yapma gereği duymadan, yeni mesajı yazılıma kolayca entegre edebilme olanağı sağlanmış olmaktadır.

Örneğin X adında yeni bir NMEA 0183 standardındaki mesaj yazılıma eklemek istendiğinde, strateji tasarım kalıbı sayesinde; Şekil 4'te gösterilen mesajlarda ve MessageManager sınıfında herhangi bir değişiklik yapmaya gerek kalmadan sadece aşağıda belirtilen iki eklemenin yapılması yeterli olacaktır:

- MsgXHandler şeklinde IMessageHandler arayüzünü gerçekleyen bir sınıf eklenip,
- MessageFactory sınıfında bu yeni mesaj için <messageType, MsgXHandler> anahtar-değer (key-value) verisini eklemek yeterli olacaktır. Bu kısımda istenirse Dependency Injection ile kod yeniden derlemeye gerek kalmadan yapılabilir.



Şekil 3: Fabrika Metot Tasarım Kalıbı Sınıf Diyagramı



Şekil 4: GPS Mesaj Ayrıştırma Sınıf Diyagramı

Strateji tasarım kalıbının avantajlarının yanında nesne sayısını artması olarak dezavantajı da bulunmaktadır. Koşul deyiminin eklenmesini gerektirecektir.

Diğer bir alternatif de koşullu deyimler kullanarak hangi alt sınıfın mesajı ele alacağına karar vermektir. Fakat bu şekilde yapıldığında yeni bir mesaj ele alınmak istendiğinde yeni bir

4. Tartışma ve Sonuç

Yazılım projelerindeki temel sorunlardan birisi; yeni ihtiyaçlara uygun olarak var olan yazılımın genişleyememesi ve herhangi bir değişiklikte var olan kod parçalarında da değişiklik yapılma gerekliliğinin doğmasıdır.

Bu makale ile GPS mesajlarının ele alınacağı bir yazılım projesinin genişlemeye açık fakat var olan kodlar üzerinde de en az değişikliğe yol açacak şekildeki yazılım tasarımı ile yazılımın yeni ihtiyaçlara cevap verebilmesi sağlanmıştır. Ayrıca kod okunabilirliği ve yazılım bakımı da daha kolay duruma getirilmiştir.

Kaynakça

- [1]. Standard NMEA-0183 sentences description, freenmea.net/docs, Erişim Tarihi: 16.08.2016
- [2]. IEC 61162-1 Maritime navigation and radiocommunication equipment and systems – Digital interfaces – Part 1: Single talker and multiple listeners Edition 4.0, November 2010.
- [3]. NMEA Data, www.gpsinformation.org/dale/nmea.htm, Erişim Tarihi: 16.08.2016
- [4]. Head First Object-Oriented Analysis&Design, Brett D. McLaughlin, Garry Pollice, David West, 2006, O'Reilly Media, First Edition
- [5]. SOLID (object-oriented design), [https://en.wikipedia.org/wiki/SOLID_\(object-oriented_design\)](https://en.wikipedia.org/wiki/SOLID_(object-oriented_design)), Erişim Tarihi: 20.05.2016
- [6]. Design Patterns, Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides