

Robot Tasarımı İçin Geliştirilen Petri Ağları ile Davranış Modellemesi Yaklaşımının Bir Malzeme Taşıma Robotu Modeline Uygulanması

Macit ARAZ¹, Zühal ERDEN^{2*}

¹ Mikroelektromekanik Sistemler Araştırma ve Uygulama Merkezi (MEMS), ODTÜ, Ankara, Türkiye

² Mekatronik Mühendisliği Bölümü, Mühendislik Fakültesi, Atılım Üniversitesi, Ankara, Türkiye

¹ macit.araz@mems.metu.edu.tr ^{2*} zuhal.erden@atilim.edu.tr

(Geliş/Received: 19/06/2018;

Kabul/Accepted: 28/02/2019)

Özet: Bu makalede sunulan çalışmada, mekatronik ürünlerin ve robotların davranışlarının kavramsal tasarım aşamasında modellenmesi, bilgisayarda benzetimi ve fiziksel olarak gerçekleştirilmesinden oluşan sistematik bir yöntemin geliştirilmesi, böylece bilgisayar destekli kavramsal robot tasarımına katkı sağlanması amaçlanmıştır. ‘Davranış Tabanlı Kavramsal Tasarım (DTKT)’ olarak adlandırılan bu yöntemde, tasarlanacak robotun istenen davranışı öncelikle fiziksel elemanlardan bağımsız olarak Petri ağları ile modellenmekte ve bilgisayarda benzetimi yapılmaktadır. Daha sonra robot davranışı, “masa üzeri tasarım modeli” adı verilen dağıtık bir fiziksel yapıda da benzetilmektedir. Böylece tasarımın erken aşamasında istenen davranışın gerçekleştirilmesi mümkün olmaktadır. Geliştirilen yöntem laboratuvarında beş adet robot modelinin davranış tabanlı kavramsal tasarımına uygulanmış olup, makalede bu uygulamalardan biri olan malzeme taşıma robotu örneği sunulmuştur.

Anahtar kelimeler: Davranış tabanlı kavramsal tasarım, robot tasarımı, Petri ağları, masa üzeri tasarım, kavramsal tasarım modellemesi.

Behaviour Based Conceptual Design and Application to Robot Design

Abstract: The aim of this study is to develop a systematic method for modeling, computer simulation and physical implementation of the behavior of mechatronic products and robots during the conceptual design phase, contributing to computer-aided conceptual robot design. The method is called ‘Behavior Based Conceptual Design (BBCD)’ and it involves modelling of the intended behaviour of a robot at conceptual design using Petri nets, then simulation of the behaviour both on the computer and on a distributed physical structure called the ‘desktop design model’. Thus, it is possible to realize the desired behavior in an early design stage. The developed method has been applied to the behavior-based conceptual design of five robot models in the laboratory. This paper presents one of these case studies, namely an AGV model for material handling.

Key words: Behaviour based conceptual design, robot design, Petri nets, desktop design, conceptual design modelling.

1. Giriş

Ürün geliştirme sürecinde, mühendislik yaratıcılığının koyduğu katkı bakımından kavramsal tasarım aşaması özel bir önem taşımaktadır ve ürünlerin yaşam döngüsü maliyetinin büyük bir bölümü kavramsal tasarım aşamasında belirlenmektedir [1]. Yapılan araştırmalar, tasarım sürecindeki tüm aktivitelerin %20 sini oluşturan kavramsal tasarımın ürün maliyeti ve kalitesine etkisinin %80 dolayında olduğunu göstermiştir [2]. Bu nedenle kavramsal tasarımın süresini ve maliyetini azaltmaya yönelik olarak yapılan araştırmalar son yıllarda büyük oranda artmıştır. Kavramsal tasarımda süre ve maliyet açısından tasarruf sağlayabilecek bilgisayar desteği genellikle CAD sistemleriyle gerçekleştirilmeye çalışılmaktadır. Ancak günümüzde oldukça gelişmiş CAD sistemleri bulunmakla birlikte, bu sistemler temel olarak şekillendirme tasarımı ve ayrıntılı tasarım aşamalarında, çoğunlukla grafiksel gösterim, geometrik modelleme ve analiz amaçlı olarak kullanılmaktadır. Tasarımın erken ve soyut aşaması olan kavramsal tasarımda ise CAD sistemlerinin kullanımı kolay değildir. Hatta bazı araştırmacılar temel varsayım olarak kavramsal tasarımın tümüyle otomasyonunun mümkün olamayacağını kabul etmekte, ancak yine de CAD sistemlerinin tasarımcıların erken aşamalarda yaratıcı tasarımlar yapması konusunda destek olacağını düşünmektedirler [2]. Kavramsal tasarımın otomasyonu, bu konuda sistematik yöntemlerin geliştirilmesiyle kısmen gerçekleştirilebilir.

Bu makalede anlatılan çalışmada kavramsal tasarım aşamasındaki bir mekatronik sistemin istenen operasyonel davranışının modellenmesini, bu davranışın bilgisayar ortamında ve fiziksel olarak masa üzerinde benzetimini içeren sistematik bir süreç geliştirilmiştir. ‘Davranış Tabanlı Kavramsal Tasarım (DTKT)’ olarak adlandırılan bu süreçte davranış modellemesi için DEVS (Discrete Event System Specification-Ayrık Olay Sistem

* Sorumlu yazar: zuhal.erden@atilim.edu.tr. Yazarların ORCID Numarası: ¹ ORCID, ² 0000-0002-4860-4271

Spesifikasyonu) [3] ve Petri ağları [4-7] yöntemleri kullanılmıştır. DTKT süreci laboratuvar ortamında gerek boyut, gerekse karmaşıklık açısından uygun görülen 5 farklı eğitim robotuna uygulanmıştır. Makalede örnek olarak bu uygulamalardan biri olan malzeme taşıma robotu modeli anlatılmıştır.

Makalenin bundan sonraki bölümü şöyle düzenlenmiştir: Bölüm 2’de mekatronik ürünlerin kavramsal tasarımında davranış modellemesi konusunda yapılan çalışmalar özetlenmiş ve değerlendirilmiştir. Bölüm 3’de, bu çalışmada geliştirilen süreç anlatılmış, Bölüm 4’de uygulama örneklerinden biri olan malzeme taşıma robotu için davranış tabanlı tasarım yaklaşımının uygulanması açıklanmıştır. Son bölümde ise bu çalışmadan elde edilen sonuçlar sunulmuştur.

2. Literatür Özeti

Mühendislik sistemleri enerji, malzeme ve/veya bilgi akışının performanslarını belirlediği sistemler olarak değerlendirildiğinden [8], bu akışların matematiksel modellerine dayalı performans değerlendirmeleri ayrıntılı tasarım düzeylerinde yapılabilmektedir. Ancak bu değerlendirmelerin, tasarımın en önemli değerlendirme aşaması olan kavramsal tasarımın soyut düzeyinde yapılmasını sağlayacak modeller ve yazılım araçları oldukça kısıtlıdır. Kavramsal tasarım aşamasındaki temel ihtiyaçlardan biri tasarlanmakta olan sistemin fonksiyonunu yerine getirebilecek davranışını, sistemin fiziksel modelini/prototipini geliştirmeksizin modelleyebilmektir. Bu tarz bir sistematik davranış modellemesi bilgisayar destekli kavramsal tasarım sistemlerinin geliştirilmesi için de önemli bir altyapı niteliğindedir.

Mekatronik ürünlerin kavramsal tasarımında davranış modellemesi, araştırmacılar tarafından farklı bakış açılarından ele alınarak incelenmiştir. Genel olarak bir sınıflandırma yapılırsa çalışmalar iki gruba ayrılmaktadır. Bunlardan birincisi çeşitli modelleme yöntemlerinin soyut düzeyde incelendiği/önerildiği ve bazı örneklerde uygulandığı teorik ve metodolojik çalışmalar, diğeri ise tek bir ürün ya da belli bir ürün grubunun ele alınarak modellemelerin yapıldığı çalışmalardır.

Birinci grup çalışmalar içinde fonksiyonel modelleme ön plana çıkmaktadır ve özellikle disiplinlerarası sistemlerin (mekatronik sistemler) tasarım süreçlerinde ortak bir sistematik tasarım dili ve yapısı oluşturmak bakımından önemlidir [9,10]. Mekatronik tasarım konusunda gerçekleştirilmiş modelleme ve benzetim çalışmaları içinde, Bond çizgesi ontolojisine dayalı, enerji akışını temel alan, fiziksel eleman düzeyinde modelleme içeren ve nesne-tabanlı bir modelleme dili olan Modelica [11] kullanılarak geliştirilen Schemebuilder [12] ve MathModelica [13] yazılımları önemlidir. Bu kapsamda ayrıca, mekatronik sistemlerin tasarım sürecinin otomasyonuna yönelik olarak geliştirilen ve Bond çizgeleri ile genetik programlama yöntemlerinin beraberce kullanılmasına dayanan metodolojik çalışmalar [14-16] bulunmaktadır. Mekatronik sistemlerin soyut düzeyde modellenmesinde kullanılan bir başka önemli yöntem, sistem davranışının birtakım durumlar ve durumlar arasındaki geçişlerden oluşan durum-geçiş yapılarıyla modellenmesidir. Bu amaçla UML (Unified Modeling Language-Birleştirilmiş Modelleme Dili) modelleme tekniği ve Bond çizgelerinin birlikte kullanılmasına yönelik araştırmalar yapılmıştır. UML kullanılan araştırmalarda veri alış-verişi yapan yazılım elemanları ile enerji alış-verişi yapan fiziksel elemanlar modellenerek tümleşik mekatronik tasarımda kullanılacak çeşitli veri tabanları oluşturulmuştur [17]. UML yaklaşımı özel bir uygulama olarak SysML modelleme dili aracılığıyla mekatronik sistemlerde kullanılmış [18], ayrıca elektrikli araçların kavramsal tasarım aşamasında modellenmesine de uygulanmıştır [19]. Genel olarak tüm ürünlerin, özelde ise mekatronik ürünlerin davranış modellemesinde kullanılan önemli bir yaklaşım da tasarımda kullanım süreçlerinin ve ürün-insan-çevre etkileşiminin dikkate alındığı modelleme yaklaşımlarıdır [20-22].

Belirli bir ürün ya da ürün grubu modellenmesine odaklanan ikinci grup çalışmalar; ürünlerin davranışlarıyla ilgili çeşitli kuralların belirlenmesi [23] endüstriyel robotların fonksiyonel tasarımı için uzman sistem geliştirilmesi [24] robotik görev modellemesi ve analizinde Petri ağlarının kullanılması [25] gibi çalışmaları içermektedir. Petri ağları, tasarım aşamasındaki sistemlerin soyut kavramsal düzeyde mantıksal davranışının bilgi akışına dayalı modellemesi ve benzetimi için de kullanılmış ve bu amaçla teorik bir model geliştirilerek mekatronik tasarıma uygulanmıştır [26]. Bu modelleme altyapısı daha sonra lisans eğitimi düzeyinde robot tasarımı [27-29] ve biyoesinlenmiş robot tasarımına uygulanmıştır [30, 31].

Literatürde bulunan çalışmalar incelendiğinde, mekatronik sistemlerin kavramsal tasarımına yönelik bütüncü bir sistematik bir yöntemin tam olarak geliştirilmediği anlaşılmaktadır. Birinci gruptaki çalışmalar bu alanda metodolojik olarak önemli katkılar sağlamakla birlikte mekatronik sistem davranışlarının modellenmesi konusunda çalışmalar içermemektedir. Bu çalışmalar ağırlıklı olarak fiziksel elemanların belli olduğu, kavramsal tasarımın nispeten daha detay aşamalarındaki modellemeler için uygun yapıdadır. Bu çalışmaların bir başka eksik yönü de modellenmenin sistemin fiziksel olarak gerçekleşmesi ile bütünleştirilmemiş yaklaşımlar olmasıdır. İkinci gruptaki çalışmalar ise daha çok ürün temelli olup, genel bir sistematik oluşturma ihtiyacına cevap verememektedir. Literatürdeki çalışmalarda bulunan bu eksikliklerin değerlendirilmesi sonucunda, bu makalede

anlatılan mekatronik ürünlerin kavramsal tasarım aşamasında uygulanabilecek sistematik bir yöntem geliştirilmiştir. Geliştirilen yöntem tasarlanacak sistemin operasyonel davranışının önce DEVS ve Petri ağı ile modellenerek bilgisayar ortamında benzetimini; daha sonra bu davranışın masa üzerinde ve bir çeşit pano yapısı içinde, robotun geometrik ve dinamik parametrelerinden bağımsız bir şekilde, fiziksel olarak elde edilmesini içermektedir. Bilgi akışına dayalı davranış modellemesi ile fiziksel sistem davranışının modellenmesi yaklaşımlarını birlikte kullanan bu yöntem Bölüm 3’de anlatılmıştır.

3. Kavramsal Mekatronik Tasarımda Davranış Tabanlı Modelleme

Mekatronik ürün geliştirme süreci için önerilen DTKT yaklaşımı, tasarlanacak olan mekatronik sistemin istenen operasyonel davranışının soyut düzeyde DEVS ile kavramsal gösterimi, Petri ağı ile modellenmesi ve benzetimi, davranışsal değerlendirme ve seçilen davranışın masüstü tasarım modeli biçiminde fiziksel olarak gerçekleştirilmesine dayanmaktadır. Genel bir mekatronik sistemin DEVS modeli, sistemin çevresiyle olan ilişkisini de içerecek biçimde en soyut düzeyde geliştirilmiş ve Şekil 1’de gösterilmiştir. Burada mekatronik sistemin çevresiyle ilişkisi iki yönlüdür. Bu etkileşimin bir yönü çevrede meydana gelen değişimlere göre sistemin davranışının değişmesi, diğer yönü ise mekatronik sistemin motorik hareketi sonucu çevrede oluşan durum değişimidir. Mekatronik sistemlerin operasyonel davranışları kavramsal tasarım düzeyinde çeşitli durumlar ve durumlar arası geçişlerle tanımlanabilir. Bu çalışmada operasyonel davranışın modellenmesinde, Denklem 1’de verilen atomik DEVS modelinden (M) [3] yararlanılmıştır.

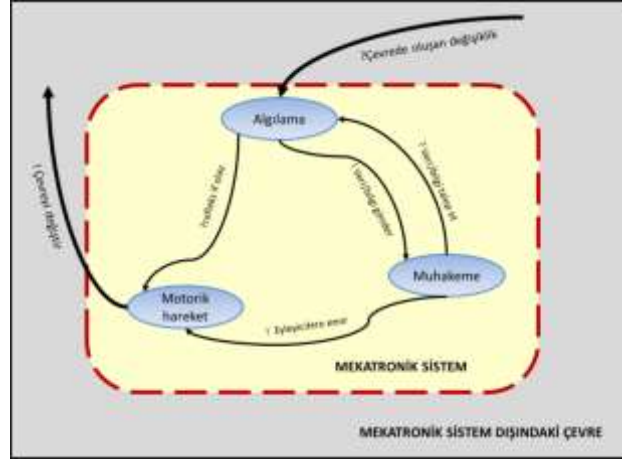
$$M = \langle S, X, Y, s_0, \tau, \delta_x, \delta_y \rangle \quad (1)$$

Yukarıdaki ifadede, S sonlu durumlar kümesini, X sonlu girdi olayları kümesini, Y sonlu çıktı olayları kümesini, $s_0 \in S$ sistemin başlangıç durumunu göstermektedir. $\tau: S \rightarrow Q [0, \infty]$ her sistem durumunun süresini ifade eden zaman ilerleme fonksiyonunu belirtmekte olup, $Q [0, \infty]$ pozitif reel sayılar kümesidir. $\delta_x: S \times X \rightarrow S \times \{0, 1\}$ harici geçiş fonksiyonu olarak tanımlanır ve bir girdi olayının sistemin durumunu nasıl değiştirdiğini ifade etmek için kullanılır. $\delta_y: S \rightarrow Y \Phi \times S$ çıkış ve dahili geçiş fonksiyonu olup $Y \Phi = Y \cup \{\Phi\}$ olarak tanımlanır. Burada $\Phi \notin Y$ ‘sessiz olay’ olarak adlandırılır. Çıkış ve dahili geçiş fonksiyonu bir sistem durumunun kendi içinde (dahili olarak) değişimini ve bu değişim sırasında bir çıktı olayı üretmesini ifade eder. Grafikselleştirimde girdi olayları ‘? olay adı’, çıktı olayları ise ‘! olay adı’ biçiminde gösterilir.

Bir mekatronik sistemin davranış tabanlı modellemesi, öncelikle sistemin istenen bir işlevi nasıl yerine getireceğini ifade eden operasyonel davranışın formal olmayan tanımıyla başlar ve bu tanım yukarıda verilen formal DEVS yapısı ile ifade edilir. Bir mekatronik sistemin operasyonel davranışı en genel düzeyde ‘algılama’, ‘muhakeme’ ve ‘motorik hareket’ olmak üzere üç temel sistem durumu ile ifade edilir. Sistem ‘algılama’ durumundayken çevresinden veri toplar ve bu verileri işleyerek kullanılabilir bilgiye dönüştürür. ‘Muhakeme’ durumu, sistemin bilgileri kullanarak değerlendirmesi ve bu değerlendirme sonucu gerekli kararları vererek çevresinde meydana gelen değişikliklere göre fiziksel (motorik) davranış geliştirmek için altyapı oluşturmasını ifade eden durumdur. ‘Motorik hareket’ ise gerek verilen kararlar doğrultusunda, gerekse çevrede meydana gelen değişikliklere bir refleks olarak fiziksel davranış ortaya konmasını belirten durumdur. Sistemin operasyonuna başladığı andan itibaren çevreden veri toplaması gerektiğinden modelin başlangıç durumu ‘algılama’ olarak varsayılmıştır. Mekatronik sistemin yukarıda tarif edilen genel operasyonel davranışı DEVS yapısı ile formal olarak Denklem 2 ve Denklem 3’de gösterilmiştir.

$$S = \{\text{Algılama, Muhakeme, Motorik Hareket}\} \quad \text{"Sistem durumları kümesi"} \quad (2)$$

$$s_0 = \text{Algılama} \quad \text{"Sistemin başlangıç durumu"} \quad (3)$$



Şekil 1. Bir mekatronik sistemin DEVS modeli.

Şekil 1'deki genel mekatronik sistem davranışı modelinde dört çeşit durum değişimi tanımlanmıştır. Sistem 'algılama' durumundayken 'muhakeme' durumuna ya da 'motorik hareket' durumuna geçebilir. 'Algılama' durumundan 'muhakeme' durumuna geçişte bir veri gönderme olayı meydana gelir ve bu olay 'algılama' durumu için bir 'çıkış olayı'dır. 'Algılama' durumundan 'motorik hareket' durumuna geçiş ise çevrede meydana gelen bir refleksif 'girdi olayı' sonucu sistemin 'muhakeme' durumuna geçmeden doğrudan 'motorik hareket' durumuna geçişini modellemek için kullanılmıştır. Üçüncü durum değişimi 'muhakeme' durumundan 'algılama' durumuna geçiştir ve bunun için sistem 'muhakeme' durumundayken çevre algılama ile ilgili veri talep eder ki bu bir 'girdi olayı' olarak tanımlanmıştır. Son olarak 'muhakeme' durumundan 'motorik hareket' durumuna bir geçiş vardır ve bu geçiş eyleycilere gönderilen emirle (çıkış olayı) gerçekleşir. Girdi ve çıkış olayları Denklem 4 ve Denklem 5'te verilen kümelerle gösterilir:

$$X = \{? \text{ veri/bilgi talep et, ? refleks hareketi}\} \quad \text{"Girdi olayları kümesi"} \quad (4)$$

$$Y = \{! \text{ veri/bilgi gönder, ! eyleycilere emret}\} \quad \text{"Çıkış olayları kümesi"} \quad (5)$$

Zaman ilerleme fonksiyonunun değeri en soyut düzeydeki bu modellemede sonsuz olarak kabul edilmiştir, bu durum bir girdi ya da çıkış olayı meydana gelmediği sürece sistemin durum değiştirmedeği varsayımına dayanmaktadır ve Denklem 6 ile ifade edilir:

$$\tau(\text{Algılama}) = \tau(\text{Muhakeme}) = \tau(\text{Motorik Hareket}) = \infty \quad (6)$$

Mekatronik sistemin sınırları içinde harici geçiş fonksiyonu (δ_x) ile ifade edilen olaylar sonucu meydana gelen durum değişimleri Denklem 7 ve Denklem 8'de tanımlanmıştır:

$$(\text{Algılama, ? refleks hareketi}) \rightarrow (\text{Motorik Hareket, 1}) \quad (7)$$

$$(\text{Muhakeme, ? veri/bilgi talebi}) \rightarrow (\text{Algılama, 1}) \quad (8)$$

Çıkış ve dahili geçiş fonksiyonu ise Denklem 9 ve Denklem 10'da tanımlanmıştır.

$$\delta_y(\text{Algılama}) = (! \text{ veri/bilgi gönder, Muhakeme}) \quad (9)$$

$$\delta_y(\text{Muhakeme}) = (! \text{ eyleycilere emret, Motorik Hareket}) \quad (10)$$

DTKT yaklaşımında Şekil 1'de gösterilen DEVS modeli, bir mekatronik sistemin operasyonel davranışını ifade eden Petri ağı modelini oluşturmak için temel olarak kullanılmaktadır. DEVS modeli ve Petri ağı modeli arasındaki ilişki kavramsal robot tasarımı açısından incelenmiştir. Petri ağı modelinin oluşturulması için, öncelikle kavramsal tasarımı yapılacak robotun günlük konuşma dilinde ifade edilen operasyonel davranışı DEVS modeli olarak gösterilir. Bu modelden yararlanılarak robotun davranışı Petri ağı ile modellenir ve bilgisayarda benzetimi

yapılır. Bu çalışmada Petri ağı modellemesi ve benzetimi için Artifex™ yazılımı [32] kullanılmıştır. Son olarak robot davranışı bir masa üzeri tasarım düzeneğinde fiziksel olarak elde edilir. Bu çalışmada, tasarlanacak robotun n adet motorik hareket durumu olduğu varsayılmış ve bunlar bir başka küme (S_{MA}) ile Denklem 11'de gösterilmiştir.

$$S_{MA} = \{MA_1, MA_2, MA_3, \dots, MA_n\} \quad (11)$$

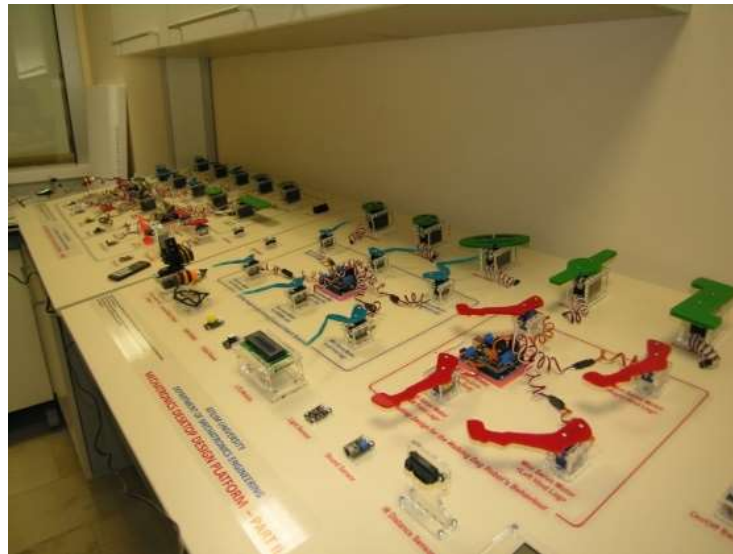
Burada MA_i ($i = 1, 2, \dots, n$) kümesinin her elemanı robotun mekatronik organlarının farklı konfigürasyonlarıyla tanımlanan bir motorik hareketi göstermektedir. Bu durum matematiksel olarak Denklem 12'de verilmiştir:

$$MA_i = \{O_{jk} \mid O_{jk}, j \text{ numaralı mekatronik organın } k \text{ numaralı konfigürasyonu}\} \quad (12)$$

Yukarıdaki ifadede $j = 1, 2, \dots, m$ (m : organ sayısı) ve $k = 1, 2, \dots, r$ (r : her organ için konfigürasyon sayısı) olarak tanımlanmıştır. Oluşturulan DEVS modeli daha sonra robot davranışının Petri ağı modelinin geliştirilmesi için kullanılmaktadır. DEVS modelindeki olaylar Petri ağı modelinde aktif elemanlar olan geçişlerle ifade edilir. Durumlar ise Petri ağı modelinde 'işaretleme' olarak gösterilmiştir. Bu çalışmada Petri ağı modelindeki 'ortamlar' robotun sensör bilgileriyle ilgili durumları ifade etmektedir.

Çalışmamızda, DEVS modelinden Petri ağı modelinin sistematik olarak elde edilmesi için, davranışı modellenen robotun her organı (O_j) geliştirilen Petri ağı modelinin en üst düzeyinde gösterilmiş ve daha alt düzeyde ayrıntılandırılmıştır. Eşleştirmenin genel yapısı şöyle özetlenebilir: Organlar (O_j) en üst düzey Petri ağındaki düğümler, organ davranışları da alt düzeylerde ayrı Petri ağları olarak modellenmiştir. Başlama, karar verme v.b durumlar ile sensörler organlara ait Petri ağlarındaki ortamlar, olaylar ise ("? veri/bilgi talep etme", "teyleyicilere emir gönderme" gibi) organlara ait Petri ağlarında geçişler olarak modellenmiştir. Bu çalışmada geliştirilen masa üzeri tasarım düzeneği fiziksel bir yapı olduğundan modellenen davranışı gerçekleştirecek elemanlar mühendislik bilgisi ve deneyimi ile seçilmiştir. Genel bir eşleştirme olarak masa üzeri tasarım düzeneğindeki sensörlerin Petri ağındaki ortamlar; eyleyici ve işlemcilerin de Petri ağındaki geçişler için kullanıldığını söyleyebiliriz.

Bu makalede anlatılan davranış tabanlı modelleme yaklaşımı laboratuvar düzeyinde 5 adet eğitim amaçlı robot için uygulanmıştır. Bunlar 'eğri izleyen robot', 'köpek robot', 'hamamböceği robot', 'kurbağa robot' ve 'malzeme taşıma robotu'dur. Bu örneklerin tümünün fiziksel modellerinden oluşan masa üzeri tasarım düzeneği Şekil 2'de görülmektedir. Bu örnek çalışmalardan biri olan malzeme taşıma robotunun modellenmesi ve masaüstü tasarımı Bölüm 4'de anlatılmıştır.

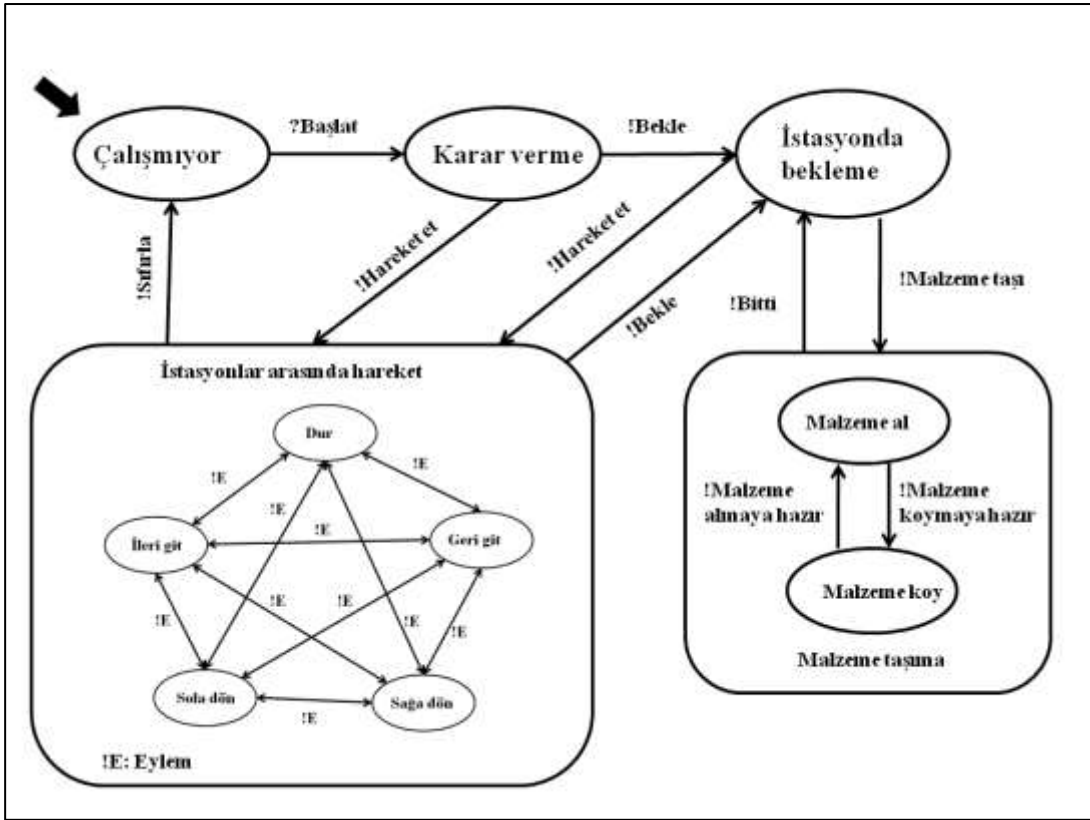


Şekil 2. Masa üzeri tasarım düzeneği.

4. Örnek Uygulama: Malzeme Taşıma Robotu

Bu örnekte, imalat hatlarında üretim tezgahları arasında malzeme ya da parça taşıyan bir malzeme taşıma robotu (AGV) için DTKT yaklaşımı uygulanmıştır. Bu örnek çalışmanın amacı DTKT yaklaşımı kullanılarak, sonlu sayıdaki iş istasyonu/imalat tezgahı arasında malzeme/parça aktaracak bir robotun davranışının bilgisayarda modellenmesi ve benzetimi, daha sonra da modellenen davranışın masaüstü tasarım düzeneğinde fiziksel olarak gerçekleştirilmesidir. Sistematik DTKT yaklaşımında öncelikle robotun operasyonel davranışının tanımlanması gerekmektedir. Bu amaçla örnek çalışmada robotun bir 'başlangıç noktası' ile dört iş istasyonu arasında önceden tanımlanmış bir fiziksel ortamda malzeme taşınması planlanmıştır.

DTKT yaklaşımına göre robotun ana fonksiyonu, ilk aşamada "karar verme ve veri işleme", "istasyonlar arasında hareket" ve "malzemelerin taşınması" olarak üç alt fonksiyona ayrılmıştır. Bu işlevsel ayrıştırmadan yararlanarak robotun operasyonel davranışını tanımlayan ve Şekil 3'de gösterilen DEVS modeli geliştirilmiştir. Bu modelde robotun tüm istasyonlara ulaştığı ve malzemeleri istasyonlar arasında sorunsuz şekilde taşıdığı varsayımı yapılmıştır. DEVS modelinde, robotun operasyonel davranışı "Çalışmıyor", "Karar Verme", "İstasyonda Bekleme", "Malzeme Taşıma" ve "İstasyonlar arasında hareket" olmak üzere 5 farklı durumdan oluşmaktadır. "Çalışmıyor", sistemin başlangıç durumu olarak tanımlanmıştır ve Şekil 3'de küçük ok işareti ile gösterilmiştir. Robot "Çalışmıyor" durumunda iken, karar verme davranışını temsil eden "Karar Verme" durumuna geçişi sağlayarak operasyonunu başlatmak için bir girdi olayına ihtiyaç duyulur. Bu girdi olayı DEVS modelinde "Başlat" komutunun alınması olarak gösterilmiştir. "Karar Verme" durumundan "İstasyonlar arasında hareket" veya "İstasyonda bekleme" durumlarına geçiş mümkündür. "İstasyonlar arasında hareket" durumunda sistemin davranışı ikinci seviye bir DEVS modeli ile temsil edilmektedir. Bu model 5 durumdan oluşmaktadır ve bu durumlar arasındaki geçişler "Eylem (E)" olarak genel bir şekilde tanımlanan çıktı olayları yoluyla sağlanır. Benzer şekilde, "Malzeme taşıma" durumu da, "Malzemeyi al" ve "Malzemeyi koy" olmak üzere iki durumdan oluşan ikinci seviye bir DEVS modeliyle gösterilmiştir. Bu iki durum arasındaki geçişler de 'Malzeme almaya hazır' ve 'Malzeme koymaya hazır' adlı çıktı olayları ile sonuçlanır. Robotun Şekil 3'deki DEVS modelinde verilen operasyonel davranışı aşağıda açıklanmıştır.



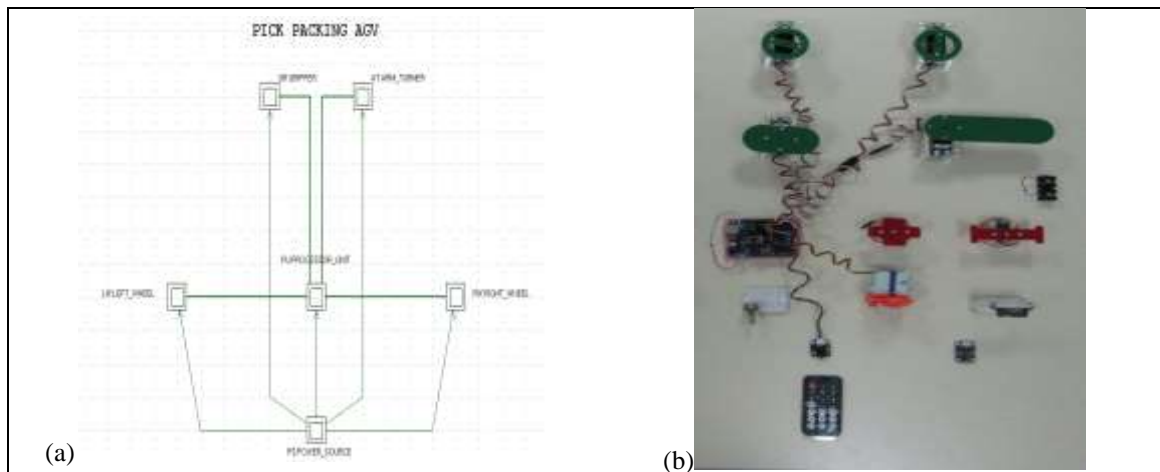
Şekil 3. Malzeme taşıma robotunun operasyonel davranışı için DEVS modeli.

Robot, başlangıçta "Çalışmıyor" durumundayken, bir görev numarası ile birlikte "Başlat" komutunu alır. Bu komutu alması sistemin "Çalışmıyor" durumundan "Karar Verme" durumuna geçişini sağlayan bir girdi olayı olarak kabul edilir. "Karar Verme" durumundayken sistem "İstasyonlar arasında hareket" durumu veya "İstasyonda bekleme" durumuna geçiş yapabilir. Bu geçişlerin gerçekleşmesi sonucu, ortaya çıkan eylemleri temsil eden, "!Hareket Et" ve "!Bekle" isimli çıktı olayları meydana gelir. Sistemin operasyonu sırasında, belli bir görevi gerçekleştirmek için "İstasyonda Bekleme", "İstasyonlar arasında hareket" ve "Malzeme taşıma" olmak üzere üç durum arasında sonlu sayıda geçişler gerçekleşir. Durumlar arasındaki geçişler çıktı olayları "!Hareket Et", "!Bekle", "!Malzeme taşı" ve "!Bitti" ile gösterilir. Bir görev bittiğinde, sistem "İstasyonlar arasında hareket" durumundan tekrar "Çalışmıyor" başlangıç durumuna geçer ve yeni bir girdi olayı (yeni bir "Başlat" komutunun alınması) için hazır hale gelir. Bu geçiş de "Sıfırla" isimli çıktı olayı ile ifade edilir. Robotun DEVS modelindeki birinci seviye durumlarla, girdi ve çıktı olaylarının tanımları Tablo 1'de verilmektedir.

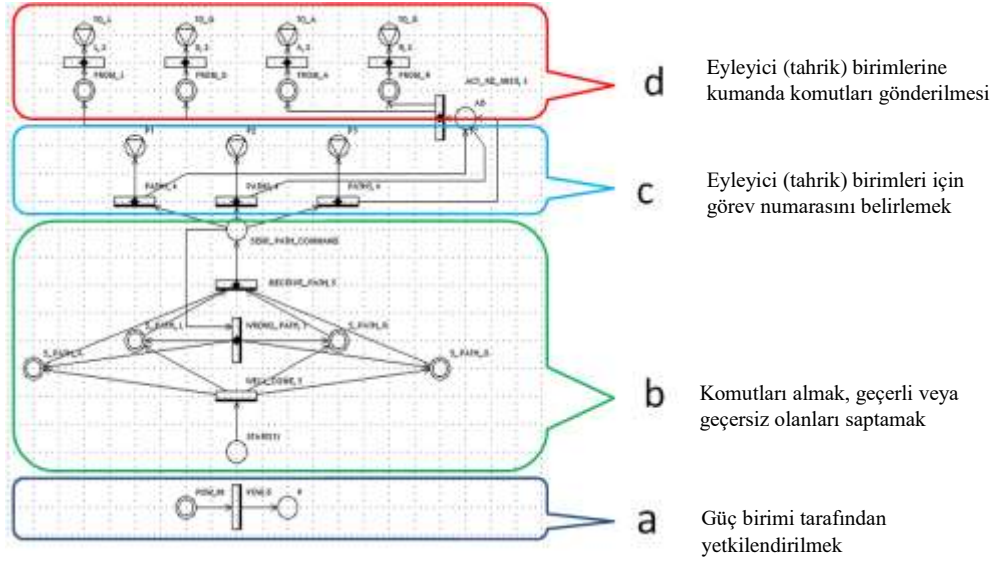
Tablo 1. Malzeme taşıma robotunun DEVS modelindeki durumlar ve olaylar.

| Durum/Olay | Tanım |
|-------------------------------------|---|
| <i>Çalışmıyor</i> | Robotun çalışmıyor olduğu durumdur. |
| <i>Karar verme</i> | Robot bu durumda iken, gelen görev numarasına göre ne yapacağına karar verir. |
| <i>İstasyonda bekleme</i> | Robotun, bir istasyonda beklediği durumdur. |
| <i>İstasyonlar arasında hareket</i> | Robotun istasyonlar arasında hareket etmekte olduğu durumdur. |
| <i>Malzeme taşıma</i> | Malzemelerin robot tarafından taşındığını gösteren durumdur. |
| <i>?Başla</i> | Robotun çalışmaya başlamasını sağlayan bir girdi olayı. |
| <i>!Sıfırla</i> | Görev bittikten sonra sistemi sıfırlayan dahili bir komutu temsil eden bir çıktı olayı. |
| <i>!Hareket et</i> | Robotun "Karar Verme" veya "İstasyonda Bekleme" durumlarından "İstasyonlar arasında hareket" durumuna geçmesiyle hareket etmeye başladığını gösteren bir çıktı olayı. |
| <i>!Bekle</i> | Robotun "Karar Verme" ya da "İstasyonlar arasında hareket" durumundan "İstasyonda bekleme" durumuna geçişini belirten bir çıktı olayı. |
| <i>!Malzeme taşı</i> | Robotun malzemeleri taşımaya başladığını ve "İstasyonda bekleme" durumundan "Malzeme taşıma" durumuna geçişini belirten bir çıktı olayı. |
| <i>!Bitti</i> | Robotun "Malzeme taşıma" durumunun sona erdiğini ve tekrar "İstasyonda bekleme" durumuna geçtiğini gösteren çıktı olayı |

DTKT yaklaşımının bir sonraki aşamasında, robotun DEVS modelinden yararlanarak, hedeflenen davranışın Petri ağı modeli ile benzetimi gerçekleştirilmiştir. Bu çalışmada Petri ağı modeli C tabanlı bir yazılım olan Artifex™ [32] kullanılarak geliştirilmiştir. Şekil 4(a)'da robotun altı farklı alt sistemden oluşan genel Petri ağı modeli gösterilmiştir. Bu alt sistemler güç kaynağı, işlemci ünitesi, sağ ve sol tekerlekler, robot kol ve tutucu birimidir. Şekil 4(b)'de hedeflenen robot davranışının fiziksel olarak gerçekleştiği masa üzeri tasarım düzeneği görülmektedir. Alt sistemlerin her birinin davranışları da ayrı ayrı Petri ağları ile modellenmiştir. Örneğin işlemci biriminin ayrıntılı Petri ağı yapısı Şekil 5'de verilmiştir.



Şekil 4. Malzeme taşıma robotunun (a)Artifex™ ortamında davranış modeli (b) masa üzeri tasarımı.



Şekil 5. İşlemci biriminin Petri ağı yapısı.

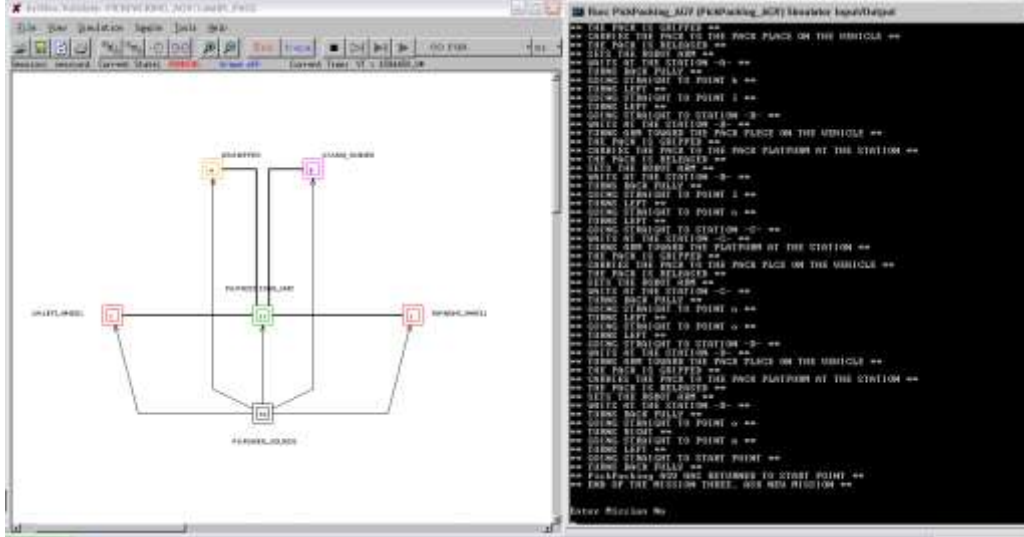
Artifex™ yazılımının benzetim özelliği kullanılarak, bilgisayar ortamında robotun Petri ağı ile modellenen operasyonel davranışının benzetimi gerçekleştirilmiştir. Benzetime ait ekran görüntüsü Şekil 6'da gösterilmiştir. Malzeme taşıma robotunun hedeflenen operasyonel davranışının modellenmesi ve benzetiminden sonra, davranış tabanlı kavramsal tasarımın son aşaması olan masa üzeri tasarım gerçekleştirilmiştir. Şekil 4(b)'de gösterilen masa üzeri tasarım, robotun modellenen davranışının, geometrik ve dinamik parametrelerden bağımsız olarak dağıtılmış bir fiziksel yapıda gerçekleşmesini sağlamaktadır. Robotun masa üzeri tasarımı, sol ve sağ tekerlek olarak iki tekerlek, bir IR uzaktan kumanda birimi, bağlantıları olan iki servo motor ve bir tutucu kullanılarak oluşturulmuştur. Güç, ayarlanabilir bir güç kaynağı cihazından harici olarak tedarik edilmektedir. Sağa ve sola dönme düz hareket, öne ve arkaya dönük iki tekerlek vasıtasıyla fiziksel olarak gerçekleştirilir. Bir servo motor, robot kolunu zemine paralel olarak (X-Y düzlemi olarak düşünebilir) ve başka bir servo motor kolu Z ekseninde hareket ettirir. Masa üzeri tasarım, malzemeleri tutma ve bırakma için mini bir servo motor kullanarak çalışan bir mekanik tutucu içerir. Sistem, bir IR uzaktan kumanda kullanarak komutlar ekleyerek görev yapmaya ve çalışmaya başlar. Malzeme taşıma robotunun masa üzeri tasarımı fiziksel olarak da, modellenen davranış gerçekleştirdiği değerlendirilmiştir. Böylece DTKT sürecinde gerekli tüm aşamalar örnek uygulama üzerinde tamamlanmış; hedeflenen operasyonel davranışın modellenmesi ve masa üzeri tasarım yapısında fiziksel gerçeklemesine kadar olan sistematik yöntem bu örnekte uygulanmıştır.

Bu örnekte görüldüğü gibi DTKT, mekatronik ürünlerin ve robotların erken dönem kavramsal tasarımında kullanılabilir özgün bir yöntemdir. Bu yöntemle, tasarlanacak robotun planlanan operasyonel davranış sistematik ve kısmen bilgisayar destekli bir metodoloji kullanarak öncelikle fiziksel elemanlardan tamamen bağımsız bir şekilde modellenmekte ve bilgisayarda benzetimi yapılmaktadır. Daha sonra, robotun davranışı bir masa üzeri tasarım modeliyle fiziksel olarak da gerçekleştirilmektedir. Bu yöntemin kullanılmasıyla kavramsal robot tasarımında istenen davranış sağlayabilecek alternatif çözümlerin, gerek bilgisayarda gerekse fiziksel olarak sistematik bir şekilde değerlendirilmesi mümkün olabilecektir. Bu yaklaşım kavramsal mekatronik tasarımda kullanılmak üzere ileride geliştirilebilecek bilgisayar destekli tasarım araçları için bir temel oluşturabilecektir.

5. Sonuçlar ve Tartışma

Küresel rekabet ortamında giderek artan bir önem taşıyan yenilikçi ve özgün ürünlerin hızla geliştirilmesi ihtiyacı, bu alanda katkı sağlayacak tasarım yöntemlerinin de geliştirilmesine öncülük etmektedir. Bu nedenle tasarım araştırmalarında, yeni ürün tasarımında belirleyici aşama olan kavramsal tasarımın bilgisayar desteği ile yapılması ve kısmen otomasyonuna yönelik yaklaşımlar önem kazanmıştır. Bu tür metodolojik çalışmalar klasik mühendislik disiplinleri için oldukça fazladır ve başarılı sonuçlar vermiştir. Giderek artan bir önem kazanan mekatronik sistemlerin tasarımında farklı mühendislik disiplinlerinin tümleştirilmesi gerekmektedir. Kendine özgü bir felsefesi olan mekatronik sistem tasarımında kavramsal tasarım aşamasının bilgisayar destekli olarak yapılması bu sistemlere özgü ve disipline bağlı olmayan sistematik tasarım yöntemlerinin geliştirilmesiyle

mümkün olabilecektir. Bu makalede sunulan çalışmada kavramsal tasarım aşamasında mekatronik sistemlerin davranışlarının modellenmesi ve benzetimi için geliştirilen DEVS ve Petri ağı modellemesine dayalı sistematik bir yapı anlatılmıştır. Bu metodoloji temel olarak robot tasarımında istenen davranışın DEVS ve Petri ağı ile modellenmesi, bu davranışın bilgisayar ortamında benzetimi ve masa üzeri tasarımda fiziksel olarak gerçekleştirilmesinden oluşmaktadır.



Şekil 6. Malzeme taşıma robotu Petri ağı modelinin benzetiminden bir ekran görüntüsü.

Bu çalışmanın sonuçları ışığında planlanan gelecek çalışmalar şöyle özetlenebilir. *Sistematik davranış tabanlı modelleme yaklaşımı için bilgisayar otomasyonu:* Sürecin aşamaları arasındaki geçişlerin sistematik olmakla birlikte tasarımcı tarafından gerçekleştiriliyor olması esneklik sağlamakta ancak daha fazla zaman almakta ve insan (tasarımcı) hatalarına açık olmaktadır. Bu nedenle sürecin tasarımcıdan bağımsız olarak otomasyonu ve tümüyle bilgisayar ortamında kontrolü üzerinde çalışılması planlanmaktadır. *Kullanılan sanal benzetim yönteminin iyileştirilmesi:* Robot davranışının bilgisayarda benzetiminin daha verimli hale getirilmesi için ayrıntılı çalışmalar yapılması gerekmektedir. *Kavramsal robot tasarımı için davranış değerlendirme yapılabilmesine olanak sağlayacak tekniklerin geliştirilmesi:* Tasarlanacak robot için farklı davranış alternatiflerinin değerlendirilmesi gerekir. Böyle bir değerlendirmede kullanılacak değerlendirme ölçütlerinin geliştirilmesi ve sunulan sistematik yaklaşım temelinde uygulanması üzerinde araştırma yapılması gereken bir alandır. *Geliştirilen yöntemin yaratıcılık, kavramsal tasarımın kısmen otomasyonu ve/veya zaman ve maliyet avantajlarını gösterebilecek karmaşık tasarım örnekleri üzerinde karşılaştırmalı uygulaması:* Önerilen yöntemin gerçekçi koşulları ve kısıtları olan bir (veya mümkünse daha fazla) karmaşık tasarım örneği/örnekleri için uygulanarak, bu yöntem kullanıldığı ve kullanılmadığı durumlarda ortaya çıkan kavramsal tasarım ve masa üzeri tasarımların karşılaştırmalı değerlendirmesinin yapılması gerekmektedir.

Teşekkür

Bu makalede sunulan çalışma Atılım Üniversitesi tarafından, BAP (Bilimsel Araştırma Programı) kapsamında desteklenmiştir (Proje No: ATÜ-BAP-1011-07). Z.E. fikir sahibi olup davranış tabanlı modelleme yaklaşımını geliştirmiş, M.A. örnek uygulama için modelleme, kodlama ve benzetimi yapmıştır.

Kaynaklar

- [1] Cao DX, Zhu NH, Cui CX, Tan RH. An agent-based framework for guiding conceptual design of mechanical products. *Int J Prod Res* 2008; 46(9): 2381-2396.
- [2] Cao DX, Fu, WM. A knowledge-based prototype system to support product conceptual design. *Computer-Aided Design and Applications* 2011; 8(1): 129-147.
- [3] Ziegler BP. DEVS representation of dynamic systems: event-based intelligent control. *P IEEE* 1989; 77(1): 72-80.
- [4] Peterson JL. Petri Nets. *ACM Comput Surv* 1977; 9(3): 223-252.
- [5] Reisig W. Petri Nets-An Introduction. Berlin: Springer-Verlag, 1985.

- [6] Murata T. Petri Nets: properties, analysis and applications, *P IEEE* 1989; 77(4): 541-580.
- [7] Reisig, W. A Primer in Petri Net Design. Berlin: Springer-Verlag, 1992.
- [8] Pahl G, Beitz W, Feldhusen J, Grote KH. Engineering Design-A Systematic Approach (3rd Ed.). Londra: Springer-Verlag, 2008.
- [9] Erden MS, Komoto H, Van Beek TJ, D'Amelio V, Echavarría E, Tomiyama T. A review of function modelling-approaches and applications. *AI EDAM* 2008; 22(2): 147-169.
- [10] Deng Y, Ma Y, Shi L. A behavioural process design model for development of assembly devices. *International Journal of Product Development* 2013; 18(5): 445-46.
- [11] Elmqvist H, Mattsson SE, Otter M. Modelica - a language for physical system modelling, visualization and interaction. *IEEE Symposium on Computer-Aided Control System Design*; 22-27 Ağustos 1999; Hawaii-ABD. 630-639
- [12] Bracewell RH, Sharpe JEE. Functional descriptions used in computer support for qualitative scheme generation-Schemebuilder. *AI EDAM* 1996; 10(4): 333-346.
- [13] Fritzon P, Gunnarsson J, Jirstrand M. MathModelica-an extensible modelling and simulation environment with integrated graphics and literate programming. The 2nd International Modelica Conference; 18-19 Mart 2002; Oberpfaffenhofen-Almanya. 41-54.
- [14] Fan Z, Seo K, Hu J, Goodman ED, Rosenberg RC. A novel evolutionary engineering design approach for mixed-domain systems. *Engineering Optimization* 2004; 36(2): 127-147.
- [15] Fan Z, Wang J, Goodman ED. Exploring open-ended design space of mechatronic systems, *International Journal of Advanced Robotic Systems* 2004; 1(4): 295 – 302.
- [16] Behbahani S, de Silva CW. Mechatronic design evolution using bond graphs and hybrid genetic algorithm with genetic programming. *IEEE/ASME Transactions on Mechatronics* 2013; 18(1): 190-199.
- [17] Secchi C, Bonfe' M, Fantuzzi C. On the use of UML for modelling mechatronic systems, *IEEE T Autom Sci Eng* 2007; 4(1): 105–113.
- [18] Chen R, Liu Y, Cao Y, Zhao J, Yuan L, Fan H. ArchME: A Systems Modeling Language extension for mechatronic system architecture modeling. *AI EDAM* 2018; 32(1): 75-91.
- [19] Sell R, Tamre M, Lehtla M, Rosin A. A conceptual design method for the general electric vehicle, *Estonian Journal of Engineering* 2008; 14(1): 3-16.
- [20] van der Vegte WF, Horvath I. Consideration and modelling of use processes in computer-aided conceptual design: a state of the art review. *Transactions of the SPDS-Journal of Integrated Design and Process Science* 2002; 6 (2): 25-59.
- [21] van der Vegte WF. A survey of artefact simulation approaches from the perspective of application to use processes of consumer durables. The 6th International Symposium on Tools and Methods for Competitive Engineering; 18-22 Nisan 2006; Ljubljana-Slovenya. 617-632.
- [22] Liu C, Hildre HP, Zhang H, Rølvåg T. Conceptual design of multi-modal products. *Res Eng Des* 2015; 26(3): 219-234.
- [23] Shi L, Deng YM. Phenyl-Loop Behaviors of Mechatronic Products and their Usefulnesses for Conceptual Design. The 2nd International Conference on Mechanical and Electronics Engineering; 1-3 Ağustos 2010; Kyoto-Japonya. 1346-1350.
- [24] Tor SB, Lee SG, Britton GA, Zhang WY. Knowledge-Based Functional Design of Industrial Robots, *Int J Prod Res* 2008; 46(16): 4501-4519.
- [25] Gholami A, Sadeghi BB. Soccer Goalkeeper Task Modeling and Analysis by Petri Nets. *Journal of Computer and Robotics* 2018; 11(1): 77-85.
- [26] Erden Z, Erden A, Erkmen AM. A Petri Net approach to behavioural simulation of design artefacts with application to mechatronic design. *Res Eng Des* 2003; 14(1): 34-46.
- [27] Erden Z. Representation of the operational behaviour of an educational robot at conceptual design using Petri Nets. *ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis*; 12-14 Temmuz 2010; İstanbul-Türkiye. 855-862.
- [28] Erden Z. State-based conceptual design in mechatronics via Petri Nets. *Control Eng Appl Inf* 2011; 13(2): 70-75.
- [29] Araz M, Erden Z. Behavioural Representation and Simulation of Design Concepts for Systematic Conceptual Design of Mechatronic Systems Using Petri Nets. *Int J Prod Res* 2014; 52(2): 563-583.
- [30] Konez-Eroğlu A, Erden Z, Erden A. Biological System Analysis in Bioinspired Conceptual Design (BICD) for Bioinspired Robots. *Control Eng Appl Inf* 2011; 13(2), 81-86.
- [31] Erden Z, Konez-Eroğlu A, Erden A. Kavramsal Mekatronik Tasarımda Davranış Tabanlı Modelleme ve Biyobenzetim Robot Tasarımına Uygulanması. *Mühendis ve Makina* 2011; 52(618):50-59.
- [32] RSoft. Artifex™ 4.4 Getting Started Handbook 2004. RSoft Design Group, ABD.