State-Based Conceptual Design in Mechatronics via Petri Nets: A Case Study for an Educational Robot

Zuhal Erden*

**Mechatronics Engineering Department, Atilim University, 06836 Incek, Golbasi, Ankara, Turkey (Tel:90-312-5868302; e-mail: zuhal@ atilim.edu.tr)*

Abstract: Use of state-based models to represent artifact behaviors at conceptual design is a challenging area for design research. Conceptual design of mechatronic systems needs a multi-domain approach in which the “logical behavior” of a mechatronic design artifact is described without any physical realization. This paper presents a case study on state-based representation for the intended behavior of a non-existent robot at early conceptual level. The behavior is defined through a demonstrative scenario and represented as states and state transitions independent of any physical embodiment. Discrete Event System Specification (DEVS) and Petri Net formalism are used for the model. This representational model is first step towards the development of a virtual prototype for the logical behavior of robot design.

Keywords: State-based design, Petri Net, conceptual design, mechatronics, educational robot.

1. INTRODUCTION

A significant problem encountered in conceptual design is to develop and simulate the intended behaviour of a design artefact to perform an overall function without developing its physical model. To overcome this problem, designers need virtual prototypes based on behavioural models that reduce the design time at conceptual design stage. These models use operational-logical descriptions, which are based on the decomposition of a process into sub-processes (Frederick van der Vegte, 2006). From engineering design perspective, operational-logical descriptions are considered to be based on the decomposition of an overall function into sub-functions and these descriptions are used to develop a model for the corresponding behaviour of a design artefact at conceptual level.

Mechatronic systems require synergistic integration of mechanics, electronics, information technology and control engineering in a design artefact starting from the conceptual design phase. This synergy can be accomplished by treating all elements forming the final product as equally important during the entire design process, irrespective of their physical nature (Mrozek, 2003). This approach guides the conceptual design of mechatronic systems such that, firstly solution concepts are selected and elaborated, and then interconnection of these concepts into an appropriate system is realized. Selecting physical implementation for each concept is considered after the through validation of the conceptual design. The importance of manipulating abstract solution concepts during mechatronic system design results in considerable effort for modelling the behaviour of a mechatronic design artefact at this level. Mechatronic system behaviour exhibit discrete and continuous characteristics and these systems are treated as hybrid systems which can be modelled using state-based representations. In a state-based representation, modelling of functional interactions in a mechatronic system is investigated in terms of changes in the states of the system. Generally, discrete behaviour is modelled by state-based modelling tools whereas continuous behaviour is modelled via differential equations. Due to the difficulty of using multiple formalisms, deriving an abstract model for the operative part is required for conceptual design (Moncelet, 1998).

As a case study for state-based conceptual design, the main objective of the present study is to model logical behaviour of a non-existent rabbit-like robot at the conceptual design level. The behaviour is based on a predefined scenario that describes the actions to be performed, when certain environmental effects occur. The behaviour is modelled as a discrete event system behaviour which is implemented using Petri Net formalism (Peterson, 1977; Murata, 1989). The actions are considered as states of the robot, while environmental effects are treated as events. This study is an initiation for developing a virtual prototype of the behaviour for an educational robot and for comparison with the actual behaviour of its future physical prototype.

The paper is organized as follows. Section 2 presents a brief overview of behavioural modelling approaches for state-based conceptual design. Section 3 introduces the state-based conceptual design approach for a mechatronic system. In Section 4, state-based representation for the conceptual design of a rabbit-like robot is described. Finally, in Section 5 conclusions and future work are stated.

2. LITERATURE REVIEW

Being a multi-domain problem, conceptual design of mechatronic systems needs a special design philosophy which is different than conventional single-domain engineering design problems simply because of the need for integrating several types of energy behaviours in a physically integrated system. Multi-domain design is difficult because such systems tend to be very complex and most current simulation tools operate over a single domain (Goodman et al., 2002). Multi-domain design philosophy supports the necessity for an intensive interaction and integration between different engineering disciplines in order to develop efficient, compact, precisely-controlled, task repeatable, reliable, re-programmable and flexible (multi-purpose) products. The key property of mechatronics is the integration of the mechanical, electronic, software and control engineering fields starting from the early design stages, particularly at the conceptual design stage. Even an identification of the design need requires mechatronic approaches in novel designs. The mechatronic concepts have high diffusion through the design stages from the identification of the need down to the physical production. This diffusion is achieved through functional and behavioural synergy, which can be accomplished through the development of concept variants on a functional basis and formal representation of their behaviour during the conceptual design stage, regardless of any physical structuring.

Two important approaches towards designing interdisciplinary mechatronic systems are described in (Mrozek, 2003). One of them is the visual modelling with UML (Unified Modelling Language) which has been developed as a language for the modelling of information systems. It can be used to describe all elements of mechatronic systems on different levels of abstraction. The second approach is the use of Modelica (Elmqvist et al., 1999; Fritzson, 2006) which is an object-oriented language for physical modelling of complex systems. State transition diagrams are used to develop a modelling framework to support conceptual design of multiple interaction-state mechatronic devices (Xu et al., 2005). In such devices, interactions between elements of use (environment) and elements of the device can have different qualitative structures (different interaction topologies).

In order to contribute the modelling and simulation of multi-disciplinary designs at the early conceptual level, a research work was initiated to develop a Petri Net based model for simulating the behaviour of concept variants. As the first step of the research, the PNDN-**P**etri **N**et Based **D**esign **N**etwork was developed (Erden et al., 2003). PNDN models the information flow through the functions of a design artefact using a Petri Net-based formalism leading to the representation of the artefact's *logical behaviour*. PNDN is intended to represent and analyse artefact behaviours by using information flow and logical relations that result in these behaviours. PNDN utilizes information flow to compare and reveal flaws and drawbacks of the available design alternatives. PNDN models logical behaviour of any design artefact on a functional basis independent of any physical realization. This makes the PNDN applicable to especially multi-disciplinary design philosophies such as mechatronic design.

Information flow (logical behaviour) should be integrated with material and energy flows in a unified model for complete simulation of the behaviour of design artefacts at the conceptual design phase. Modelling of the energy and material flow is treated as modelling the "operational behaviour" of the design artefact. Combining logical behaviour and operational behaviour of an artefact in a unique modelling framework forms a bridge from symbolic reasoning in conceptual design to a formal one in the embodiment and detailed design phases. In addition, modelling and integration of energy, material and information flows allows the designer to analyse the interactions between various functions of the system under conceptual design and to determine any deadlocks related with the three flow phenomena for the operation of the system. Thus, necessary changes can be done during an early design phase to prevent any deadlock. Further research is initiated such that the ultimate objective is to develop a formal methodology for the modelling of material, energy and information flows in a single design network model using the Petri Net formalism. This paper presents a case study towards this ultimate objective such that a Petri Net model is developed to represent the behaviour of a non-existent educational robot, at high level of abstraction during conceptual design.

Representation of the behaviour of a non-existent design artefact requires formal modelling for qualitative simulation. An educational robot is considered as a hybrid system with continuous and discrete behaviours. A widely used formalism for modelling such systems is DEVS (Discrete Event System Specification) formalism (Ziegler, 1989). It is used to create system models in which the discrete-event behaviour is modelled using a Finite State Machine (FSM) called DEVS diagram and the continuous behaviour is modelled algebraically using dedicated differential equations (Frederick van der Vegte, 2006). Since the present study concerns an artefact's behaviour at the conceptual design level, the model is developed only for the discrete-event behaviour, reserving the modelling of continuous behaviour to further studies. The discrete-event behaviour is represented by using DEVS formalism together with Petri Nets. This type of qualitative modelling allows designers to simulate the intended behaviour of a non-existent artefact using the mathematical constructs of Petri Nets such as reachability and liveness (deadlock free operation) at conceptual design level. Since the objective of this research is to represent the behaviour of an educational robot, simulation is also left to a further study.

3. STATE-BASED CONCEPTUAL DESIGN APPROACH FOR MECHATRONIC SYSTEMS

The presented approach is based on discrete event system modeling in which the behavior of a mechatronic system is represented as a sequence of events. Each event occurs at an instant in time and marks a change of state in the system. Events may possibly have a continuous evolution once they start, but the primary focus is on the beginning and the end of such events, since ends can cause new beginnings. DEVS defines system behavior as well as system structure at an abstract level. System behavior in DEVS formalism is described using input and output events together with states.

State-based modeling approach presented in this paper starts with an informal description for the behavior of a mechatronic system to accomplish an overall task. Then it is converted into a formal DEVS representation. The operation of a mechatronic system at the highest level of abstraction is composed of three states as "PERCEPTION", "COGNITION" and "MOTORIC ACTION" (Fig 1). The system communicates its environment to collect and process data during "PERCEPTION" state. In "COGNITION" state, processed data is used with proper reasoning and decision making to respond predictable/unpredictable changes in the environment. In this representation, it is assumed that all or some of the data may be converted into information in the perception or cognition stages. "MOTORIC ACTION" is the state in which physical task execution is performed in accordance with decision making and/or as a reflexive response to changes in environment. "PERCEPTION" is decided as the initial state; because once the system starts its operation, it is expected to start collecting data from the environment for processing and decision making to create a motoric action. The environment outside a mechatronic system is defined as the physical medium which includes the physical world, and other mechatronic/non-mechatronic systems.

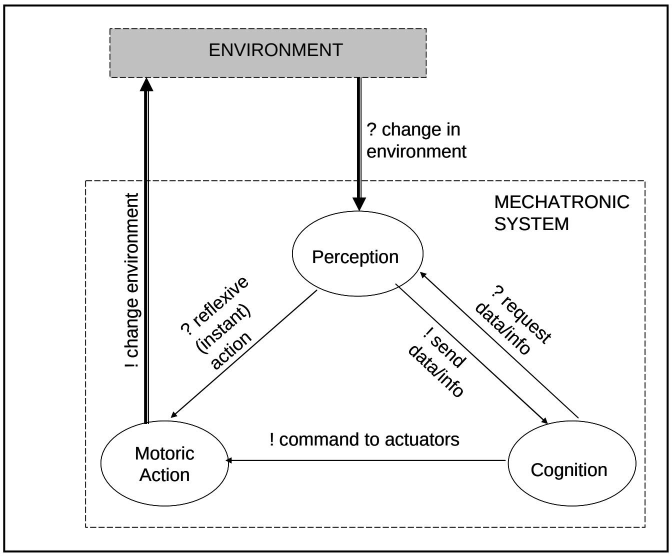

Fig. 1. State-based representation of a mechatronic system.

Four types of state changes are defined for mechatronic system behavior. When the system is in "PERCEPTION" state, the state may change either to "COGNITION" by sending data (an output event for "PERCEPTION") for decision making or to "MOTORIC ACTION" as a result of a reflexive input from the environment (an input event for "PERCEPTION") which may not require any cognitive process. "COGNITON" may change into "PERCEPTION" state, when there is a request for data/info (an input event for "COGNITION") to make decisions about motoric actions. Obviously, there is a change from "COGNITION" to "MOTORIC ACTION" by producing commands to actuators (an output event for "COGNITION"). These input and output events are explained in Table 1.

Table 1. Event descriptions for state transitions of a mechatronic system

Event Name	Description	Property
? request data/info	Cognition requires data/info for decision making.	Input event
? reflexive action	An unexpected occurrence of an event is recognized.	Input event
! send data/info	Processed data/info is sent to cognition for reasoning and decision making	Output event
! command_to_actuators	A command is sent to actuators for physical task execution	Output event

DEVS representation for abstract mechatronic system behaviour is refined by a case study which is explained in the following section.

4. STATE-BASED DESIGN OF A RABBIT-LIKE ROBOT

Use of DEVS formalism for the representation of a mechatronic system is applied in a case study which aims at developing the behavioural model for the conceptual design of a novel rabbit-like robot for undergraduate mechatronics design education practice (Erden, 2010). The educational rabbit-like robot has been developed within the context of a senior level course sequence "MECE 401/402 Mechatronics Design I/II" in the Mechatronics Engineering Department of ATILIM University (MECE 401-402 Course Website). Model structuring is based on description of a scenario for the robot's intended behaviour. This behaviour is then represented as a DEVS model. In the DEVS model, the robot's behaviour is considered as discrete-event system behaviour composed of a set of states and state transitions are resulted from event occurrences.

4.1 Operational Scenario for the Rabbit-Like Robot

An operational scenario for the rabbit-like robot is described as shown in Fig 2. The robot is required to exhibit the following behaviours as "frightened", "sleepy", "shock", "happy" and "pain". After starting its operation, when heat is perceived the "pain" behaviour is expected. With the "pain" behaviour ears are straight, eyes are straight open, body is in reflex motion and the robot is screaming. Then a loud noise is perceived by the robot, so it shows "frightened" behaviour in which ears move back, eyes are wide open, and the whole body is shaking. After that, a human strokes the rabbit robot; it shows "happy" behaviour. In happy behaviour, ears are shaking, eyes are flashing, body is in happy motion, tail is shaking and happy sound is emitted. When the robot perceives the wall in front, it starts to turn left and continue walking. When intensive light occurs, the rabbit robot goes into shock and does not move until the light is removed.

After a walk of 10 seconds period, the rabbit shows sleepy behaviour at the end of the path. In sleepy behaviour, ears move back, eyes are closed and body is in lying position. In the present state of the research, conflicting information and commands are ignored for the simplicity.

4.2 States, Events and State Transitions

The operational scenario described in the previous section leads to the definition of some states and events for possible state transitions. The states of the rabbit-like robot's behaviour are defined as follows:

IDLE: The rabbit robot is OFF mode, so the eyes are closed. It is silent, the body has no motion, tail is down and ears are straight.
WALKING: The robot is ON mode and it is walking. The eyes are opened, tail is down and ears are straight. It is silent.
PAIN BEHAVIOUR: The robot is screaming. Body is in reflex motion, eyes are opened, tail is down and ears are straight.
FRIGHTENED: It is silent. Body is shaking, eyes are opened, tail is down and ears are on the back side.
HAPPY BEHAVIOUR: The rabbit robot emits "happy sound". Body is in happy motion, eyes are flashing, tail and ears are shaking.
TURN LEFT: Happy behaviour continues, only the body motion is changed. Robot is turning left.
WALKING HAPPY: Happy behaviour continues; only the body motion is changed. Robot is walking straight.
SHOCK BEHAVIOUR: The robot is silent and it has no body motion. Eyes are opened, ears and tail are freezing.
WALKING-10: The shock behaviour is continued, but body motion changes, it is walking.
SLEEPY BEHAVIOUR: The robot is lying and it is silent. Tail is in current position, eyes are closed and ears are on back side.

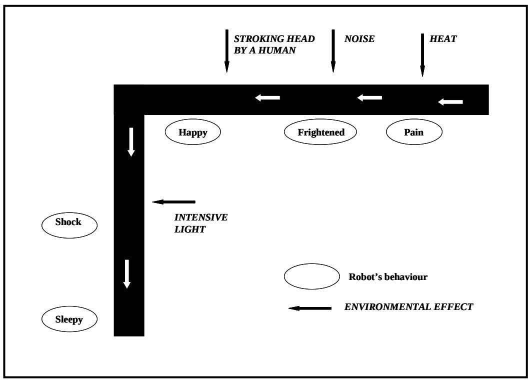

Fig. 2. Operational scenario for the rabbit robot.

Transitions between the above listed states may occur as a result of the occurrences of input and output events which are the environmental effects described in the scenario and they are given in Fig 3 as a graphical DEVS model. DEVS model is used to develop a Petri Net representation for robot's behaviour.

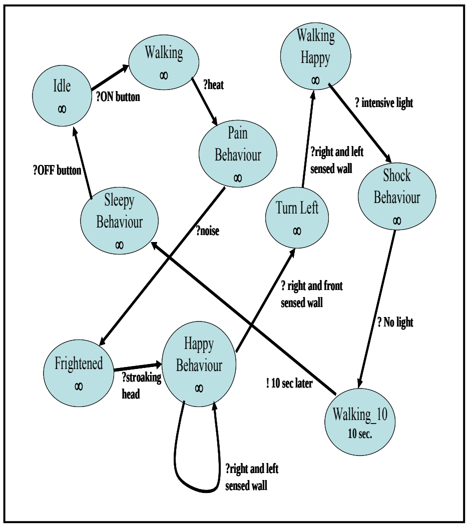

Fig. 3. DEVS model for the rabbit robot's behaviour.

4.3 Petri Net Model for the Robot's Behavior

A Petri Net is one of the several mathematical modelling languages for the description of discrete event systems (Peterson, 1977; Murata, 1989). A Petri net is a directed bipartite graph, which consists of two types of nodes as *transitions* and *places*.

Transitions (represented by rectangles) describe discrete events that may occur in the system modelled by a Petri Net and places (represented by circles) describe conditions in the system. *Directed arcs* run from a place to a transition or vice versa, and they are represented by arrows. Directed arcs describe which places are pre- and/or post-conditions for different transitions. The places from which an arc runs to a transition are called the *input places* of the transition; the places to which arcs run from a transition are called the *output places* of the transition. A *marking* of a Petri Net represents a *state* of the system and it is a mapping of the places on the set of non-negative integers. Graphically a marking is represented by a black dot (•) called *token* that can be deposited in the places of a Petri Net and identifies the occurrence of signal or state characterized by the place. The presence of a token in a place is interpreted as holding the truth of the condition associated with the place.

A Petri Net is mathematically represented by a 5-tuple (P, T, F, W, M_0) (Peterson, 1977) where,

- $P = \{p_1, p_2, p_3, \ldots\ldots, p_n\}$ is a finite set of *Places*,
- $T = \{t_1, t_2, t_3, \ldots\ldots, t_m\}$ is a finite set of *Transitions* such that; $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$,
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs from Places to Transitions or from Transitions to Places
- $W: F \rightarrow \{1,2,3, \ldots\ldots\}$ is a weight function,
- $M_0: P \rightarrow \{0,1,2,3, \ldots\ldots\}$ is the initial marking

In order to model the dynamic behaviour of a system, a state or marking in a Petri Net is changed according to the following *transition firing rule* (Peterson, 1977; Murata, 1989).

The Petri Net model development for the behaviour of the rabbit-like robot was initiated using its DEVS model. Places and transitions are defined based on the states and events presented in Section IV.B. The PN model of the rabbit robot is called as the PN_Rabot which is formally represented as a 5-tuple (P, T, F, W, M_0) where,

$P = \{p_i \mid i = 1,2,3, \ldots, 23\}$

$T = \{t_j \mid j = 1,2,3, \ldots, 9\}$

All of the arc weights in the model are defined as 1. Places and transitions in the PN_Rabot are listed in Table 2. The initial state of the robot is taken as the IDLE state in which the robot is in OFF mode, so the eyes are closed. It is silent, the body has no motion, tail is down and ears are straight. The initial state is represented by an initial marking M_0 as follows;

$$M_0 = [11000100000001000010100]$$

The net structure with the set of directed arcs and the initial marking is represented in Fig 4. When the system is in the initial state represented by M_0, T8 ("push ON button" transition) is enabled and it fires resulting in another marking M_1 as follows:

$$M_1 = [01000100000001000101000]$$

This new marking represents a new state in which the robot is in ON mode and it is walking. The eyes are opened, tail is down and ears are straight. It is still silent. Marking M_1 represents WALKING state (Fig 5). Then, heat is perceived by the robot whose state changes from WALKING to PAIN BEHAVIOUR. In the PN_Rabot, this state transition is modelled as follows: In marking M_1, "heat perceived" transition is enabled and when it fires marking M_2 is obtained such that the robot is screaming, body is in reflex motion, eyes are opened, tail is down and ears are straight.

$$M_2 = [01000100010000101001000]$$

Table 2. Places and transitions in PN_Rabot

Places	Transitions
P1: eyes closed	T1: 10sec finished
P2: ears straight	T2: intensive light
P3: ears back	T3: push OFF button
P4: ears freezing	T4: head stroked
P5: ears shaking	T5: noise received
P6: tail down	T6: perceiving wall
P7: tail shaking	T7: no wall in front
P8: tail freezing	T8: push ON button
P9: body shaking	T9: no intensive light
P10: eyes open	T10: heat perceived
P11: eyes flashing	
P12: happy sound	
P13: happy motion (body)	
P14: silent	
P15: scream	
P16: turning left	
P17: body reflex	
P18: walking	
P19: body-no-motion	
P20: ON	
P21: OFF	
P22: body freeze	
P23: lying	

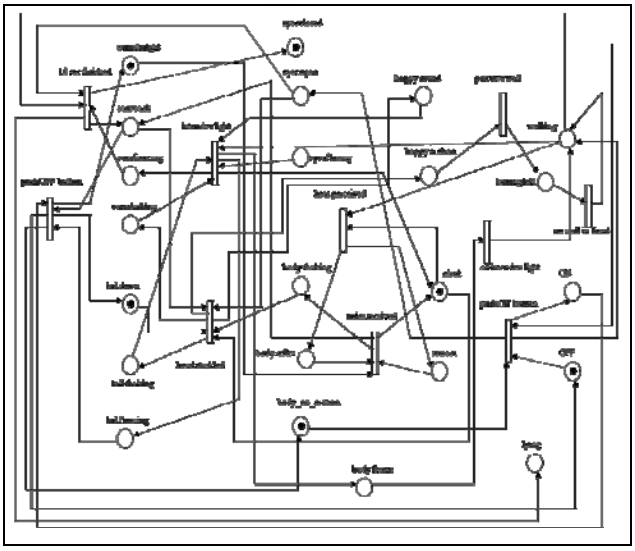

Fig. 4. Petri Net model for the operational behaviour of the rabbit robot with the initial marking

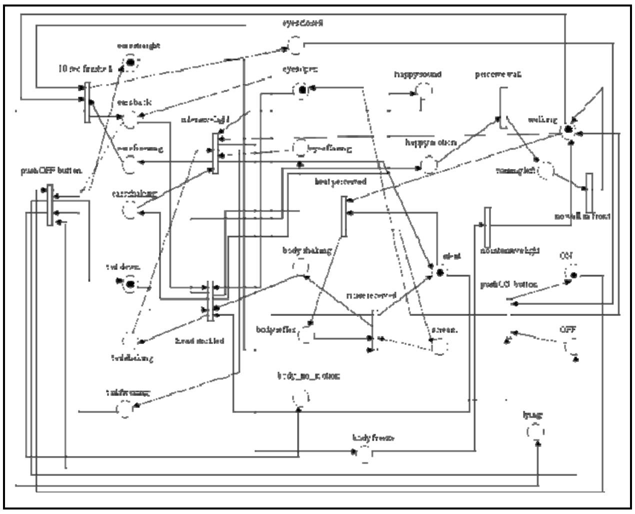

Fig. 5. PN_Rabot with marking M_1.

The remaining state transitions are represented by the following reachability tree for the PN_Rabot. The robot returns back to its IDLE state as it is expected.

M_0 = [11000100000001000010100] IDLE
M_1 = [01000100000001000101000] WALKING
M_2 = [01000100010000101001000] PAIN BEHAVIOUR
M_3 = [00100100110001000001000] FRIGHTENED
M_4 = [00001010011110000001000] HAPPY BEHAVIOUR
M_5 = [00001010011100010001000] TURN LEFT
M_6 = [00001010011100000101000] WALKING HAPPY
M_7 = [00010001000001000001010] SHOCK BEHAVIOUR
M_8 = [00010001000001000101000] WALKING_10
M_9 = [10100001000001000001001] SLEEPY BEHAVIOUR
M_{10} = [11000100000001000010100] IDLE

5. CONCLUSIONS

An important challenge in mechatronic design research is to model the behaviour of mechatronic systems at conceptual design using precise specifications and modelling languages. Mechatronic systems span over multi-domains rather than being tailored towards a particular domain. A uniform modelling formalism is required for the discrete and continuous behaviours of mechatronics systems across all disciplines at every design level. This paper presents the initial part of a research on developing a modelling framework for the behaviour of a mechatronic system at an abstract level in conceptual design independent of any physical architecture. The state-based representation for the behaviour of a non-existent educational rabbit-like robot is described in this study. The behaviour of the robot is defined as an operational scenario that is formally represented as a combination of states and state transitions using Petri Net formalism. The study in this paper reveals that once the behaviour is verbally described, it can be symbolized and a Petri Net model is used to model the discrete dynamic behaviour at an abstract level during conceptual design independent of any physical realization. Future work on the presented study is aimed at simulating the Petri Net model and evaluating its performance.

ACKNOWLEDGEMENTS

This research project is supported by TUBITAK-The Scientific and Technological Research Council of Turkey (Project No: 109M379 "Development of biomimetic design methodology with reverse engineering in cognitive recognition and control of biomimetic robots" as a joint project between the University of Craiova, Romania and Atılım University, Turkey).

REFERENCES

Elmqvist, H., Mattsson, S. E. and Otter, M. (1999). Modelica-A Language for Physical System Modelling, Visualization and Interaction, In *IEEE Symposium on Computer-Aided Control System Design (CACSD '99)*, Hawaii, USA.

Erden, Z. (2010). Representation of the Operational Behaviour of an Educational Robot at Conceptual Design Using Petri Nets, In *Proc. ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis (ESDA 2010).* 855-862. İstanbul, Turkey.

Erden, Z., Erden, A. and Erkmen, A. M. (2003). A Petri Net Approach to Behavioural Simulation of Design Artefacts with Application to Mechatronic Design. *Research in Engineering Design*, 14(1), 34-46, 2003.

Frederick van der Vegte, W. (2006). A Survey of Artefact Simulation Approaches from the Perspective of Application to Use Processes of Consumer Durables. In *Proceedings of TMCE 2006*, 617-632.

Fritzson, P, (2006), "Introduction to Object-Oriented Modelling and Simulation with OpenModelica - A Tutorial", Available at http://www. modelica.org/ (Accessed: April 22, 2010)

Goodman, E.D., Seo, K., Rosenberg, R. C., Fan, Z., Hu, J. and Zhang, B. (2002). Automated Design Methodology for Mechatronic Systems Using Bond Graphs and Genetic Programming. In *Proc. 2002 NSF Design, Service and Manufacturing Grantees and Research Conference.* 206-221, San Juan, Puerto Rico.

MECE 401-402 Course website. [Online]. Available: http://mechatronics.atilim.edu.tr/courses/mece401/

Moncelet, G., Christensen, S., Demmou, H., Paludetto, M. and Porras, J. (1998). Analysing a Mechatronic System with Colored Petri Nets. *International Journal on Software Tools for Technology Transfer.* 2(2), 160-167.

Mrozek, Z. (2003). Computer Aided Design of Mechatronic Systems. *Int. J. Appl.Math. Comput. Sci.* 13(2), 255-267.

Murata, T. (1989). Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4), 541-580.

Peterson, J. L. (1977). Petri Nets, *ACM Computing Surveys*, 9, 223-252.

Xu, C., Gupta, S.K., Yao, Z., Gruninger, M. and Sriram, R. (2005). Towards Computer-Aided Conceptual Design of Mechatronic Devices with Multiple Interaction States. In *Proc. ASME Design Engineering Technical Conference and Computers and Information in Engineering Conference (DETC 2005)*, Long Beach, California, USA.