Yazılım Emniyeti Perspektifinin Görünümler ve Ötesi Mimari Çerçevesine Uygulanması

Havva Gülay Gürbüz[1], Nagehan Pala Er[2], Bedir Tekinerdogan[2]

Bilkent Üniversitesi, Bilgisayar Mühendisliği Bölümü, 06800, Ankara, Türkiye
[1] havva.gurbuz@bilkent.edu.tr
[2]{nagehan,bedir}@cs.bilkent.edu.tr

Özet. Farklı paydaşların farklı ilgilerini gösteren mimari görünümleri modellemek için birçok *mimari bakış açısı* yaklaşımı ortaya konulmuştur. Mimari bakış açılarında sistem ilgileri ile beraber kalite ilgileri de ele alınmalıdır. Bu bağlamda mimari görünümlerdeki kalite ilgilerini gösterebilmek için *mimari perspektifler* tanımlanmıştır. Bir mimari perspektif, istenen kalite ilgisinin gerçekleştirilebilmesi için uygulanması gereken aktiviteleri, taktikleri ve ilkeleri içerir. Her bir perspektif, birden fazla mimari görünüm tarafından ele alınabilir. Bazı kalite ilgileri için literatürde farklı mimari perspektifler sunulmuştur. Yazılım emniyeti perspektifi, Rozanksi ve Woods'un perspektif tanımına göre emniyet ilgilerini mimari görünümlerde ele alabilmek için daha önceki bir çalışmamızda tanımlanmıştır. Yazılım emniyeti, emniyet-kritik sistemler için önemli bir ilgi olup, emniyet-kritik durumlardaki risklerin azaltılması ya da ortadan kaldırılması için mimari analiz sürecinin başlangıç aşamasından itibaren değerlendirilmesi gerekir. Bu çalışmada yazılım emniyeti perspektifinin Görünümler ve Ötesi mimari çerçevesine uygulanması sunulmaktadır. Yazılım emniyeti perspektifinin Görünümler ve Ötesi mimari çerçevesine uygulanması, emniyet ilgileriyle alakalı tasarım ve analiz kararlarında sistem ve yazılım mimarlarına yardımcı olmaktadır. Yazılım emniyeti perspektifinin uygulaması gerçek bir endüstriyel uygulama kullanılarak açıklanmaktadır.

Anahtar Kelimeler: Yazılım mimari tasarımı, yazılım mimari modelleme, yazılım mimari analizi, emniyet-kritik sistemler

1 Giriş

Tasarım aşamasında karar alınırken farklı paydaşlar için *mimari görünümlerin (architectural views)* modellenmesi, yazılım mimari tasarımında kullanılan yaygın pratiklerden birisidir [1]. Bir mimari görünüm belirli bir ilgiyi desteklemek için tanımlanan sistem elemanları ve bu elemanlar arasındaki ilişkileri gösterir. Bir sistemin birden fazla görünüm ile ifade edilmesi ilgilerin ayrıştırılmasını sağlar. Böylelikle her bir paydaş için yazılım mimarisinin modellenmesine, daha iyi anlaşılmasına ve analiz edilmesine yardımcı olur. *Mimari bakış açıları (architectural viewpoints)*, mimari görünümlerin oluşturulması, kullanılması ve analiz edilmesi için tanımlanan kuralları ve ilkeleri içerir. Mimari bakış açıları bir *mimari çerçeve (architectural framework)*

tanımlanarak organize edilir ve yapılandırılır. Mimari çerçevelere Kruchten'in 4+1 Görünüm Modeli [7], Rozanski ve Woods'un mimari görünümler yaklaşımı [11] ve Görünümler ve Ötesi (Views&Beyond) yaklaşımı [1] örnek olarak verilebilir.

Kalite ilgilerini adresleyebilmek için, genel olarak, mimari çerçeve yaklaşımlarında ayrı birer mimari görünüm sunulmamıştır. Örneğin, ISO/IEC 42010 standardı [6] farklı ilgileri adreslemek için özel bakış açıları tanımlamamıştır. Görünümler ve Ötesi yaklaşımında ise tanımlanan her bir görünümde kalite ilgileri dolaylı olarak ifade edilse de, kalite ilgilerini tanımlamak için ayrı görünümler sunulmamıştır. Bu kapsamda, mimari görünümlerdeki kalite ilgilerini adreslemek için, Rozanski ve Woods tarafından *mimari perspektif (architectural perspective)* yaklaşımı [11] ortaya konulmuştur. Bir mimari perspektif istenen kalite ilgisini gerçekleştirebilmek için uygulanması gereken aktiviteleri, taktikleri ve ilkeleri içerir ve birden fazla mimari görünüm tarafından dikkate alınır. Rozanski ve Woods güvenlik, performans, ölçeklenebilirlik, erişilebilirlik gibi bazı kalite ilgileri için mimari perspektifler tanımlamıştır.

Emniyet-kritik sistemlerdeki bir aksama ya da işlev bozukluğu ölümlere, insanlar üzerinde ciddi yaralanmalara ya da çevresel hasarlara neden olabileceği için, bu sistemler tasarlanırken yazılım emniyeti önemli bir ilgi haline gelmektedir. Riskleri azaltmak ya da ortadan kaldırmak için emniyet ilgisinin gerçekleştirim yapılmadan önce tasarım aşamasının başlangıcından itibaren değerlendirilmesi kritik bir önem arz etmektedir. Emniyet ilgisini ele alabilmek için birçok standart ve gerçekleştirim yaklaşımları tanımlanmış, fakat hiçbiri emniyet ilgisini mimari modelleme düzeyinde göz önünde bulundurmamıştır. Bu sebeple, daha önceki çalışmamızda emniyet ilgisini mimari düzeyde adresleyebilmek için Rozanski ve Woods yaklaşımındaki kılavuza göre yazılım emniyet perspektifini tanımlanmıştır [2][3]. Emniyet perspektifi emniyet ilgileriyle alakalı tasarım ve analiz kararlarının alınmasında sistem ve yazılım mimarlarına yardımcı olmaktadır. Bu çalışmada yazılım emniyet perspektifinin Görünümler ve Ötesi mimari çerçevesine uygulaması sunulmaktadır. Yazılım emniyeti perspektifinin uygulaması gerçek bir endüstriyel örnek uygulama kullanılarak açıklanmaktadır.

Bu çalışmanın kalanı şu şekilde organize edilmiştir: Bölüm 2'de yazılım mimarisi modelleme ile ilgili teorik altyapı anlatılmaktadır. Yazılım emniyet perspektifi Bölüm 3'te açıklanmaktadır. Bölüm 4'te yazılım emniyet perspektifinin Görünümler ve Ötesi yaklaşımına uygulaması sunulmaktadır. Bölüm 5'de gerçek bir endüstriyel uygulama kullanılarak yazılım emniyet perspektifinin uygulanması anlatılmaktadır. Bölüm 6, ilgili çalışmaları sunmaktadır. Son olarak Bölüm 7 çalışma sonuçlarını paylaşmaktadır.

2 Yazılım Mimarisi Modelleme

2.1 Görünümler ve Ötesi Mimari Çerçevesi

Görünümler ve Ötesi (Views&Beyond) mimari çerçevesi [1] yazılım mimari tasarımını desteklemek için her biri farklı stiller içeren üç farklı mimari görünümden oluşmaktadır. Bu yaklaşımda bakış açılarını tanımlamak için *görünüm tipi (view type)* ve *stil (style)* kavramları kullanılmaktadır. Bu yaklaşıma göre sistemin temel gerçekleştirim birimlerinin tanımlanması için *modül görünümü (module view)*, sistemin

çalışma birimlerinin tanımlanması için *bileşen&bağlantılar görünümü (component & connector view)*, ve sistemdeki yazılımlar ile yazılım geliştirme ve çalıştırma ortamları arasındaki ilişkinin tanımlanması için *tahsis görünümü (allocation view)* oluşturulmuştur.

Modül görünümünde modüller birbiriyle tutarlı sorumlulukları gerçekleyen birincil sistem elemanlarıdır. Bu görünüm kapsamında *ayrıştırma (decomposition), kullanım (uses), genelleştirme (generalization), katmanlı (layered), ilgiler (aspects), veri modeli (data model) stilleri* tanımlanmıştır. Ayrıştırma stili sistemdeki gerçekleştirim birimlerinin modül ve alt-modül olarak nasıl ayrıldığını ve sistemdeki sorumlulukların modüller ve alt-modüller arasında nasıl paylaştırıldığını gösterir. Kullanım stili modüller ve alt-modüller arasındaki bağımlılıkları gösterir. Genelleştirme stili modüller arasındaki kalıtım ilişkisini gösterir. Katmanlı stil, birbiriyle ilgili modülleri katman adı verilen bir yapıda toplayarak, katmanlar arasındaki kullanılabilirlik ilişkisini tanımlar. Bir katmandan diğer katmana kullanabilir ilişkisi tanımlanmışsa, birinci katmandaki herhangi bir modül ikinci katmandaki herhangi bir modülü kullanabilir demektir. İlgiler stili enine kesen ilgileri gerçekleyen modülleri ve bu modüllerin sistemdeki diğer modüllerle birlikte nasıl ilişkilendirildiğini gösterir. Veri Modeli stili ise veri varlıklarının yapılarını ve birbirleri arasındaki ilişkileri tanımlar.

Bileşen&bağlantılar görünümünde bileşenler sistemdeki çalışma birimleri olarak ele alınmıştır. Bu görünüm sistemin çalışma zamanı açısından ele alınmasını sağlar. Bu görünüm kapsamında *çağrı-dönüş (call-return), veri akışı (data flow), olay-tabanlı (event-based)* ve *depo (repository)* ana stilleri tanımlanmış ve sıklıkla kullanılan stiller bu stiller altında kategorize edilerek tanımlanmıştır. Çağrı-Dönüş stili bileşenler tarafından sağlanan servisleri ve bu servisleri eşzamanlı bir şekilde başlatan diğer bileşenleri gösterir. Çağrı-Dönüş stili altında *istemci-sunucu (client-server), noktadan noktaya (peer-to-peer)* ve *servis odaklı mimari (service-oriented architecture) stilleri* tanımlanmıştır. Veri Akışı stili sistemdeki veri akışını modeller. Veri Akışı stili kapsamında *borular ve filtreler (pipe-and-filters) stili* tanımlanmıştır. Olay-Tabanlı stili eşzamanlı olmayan olay ve mesajlar üzerinden etkileşimde bulunan bileşenleri gösterir. Bu stil kapsamında *yayımla-abone ol (publish-subscribe) stili* tanımlanmıştır. Depo stili büyük miktarda kalıcı ve paylaşılan veriler üzerinden etkileşimde bulunan bileşenleri gösterir. Depo stili kapsamında *paylaşılan veri (shared data) stili* tanımlanmıştır. Bütün bu stillerin yanında *çok katmanlı (multi-tier) stili* de tanımlanmıştır. Bu stilde bileşenler *katman (tier)* adı verilen yapılarda gruplandırılarak gösterilir.

Tahsis Görünümü'nde sistemdeki yazılım elemanları ve bu yazılım elemanlarının geliştirme ve çalıştırma ortamları arasındaki ilişkisi tanımlanmıştır. Bu görünüm kapsamında *konuşlandırma (deployment), kurulum (install), iş ataması (work assignment) stilleri* tanımlanmıştır. Konuşlandırma stili yazılım bileşenleri ve bağlantıları ile yazılımın üzerinde çalıştığı donanımlar arasındaki eşleştirmeyi belirtir. Kurulum stili mimarideki bileşenler ile çalışma ortamındaki dosya sistem yapıları arasındaki eşleştirmeyi tanımlar. İş Ataması stili yazılım bileşenleri ile bu bileşenlerin geliştirilmesinden sorumlu çalışanlar, takımlar ve çalışma birimleri arasındaki eşleştirmeyi tanımlar.

2.2 Mimari Perspektifler

Mimari tasarımı desteklemek için Rozanski ve Woods yedi farklı bakış açısı içeren bir mimari çerçeve ortaya koymuştur [11]. Rozanski ve Woods kalite ilgilerinin tanımlanan mimari bakış açılarını çapraz kestiklerini (crosscutting) ve her bir kalite ilgisi için farklı birer bakış açısı tanımlamanın anlamlı olmadığını belirtmiştir. Bu sebeple Rozanski ve Woods istenilen kalite ilgilerinin gerçekleştirilebilmek için uygulanması gereken aktiviteleri, taktikleri ve ilkeleri içeren mimari perspektif kavramını sunmuştur. Sistem genelinde kalite ilgilerini ele alabilmek için tanımlanan her ilgili perspektif birden fazla bakış açısına uygulanabilir. Böylelikle mimari görünümler mimariyi tanımlarken, mimari perspektifler sistemin istenen kalite özelliklerini sağladığını garantilemek için mimariyi analiz etmeye ve üzerinde değişiklikler yapmaya olanak sağlar. Rozanski ve Woods *güvenlik (security), performans & ölçeklenebilirlik (performance & scalability), kullanılırlık & dayanıklılık (availability & resilence), evrim (evolution), erişilebilirlik (accessibility), geliştirim kaynağı (development resource), uluslararasılaştırma (internationalization), yerleşim (location), düzenleme (regulation)* ve *kullanılabilirlik (usability)* perspektiflerini tanımlamıştır. Her bir perspektif ilgili olduğu kalite ilgisini gerçekleştirmektedir.

3 Yazılım Emniyet Perspektifi

Daha öncede belirtildiği gibi emniyet-kritik sistemler için emniyet önemli bir ilgidir. Emniyet ilgisini gerçekleştirebilmek için uygulanması gereken taktikleri ve ilkeleri mimari tasarım düzeyinde ele alabilmek için daha önceki çalışmamızda [2][3] Rozanski ve Woods'un perspektif tanımları referans alınarak, literatürde var olmayan yazılım emniyeti perspektifi tanımlanmıştır.

İstenen Kalite	Emniyet ile ilgili kararlar hakkında bilgi sağlamak ve ciddi hasarlara ve tehlikelere yol açabilecek sistem işlevlerini denetlemek ve kontrol etmek
Uygulanabilirlik	Tehlikeli ve emniyet-kritik işlevler içeren tüm sistemler
İlgiler	Hata (Failure), Tehlike (Hazard), Risk, Hata Toleransı (Fault Tolerance), Kullanılırlık (Availability), Güvenilirlik (Reliability), Doğruluk (Accuracy), Performans
Aktiviteler	Tehlikelerin belirlenmesi, Risklerin tanımlanması, Emniyet gereksinimlerinin belirlenmesi, Emniyet modelinin oluşturulması, Emniyet gereksinimlerine göre değerlendirme yapılması
Mimari Taktikler	Hataları ve tehlikeleri önleme, Hataları tespit etme yöntemlerini tanımlama, Oluşan hataların sonuçlarını azaltma
Problemler ve Tuzak Noktalar	Hata Toleransını tanımlama, Emniyet modelinin ya da gereksinimlerinin açık bir şekilde tanımlanmamış olması, Azımsanmış emniyet problemleri

Tablo 1. Yazılım Emniyet Perspektifi Tanımı

Tablo 1 emniyet perspektifinin tanımını sunmaktadır. Yazılım emniyet perspektifi ile emniyet ile ilgili kararlar hakkında bilgi sağlamak ve ciddi hasarlara ve tehlikelere yol açabilecek sistem işlevlerini denetlemek ve kontrol etmek amaçlanmaktadır. Bu perspektif tehlikeli ve emniyet-kritik işlevler içeren tüm sistemlerin tasarım aşamasında bir kılavuz olarak kullanılabilir. Emniyet perspektifi tarafından adreslenen ilgi-

ler literatür taraması ile belirlenmiş ve Tablo 1'de gösterilmiştir. Mimari tasarım aşamasında emniyet perspektifinin uygulanması için gereken aktiviteler ise aşağıdaki gibi sıralanmıştır:

1. *Tehlikelerin belirlenmesi:* Emniyet gereksinimlerinin belirlenebilmesi için tehlike analizi gerçekleştirilerek sistemdeki olası tehlikeler, tehlikelere sebep olabilecek durumlar, tehlikelerin sonuçları ve sistemdeki her bir tehlike için tehlikenin şiddeti [9] (ölümcül, kritik, marjinal, ihmal edilebilir) belirlenmelidir.
2. *Risklerin tanımlanması:* Tehlikeler belirlendikten sonra, bu tehlikelerle ilgili riskler tanımlanmalıdır. Risk tanımı belirlenen tehlikenin şiddeti ve oluşma olasılığı (sık sık, olası, nadiren, çok az, olası olmayan) bilgileri ile ifade edilir. Daha sonra hata ağacı analizi [8] (fault tree analysis), olay ağacı analizi, simülasyon gibi metotlarla risk değerlendirmesi yapılmalıdır.
3. *Emniyet gereksinimlerinin belirlenmesi*: Emniyet modelinin oluşturulabilmesi için, tanımlanan tehlikelere dayanılarak emniyet gereksinimlerinin belirlenmesi gerekir. Emniyet gereksinimleri belirlenirken, sistem gereksinimleri göz önünde bulundurularak ön tehlike analizi, fonksiyonel tehlike analizi, hata ağacı analizi ve ön sistem emniyet analizi [14] metotları kullanılabilir.
4. *Emniyet modelinin oluşturulması:* Sistem tasarımında kullanılan modellerin yanında, sistemdeki emniyet ilgisinin doğru bir şekilde anlaşılıp geliştirilebilmesi için emniyet-kritik elemanların ve bileşenlerin de özel olarak gösterildiği emniyet modelleri de oluşturulmalıdır. Emniyet modeli UML diyagramlarına stereotipiler eklenerek [10], alana özgü diller ile tanımlanarak [15] ve özdevinirler [17] kullanılarak emniyet gereksinimlerinden türetilebilir.
5. *Emniyet gereksinimlerine göre değerlendirme yapılması:* Emniyet modeli oluşturulduktan sonra, modelin belirlenen emniyet gereksinimleri ile uyumlu olup olmadığı kontrol edilmelidir. Bunun için bazı örnek emniyet durumları oluşturularak bunlar üzerinden emniyet modeli kontrol edilebilir. Ayrıca üçüncü otorite tarafından değerlendirme süreci gerçekleştirilebilir. Eğer oluşturulan model ile emniyet gereksinimleri arasında çelişki varsa, hatalar giderilene kadar emniyet modeli gözden geçirilip düzeltmeler yapılmalıdır.

Tanımlanan mimari perspektif tarafından adreslenen kalite özellikleri istenen düzeyde sağlanmadığında tanımlanan mimari taktikler ile yazılım tasarımına katkıda bulunulmaya çalışılmıştır. Bu çerçevede, yazılım emniyeti tasarımını destekleyebilmek için hataları ve tehlikeleri önleme, hataları tespit etme yöntemlerini tanımlama, oluşan hataların sonuçlarını azaltma taktikleri tanımlanmıştır. Yazılım tasarımı yapılırken, emniyet ilgisinin gösterilmesi ile ilgili olası sorunlar ve zorluklar hata toleransının tanımlanması, emniyet modelinin ya da gereksinimlerinin açık bir şekilde tanımlanmamış olması, azımsanmış emniyet problemleri olarak belirlenmiştir.

4 Yazılım Emniyet Perspektifinin Görünümler ve Ötesi Mimari Çerçevesine Uygulanması

Tablo 2 yazılım emniyeti perspektifinin Görünümler ve Ötesi mimari çerçevesine uygulamasını göstermektedir.

Mimari Stil	Uygulanabilirlik
Ayrıştırma	Sistemdeki hangi modüllerin ve alt-modüllerin emniyet-kritik olarak değerlendirilmesi gerektiğini belirtir.
Kullanım	Emniyet-kritik modüller ile diğer modüller arasındaki bağımlılıkları gösterir.
Genelleştirme	Emniyet-kritik modüller ve diğer modüller arasındaki kalıtım ilişkisini açıklar.
Katmanlı	Modülleri katmanlar olarak gruplar. Böylelikle hangi katmanın hangi emniyet-kritik modülleri içerdiği gösterilir. Ayrıca tanımlı olan emniyet-kritik modüllerin hangi katmanlar tarafından kullanılabileceğini tanımlar.
İlgiler	Emniyet-kritik modüllerle ilgili olan ilgi modüllerini belirtir.
Veri Modeli	Yazılım emniyet perspektifinin bu stil üzerinde etkisi yoktur.
Çağrı-Dönüş	Sistemdeki emniyet-kritik bileşenleri ve bu bileşenler ile sistemdeki diğer bileşenler arasındaki etkileşimi gösterir.
Veri Akışı	Sistemdeki emniyet-kritik işlevlerdeki veri akışını gösterir.
Olay-tabanlı	Eşzamanlı olmayan olay ve mesajlar üzerinden etkileşimde bulunan emniyet-kritik bileşenleri ve diğer sistem bileşenlerini gösterir.
Depo	Yazılım emniyet perspektifinin bu stil üzerinde etkisi yoktur.
Konuşlandırma	Emniyet-kritik bileşenler için gerekli olan donanım elemanlarını ve üçüncü parti yazılımları ve ortam elemanlarını belirtir.
Kurulum	Emniyet-kritik bileşenlerin geliştirim ortamı için gerekli olan yazılım elemanlarını, özel izinleri ve yapılandırma elemanlarını belirtir.
İş Ataması	Emniyet-kritik bileşenlerin geliştirilmesinden sorumlu kişileri, takımları ve çalışma gruplarını gösterir.

Tablo 2. Emniyet Perspektifinin Görünümler ve Ötesi Mimari Çerçevesine Uygulanması

5 Örnek Uygulama

Yazılım emniyetinin önemli olduğu alanlardan bir tanesi de aviyonik alanıdır. Aviyonik sistemlerde oluşan hataların ölümcül sonuçlara neden olan kazalara sebep olduğu görülmüştür. Aviyonik alanındaki yazılımların geliştirilmesi ve sertifikalanması için yapılması gereken aktiviteleri anlatan DO-178C [12] gibi çeşitli standartlar bulunmaktadır. Özellikle ticari aviyonik sistemlerin bu standartlara uygun olarak geliştirilmesi ve sertifikalanması gerekmektedir. Bu bölümde, yazılım emniyeti perspektifinin uygulaması gerçek bir endüstriyel aviyonik uygulama kullanılarak açıklanmaktadır. Endüstriyel aviyonik uygulama gerçekte binlerce gereksinim içermektedir. Bu bölümde anlatılan uygulama için Tablo 3'de verilen örnek bir gereksinim kümesi seçilmiştir.

1. Tehlikelerin belirlenmesi:

Emniyet mühendisliği çalışmaları kapsamında ilk olarak tehlike analizi yapılarak tehlikeler belirlenir. Bu analiz emniyet mühendisleri, sistem mühendisleri ve ilgili alandaki uzmanlarla (aviyonik mühendisleri ve pilotlar) beraber gerçekleştirilir. Örnek uygulamamız için bu analiz sonucunda belirlenen tehlikeler, olası nedenleri, sonuçları ve tehlikenin şiddeti gibi bilgilerle beraber Tablo 4'de verilmiştir. TH1, TH2, TH3 ve TH4 olarak numaralandırılan tehlikelerin şiddeti ölümcül olarak belirlenmiştir. Çünkü bu tehlikelerin olası sonucu hava aracı kazasıdır. Örneğin doğru yakıt mik-

tarı yerine yüksek bir yakıt miktarının gösterilmesi, pilotları yanlış yönlendirecek ve gidecekleri yere yakıt miktarı yetersiz olsa bile ulaşabilecekleri gibi yanlış bir izlenim edinmelerine sebep olabilecektir. TH5 olarak numaralandırılan tehlikenin şiddeti ise ihmal edilebilir olarak belirlenmiştir çünkü bu hata sadece yer istasyonu ile iletişim hatasına neden olabilir.

Gereksinim	**Açıklama**
Kalan yakıt miktarının gösterilmesi	Hava aracında kala yakıt miktarı, bütün yakıt tanklarında kalan yakıt miktarlarının toplamıdır.
Yüksekliğin gösterilmesi	Hava aracının yüksekliği deniz seviyesinden olan yükseklik olarak tanımlanır. Pilotlar özellikle iniş sırasında gösterilen yükseklik bilgisine güvenerek hareket ederler.
Konumun gösterilmesi	Hava aracının konumu GPS'den (Global Positioning System) alınan enlem ve boylam koordinatlarıdır. Hava aracının konum bilgisi rota yönetimi tarafından kullanılır ve rotadaki diğer noktaların konumları ile beraber pilotlara gösterilir. Rotadan sapma olup-olmadığını veya ne kadar sapıldığını pilotlar bu şekilde görebilirler.
Durumun (attitude) gösterilmesi	Hava aracının durumu, hava aracının X, Y ve Z eksenleri ile olan açılarıdır. Bu açılar yunuslama (pitch), sapma (yaw) ve yuvarlanma (roll) açıları olarak isimlendirilir. Genellikle, yunuslama ve yuvarlanma açıları Durum Yön Göstergesi (Attitude Director Indicator) ile gösterilir.
Radyo frekansının gösterilmesi	Radyo frekansı, yer istasyonu ile iletişim kurulan frekanstır.

Tablo 3. Örnek Uygulama için Seçilen Gereksinimler

Tehlike	**Olası Nedenleri**	**Sonuçları**	**Şiddeti**	**Olasılığı**	**Riski**
[TH1] Yakıt miktarının yanlış gösterilmesi	Yakıt sensörünün kaybı/hatası Yakıt sensörü ile iletişim kaybı/hatası Gösterge cihazının hatası	Hava aracı kazası	Ölümcül	Olası olmayan	Orta
[TH2] Yükseklik bilgisinin yanlış gösterilmesi	Yükseklik cihazının kaybı/hatası Yükseklik cihazı ile iletişim kaybı/hatası Gösterge cihazının hatası	Hava aracı kazası	Ölümcül	Olası olmayan	Orta
[TH3] Konum bilgisinin yanlış gösterilmesi	GPS cihazının kaybı/hatası GPS cihazı ile iletişim kaybı/hatası Gösterge cihazının hatası	Hava aracı kazası	Ölümcül	Olası olmayan	Orta
[TH4] Durum bilgisinin yanlış gösterilmesi	Cayroskop cihazının kaybı/hatası Cayroskop cihazı ile iletişim kaybı/hatası Gösterge cihazının hatası	Hava aracı kazası	Ölümcül	Olası olmayan	Orta
[TH5] Radyo frekansının yanlış gösterilmesi	Radyo cihazının kaybı/hatası Radyo cihazı ile iletişim kaybı/hatası Gösterge cihazının hatası	Yer istasyonu ile iletişim hatası	İhmal edilebilir	Nadiren	Düşük

Tablo 4. Örnek Uygulama için Tehlikelerin Belirlenmesi ve Risklerin Tanımlanması

2. Risklerin tanımlanması:

Her bir tehlike için tehlikenin olasılığı ve risk değerlendirmesi de Tablo 4'de verilmiştir. Genel olarak tasarım kriteri, ölümcül kategoride olan bütün tehlikelerin olasılığını "olası olmayan" şekilde tasarlamaktır. TH1, TH2, TH3 ve TH4 olarak numaralandırılan tehlikelerin risk kategorisi orta olarak belirlenmiştir, TH5 olarak numaralandırılan tehlikenin riski ise düşüktür.

3. Emniyet gereksinimlerinin belirlenmesi:

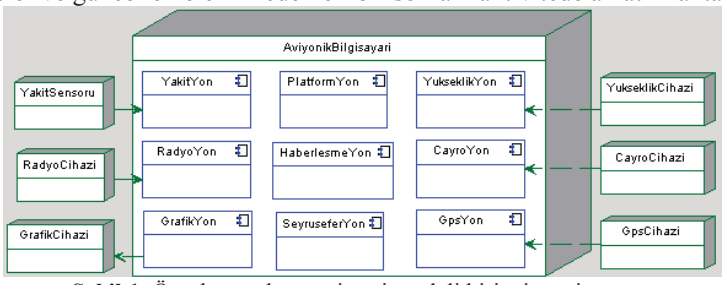
Tehlike analizi tamamlandıktan sonra emniyet gereksinimleri belirlenir. Örnek olarak, sonraki aktivitelerde de kullanılmak üzere, TH1 ile ilgili emniyet gereksinimleri Tablo 5'de listelenmiştir. Benzer emniyet gereksinimleri diğer tehlikeler için de belirlenebilir. EG1 (Emniyet Gereksinimi) yakıt miktarı bilgisinin en az iki bağımsız yakıt sensöründen alınmasını, EG2 alınan yakıt miktarı bilgilerinin karşılaştırılmasını ve karşılaştırma sonucuna göre gösterilmesini veya uyarı verilmesini, EG3 ise yakıt miktarı bilgisinin en az iki bağımsız göstergede gösterilmesini istemektedir.

No	Tanım
EG1	Yakıt miktarı bilgisi en az iki tane bağımsız sensörden alınmalıdır.
EG2	Eğer iki tane bağımsız sensörden alınan yakıt miktarları arasında belirlenen bir değerden daha fazla fark varsa, yakıt miktarı bilgisi gösterilmemelidir ve bununla ilgili bir uyarı verilmelidir.
EG3	Yakıt miktarı bilgisi en az iki tane bağımsız göstergede gösterilmelidir.

Tablo 5. Örnek Uygulama için Emniyet Gereksinimleri

4. Emniyet modelinin oluşturulması:

Bu aktivitede belirlenen emniyet gereksinimlerine uygun olarak tasarım yapılır. Tasarım süreci yinelemeli bir süreçtir. İlk olarak modeller oluşturulur ve bu modellerin bir sonraki aktivitede de anlatıldığı gibi emniyet gereksinimlerine uygun olup-olmadığı değerlendirilir. Bu değerlendirme sonuçlarına göre eğer tasarım emniyet gereksinimlerine uygun değilse modeller güncellenir. Bu yinelemeli süreç, tasarımın uygun bulunmasına kadar devam eder. Bu bölümde, örnek uygulamamızın mimari tasarımının iki versiyonu anlatılmaktadır. Birinci versiyon, emniyet gereksinimleri belirlenmeden önceki versiyondur. İkinci versiyon ise birinci versiyonun emniyet gereksinimlerine uygun olarak güncellenmesi sonucunda oluşan versiyondur. Yapılan güncellemeler ve güncellemelerin nedenleri bir sonraki aktivitede anlatılmaktadır.

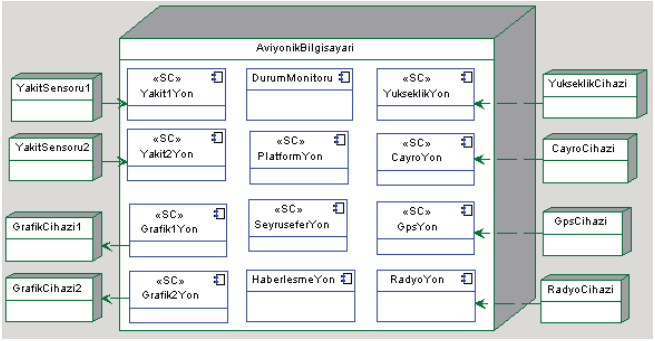


Şekil 1. Örnek uygulama mimari modeli birinci versiyon

Örnek uygulamamız için oluşturulan yazılım mimarisi modelinin birinci versiyonu Şekil 1'de verilmiştir. Birinci versiyonda bir tane yakıt sensörü ve bir tane grafik cihazı bulunmaktadır. *YakitYon,* yakıt sensörünün yöneticisidir ve yakıt sensöründen aldığı yakıt miktarı bilgisini platform yöneticisine sağlar. *YukseklikYon,* yükseklik cihazının yöneticisidir ve yükseklik cihazından aldığı yükseklik bilgisini seyrüsefer yöneticisine sağlar. *GpsYon*, GPS cihazının yöneticisidir ve GPS cihazından aldığı konum bilgisini seyrüsefer yöneticisine sağlar. *CayroYon,* cayroskop cihazının yöneticisidir ve cayroskop cihazından aldığı durum (attitude) bilgisini seyrüsefer yöneticisine sağlar. *RadyoYon,* radyo cihazının yöneticisidir ve radyo cihazından aldığı frekans bilgisini haberleşme yöneticisine sağlar. *PlatformYon* isimli yönetici yakıt miktarı bilgisini

yakıt yöneticisinden alır ve grafik yöneticisine sağlar. *SeyruseferYon* isimli yönetici yükseklik, durum ve konum bilgilerini ilgili cihaz yöneticilerinden alır ve grafik yöneticisine sağlar. *HaberlesmeYon* isimli yönetici frekans bilgisini radyo cihazı yöneticisinden alır ve grafik yöneticisine sağlar. *GrafikYon* isimli yönetici grafik cihazının yöneticisidir. Yakıt miktarı, yükseklik, durum, konum ve frekans bilgilerini ilgili yöneticilerden alır ve grafik cihazında gösterilmek üzere gönderir.

Örnek uygulamamız için oluşturulan yazılım mimarisi modelinin ikinci versiyonu Şekil 2'de verilmiştir. İkinci versiyonda sisteme emniyet gereksinimlerini sağlamak amacıyla ikinci bir yakıt sensörü ve ikinci bir grafik cihazı eklenmiştir. *Yakit1Yon* ve *Yakit2Yon* isimli yöneticiler yakıt sensörleri 1 ve 2'nin yöneticileridir. Yakıt yöneticileri, ilgili yakıt sensöründen yakıt miktarı bilgisini alır ve bu bilgileri platform yöneticisine sağlar. *Grafik1Yon* ve *Grafik2Yon* isimli yöneticiler grafik cihazları 1 ve 2'nin yöneticileridir. Yakıt miktarı, yükseklik, durum, konum ve frekans bilgilerini ilgili yöneticilerden alır ve grafik cihazlarında gösterilmek üzere gönderir. *PlatformYon* isimli yönetici her iki yakıt yöneticisinden gelen yakıt miktarını alacak şekilde güncellenmiştir. Şekil 2' de emniyet kritik olan modüller SC (Safety Critical) stereotipi ile işaretlenmiştir. Böylece emniyet kritik modüller ve emniyet kritik olmayan modüller ayırt edilebilmektedir.



Şekil 2. Örnek uygulama mimari modeli ikinci versiyon

5. Emniyet gereksinimlerine göre değerlendirme yapılması:

Son aktivite ile oluşturulan mimari tasarımın emniyet gereksinimlerine uygun olup-olmadığı değerlendirilir. Örnek uygulamamızın ilk versiyonunda bir tane yakıt sensörü ve bir tane grafik cihazı bulunduğu için *EG1* ve *EG3* numaralı emniyet gereksinimleri sağlanamamaktadır. Bu nedenle tasarıma birer tane daha yakıt sensörü ve grafik cihazı eklenerek güncelleme yapılmış ve ikinci versiyon oluşturulmuştur. Böylece *EG1* ve *EG3* gereksinimleri yedeklilik taktiği kullanılarak ikinci versiyonda sağlanmaktadır.

İkinci versiyondaki platform yöneticisi iki ayrı yakıt yöneticisinden yakıt miktarı bilgisini alacak ve *EG2* gereksiniminde belirtildiği gibi iki yakıt miktarını karşılaştıracak ve aralarındaki fark belli bir miktardan fazla ise yakıt miktarı gösterilmeyecek ve yakıt miktarı ile ilgili bir uyarı verilecek şekilde güncellenmiştir. Birinci versiyonda böyle bir karşılaştırma yapılamadığı için birinci versiyon *EG2* gereksinimini yerine getirememektedir. Bu güncelleme ile *EG2* gereksinimi de ikinci versiyonda gerçekleştirilmiştir.

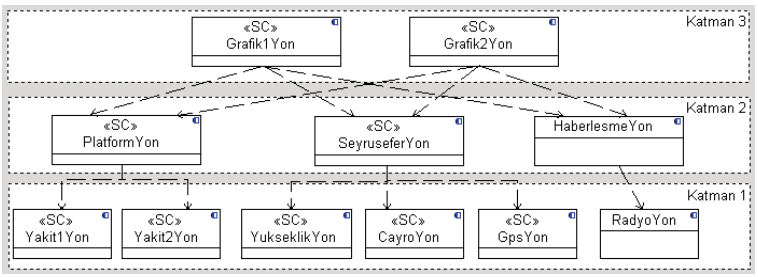
İkinci versiyonda modüllerin çalışmasını kontrol eden bir hata kontrol yöneticisi (*DurumMonitoru*) de eklenmiştir. Bu yönetici, diğer modüllerin düzgün çalışıp-çalışmadığını kontrol etmektedir ve çalışmayan modülleri yeniden başlatabilmektedir.

6. Görünümlere uygulanması:

Bu bölümde, emniyet perspektifinin örnek uygulamamıza uygulanması ile oluşan stiller anlatılmaktadır. Bu stillerin hangileri olduğu kısa bir açıklama ile beraber Tablo 6'da verilmiştir. Bu stiller bir önceki bölümde anlatılan mimari tasarımın ikinci versiyonuna göre oluşturulmuştur.

Stil	**Açıklama**
Ayrıştırma	Emniyet kritik modüller ve emniyet kritik olmayan modüller işaretlenmiştir(bkz. Şekil 3)
Kullanım	Emniyet kritik modüllerin kullandığı modüller gösterilmiştir. (bkz. Şekil 3)
Katmanlı	Her katmanda emniyet kritik modüller ve emniyet kritik olmayan modüller gösterilmiştir. (bkz. Şekil 3)
Veri Akışı	Emniyet kritik bir veri olan yakıt miktarı bilgisinin ele alındığı emniyet kritik modüller gösterilmiştir. (bkz Şekil 4)
Konuşlandırma	Emniyet kritik bir veri olan yakıt miktarı bilgisinin iki ayrı sensörden alınması ve iki ayrı göstergeye gönderilmesi. (bkz Şekil 2)

Tablo 6. Emniyet Perspektifinin Uygulanması

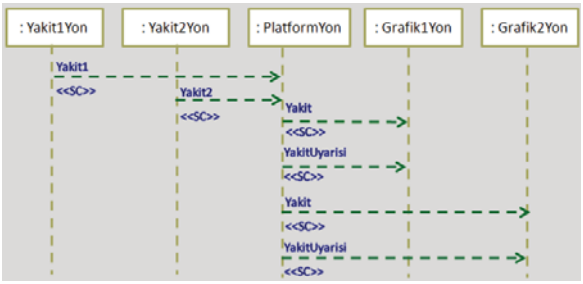


Şekil 3. Örnek Uygulama için Ayrıştırma, Kullanım ve Katmanlı Mimari Stili

Örnek uygulamamız için modül görünümlerinden ayrıştırma, kullanım ve katmanlı stillere göre olan çizim Şekil 3'de verilmektedir. Sayfa limiti nedeniyle bu üç stil bir arada gösterilmiştir. Şekil 3'deki modüller arasındaki bağlantılar ve katmanlar gösterilmediğinde ayrıştırma stili elde edilmektedir. Ayrıştırma stilinde emniyet kritik modüller ve emniyet kritik olmayan modüller "SC" stereotipi ile işaretlenmiştir. Şekil 3'de modüller arasındaki bağlantılar kullanım durumunu gösteren bağlantılardır ve bu bağlantılar ile modüller bir arada gösterildiğinde kullanım stili elde edilmektedir. Bu stilde hangi emniyet kritik modüllerin hangi emniyet kritik modülleri kullandığı görülmektedir. Emniyet kritik modüller sadece emniyet kritik modülleri kullanabilir. Örnek uygulamamızda üç katmanlı bir mimari bulunmaktadır. Şekil 3'de gösterildiği gibi birinci katmanda cihaz/sensör yöneticileri, ikinci katmanda platform, seyrüsefer ve haberleşme yöneticileri ve üçüncü katmanda da grafik yöneticileri bulunmaktadır. Katman bilgileri de görünüme eklendiğinde katmanlı stil elde edilmektedir. Bu stilde hangi katmanlarda hangi emniyet-kritik modüllerin olduğu görülmektedir.

Bileşen ve bağlantılar görünümünden veri akışı stilinin örnek uygulamamız için kullanım örneği Şekil 4'de verilmektedir. Bu çizimde yakıt miktarı bilgisinin veri

akışı gösterilmektedir. Yakıt miktarı bilgisi emniyet kritik bir veri olduğu için bu verinin işlendiği modüllerin de emniyet kritik modül olması gerekmektedir. Ayrıca veri akışını gösteren oklar da emniyet kritik veri olduğunu belirten "SC" stereotipi ile işaretlenmiştir.



Şekil 4. Örnek Uygulama için Veri Akışı Mimari Stili

Tahsis görünümünden konuşlandırma stili, emniyet modelinin oluşturulması aktivitesinde de anlatılmakta ve Şekil 2'de verilmektedir. Konuşlandırma stili emniyet kritik modüller için gerekli olan donanım elemanlarını belirtir. Şekil 2'de emniyet kritik bir veri olan yakıt miktarı verisinin alındığı yakıt sensörleri ve ilgili yakıt sensörü ile iletişim kuran yakıt sensör yöneticileri diğer cihazlar ve yöneticiler ile beraber gösterilmiştir.

6 İlgili Çalışmalar

Daha önceki çalışmalarımızda emniyet ve kurtarılabilirlik (recoverability) kalite ilgileri için mimari bakış açıları tanımlanmıştır. Kurtarılabilirlik ilgisi için yapılan çalışmada [15] mimarinin farklı şekillerde ayrıştırılmasını gerektirdiği için, bakış açısı tanımında bu ayrıştırmayı sağlayabilecek ilgili mimari elemanlar ve ilişkiler tanımlanmıştır. Emniyet ilgisi için yapılan çalışmada [2][4] ise emniyet-kritik sistemlerin tasarımını analiz etmek ve desteklemek için bir mimari çerçeve sunulmuştur. Böylece farklı kalite ilgileri tasarım aşamasının başlangıcından itibaren mimari seviyede ele alınabilmiş ve tasarım ile ilgili kararlar verilirken farklı tasarım alternatifleri değerlendirilmiştir.

Tasarımı yazılım emniyeti açısından değerlendirmek ve analiz etmek için literatürde bazı teknikler sunulmuştur. Hata ağacı analizi (fault tree analysis) bu yöntemlerden birisidir. Hata ağacı analizinde tasarımdaki hangi hataların sistemde hangi hatalara neden olabileceği analiz edilir ve mantıksal kapılar kullanılarak gerçekleştirilir [8]. Hata modu ve etkileri analizi [10] (failure mode and effects analysis) kullanılan diğer yöntemlerden birisidir. Bu yöntem ile tasarımdaki olası zayıflıkların belirlenmesi amaçlanmıştır. Sistemdeki bileşenler analiz edilerek olası hatalar ve bunların sebepleri tespit edilir.

Yazılım emniyeti planlama ve tasarımı için birçok standart tanımlanmıştır. RTCA DO-178C [12], MIL-STD-882D [9], IEC 61508 [5] bu standartlardan bazılarıdır. Bu standartlarda emniyet ilgisi için gerekli olan seviyeyi tanımlanmıştır. Fakat emniyet-kritik sistemlerin tasarımı belirgin bir şekilde ele alınmamıştır.

7 Sonuç

Emniyet-kritik sistemlerin tasarımında emniyet ilgisinin açık bir şekilde ele alınması ortaya çıkabilecek tehlikeli ve riskli durumların ortadan kaldırılması ya da azaltılması açısından önemlidir. Emniyet perspektifi, emniyet-kritik sistemlerdeki emniyet ilgilerini adresleyerek, sistem ile alakalı tasarım ve analiz kararlarında yardımcı olmaktadır. Bu çalışmada emniyet perspektifi Görünümler ve Ötesi mimari çerçevesine uygulanarak, emniyet-kritik sistemlerin tasarımı aşamasında ortaya konulan ilgili mimari görünümlerde emniyet ilgisinin ele alınması sağlanmıştır. Perspektif uygulaması gerçek endüstriyel bir çalışma üzerinde örneklendirilmiştir. Bu çalışma emniyet mühendislerine ve mimarlarına emniyet ile ilgili kararlar alırken yardımcı olmasının yanında emniyet ilgisinin yazılım geliştirme aşamasının başlangıcından itibaren değerlendirmeye alınarak mimari açıdan analiz edilmesini de sağlamaktadır.

Kaynaklar

1. Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Nord, R., Stafford, J.: Documenting Software Architectures: Views and Beyond, Addison-Wesley, Boston (2003)
2. Gurbuz, H. G.: Architecture-Driven Fault-Based Testing for Software Safety, Bilkent University, Turkey (2014)
3. Gurbuz, H. G., Tekinerdogan, B., Pala Er, N.: Safety Perspective for Supporting Architectural Design of Safety-Critical Systems, In: Proc. of 8th European Conference on Software Architecture (ECSA 2014), Agust 2014
4. Gurbuz, H. G., Pala Er N., Tekinerdogan, B.: Architecture Framework for Software Safety, In: Proc. of 8th System Analysis and Modeling Conference, September (SAM 2014), September 2014
5. IEC 61508 – Functional Safety of Electrical /Electronic/ Programmable Electronic Safety-Related Systems. International Electrotechnical Commission, (1998)
6. [ISO/IEC 42010:2007] Recommended practice for architectural description of software-intensive systems (ISO/IEC 42010).
7. Kruchten, P.: The 4+1 View Model of Architecture, IEEE Software, 12(6):42–50 (1995)
8. Leveson, N.G. , Harvey, P.R.: Analyzing Software Safety, IEEE Transactions on Software Engineering 9 (5), 569-579 (1983)
9. MIL-STD-882D, Standard Practice for System Safety, Department of Defense (2000)
10. Pataricza, A., Majzik, I., Huszerl, G., Várnai, G.: UML-based design and formal analysis of a safety-critical railway control software module. In : Proc. of Symposium Formal Methods for Railway Operation and Control Systems (FORMS03), pp 125–132, Budapest (2003)
11. Rausand, M., Hoylan, A.: System Reliability Theory, Models, Statistical Methods, and Applications, Wiley, USA (2004)
12. Rozanski, N., Woods, E.: Software Architecture Systems Working with Stakeholders Using Viewpoints and Perspectives, First Edition, Addison-Wesley, (2005)
13. RTCA DO-178C Standard, Software Considerations in Airborne Systems and Equipment Certification
14. Software Safety Guide Book, NASA Technical Standard (2004)
15. Sözer, H., Tekinerdogan, B., Aksit, M.: Optimizing Decomposition of Software Architecture for Local Recovery, Software Quality Journal, Vol. 21, No:2, pp. 203-240 (2013)
16. Wasilewski, M., Hasselbring, W., Nowotka, D.: Defining requirements on domain-specific languages in model-driven software engineering of safety-critical systems (2013)
17. Yu, G. and Wei Xu, Z.: Model-Based Safety Test Automation of Safety-Critical Software. pp. 1-3 (2010)