A Novel Methodology for Development of Distributed Industrial Wireless Sensor and Actuator Network in Reconfigurable Mechatronic Devices

Reza Abrishambaf

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Doctor of Philosophy
in
Electrical and Electronic Engineering

Eastern Mediterranean University
September 2012
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Doctor of Philosophy in Electrical and Electronic Engineering.

Assoc. Prof. Dr. Aykut Hocanın
Chair, Department of Electrical and Electronic Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Doctor of Philosophy in Electrical and Electronic Engineering.

Prof. Dr. Majid Hashemipour
Supervisor

Examining Committee

1. Prof. Dr. Majid Hashemipour
2. Prof. Dr. Mustafa Okyay Kaynak
3. Prof. Dr. Şener Uysal
4. Assoc. Prof. Dr. Mustafa K. Uyguroğlu
5. Assoc. Prof. Dr. Hasan Demirel

ABSTRACT

Recently the globalization of manufacturing industry systems has led to increase the competition in respond to today's demanding market especially in medium-size companies. The global competition requires the manufacturing systems to be flexible and reconfigurable specifically in the shop floor level where mechatronic devices reside. In this thesis, a novel methodology for the development and structural modeling of industrial wireless sensors and actuators is presented in order to provide flexibility and reconfigurability to the mechatronic devices. The proposed methodology is based on the implementation of the IEC 61499 standard for the distributed control of mechatronic systems. The methodology also addresses the existing problems of this standard for capturing the system requirements in the development process. For this reason, the Unified Modeling Language (UML) is used in order to overcome this problem. Petri net as a mathematical modeling language is employed in order to demonstrate the performance of the proposed methodology. A prototype software tool is designed to translate the developed UML diagrams to IEC 61499 standard models as XML files.

Keywords: Distributed Control Systems, IEC 61499 Function Blocks, Industrial Wireless Sensor and Actuator Networks

ÖZ

Son zamanlarda üretim endüstri sistemlerinde yaşanan globalleşme süreci, özellikle orta ölçekli şirketlerde olmak üzere günümüz talep piyasasının ihtiyaçlarının karşılanması doğrultusunda rekabet ortamının gelişmesi ve artmasına yol açmıştır. Küresel rekabet ortamı, özellikle mekatronik ürünlerinin bulunduğu fabrika ortamlarında, üretim sistemlerinin esnek ve yeniden yapılandırılabilir olmasını gerektirmektedir. Bu tez çalışmasında, mekatronik ürünlerinin esnekliği ve yeniden yapılandırılabilirliğinin sağlanması amacıyla endüstriyel kablosuz sensörler ve çalıştırıcıların geliştirilmesi ve yapısal olarak modellenmeleri için yeni bir yöntem geliştirilmiştir. Bahsı geçen yöntem mekatronik sistemlerinin dağınık kontrolü için kullanılmakta olan IEC 61499 standardının uygulanmasına dayanmaktadır. Bu yöntem ayrıca zikredilen standardın, geliştirilme sürecindeki sistem gereksinimlerinin belirlenmesi konusunda yaşamakta olduğu mevcut problemlerine de hitap etmektedir. Bu neden ile, bu problemin ortadan kaldırılması amacıyla Birleştirilmiş Modelleme Dili'nden (BMD) (Unifield Modeling Language- UML) yararlanılmıştır. Önerilen yöntemin performansının gösterilmesi amacıyla matematiksel bir modelleme dili olarak Petri net'den yararlanılmıştır. Geliştirilen BMD diyagramlarının XML dosyaları olarak IEC 61499 standart modellerine çevirilmesi amacıyla bir prototip yazılım aracı tasarlanmıştır.

Anahtar Kelimeler: Dağıtılmış Kontrol Sistemleri, IEC 61499 Standart, Endüstriyel Kablosuz Sensör ve Aktüatör Ağı

Dedicated to

My parents and brothers

ACKNOWLEDGMENT

First and foremost I thank my supervisor Prof. Dr. Majid Hashemipour and my previous supervisor Prof. Dr. Suha N. Bayindir for guiding and helping me in my study, for their patience and sharing kindly knowledge with me.

Besides my supervisors, I would like to thank Assist Prof. Dr. Mert Bal, Miami University, USA as an external co-supervisor, Prof. Dr. V. Vyatkin, University of Auckland, New Zealand and Prof. Dr. Runyi Yu for giving their time in the contribution of the thesis and papers.

Many thanks to my friends who enhanced my motivation by supporting me with their presence.

I also wish to thank all my friends at the department of Electrical and Electronic Engineering, and specially the chairman, Assoc. Prof. Dr. Aykut Hocanın.

Last but not least, I would like to express my appreciation to my parents and my brothers who supported me all my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

A	Arcs
A_i	Actuator i
c	Speed of light
D	System boundary
d	Distance
$d_{critical}$	Critical distance
$d_{crossover}$	Crossover distance
E_{elec}	Energy consumed to process one bit
$E_{n\ to\ B\ (MTE)}$	Energy required for the transmission of k bits from node n to the base
f_t	Transmission frequency
h_r	Height of the receiver's antenna from the ground
h_t	Height of the transmitter's antenna from the ground
I	Input
k	Number of bits
L	System loss factor
M	Tangible states
m	Step
N	Non negative integer number
n	Path loss exponent
n_s	Number of sensors
n_a	Number of actuators
O	Output
P	Place

S_j	Sensor j
S_{A_i}	Effected sensors by A_i actuator
T	Total time
t	Time
x	State of System
w	Weight Function
ϵ_{amp}	Energy consumed for amplifying station
λ	Wavelength
π_i	Steady State Probabilities
ADC	Analogue to Digital Converter
API	Application Programming Interface
ASRS	Automated Storage and Retrieval System
BSN	Body Sensor Network
CASE	Computer Aided Software Engineering
CC	Centralized Control
CIFB	Communication Interface Function Block
CNC	Computer Numerical Control
CPU	Central Processing Unit
DAC	Digital to Analogue Converter
DC	Distributed Control
DCS	Distributed Control Systems
DES	Discrete Event Systems
DTD	Document Type Definitions
ECC	Execution Control Chart
ESS	Engineering Support System

FB	Function Block
FBDK	Function Block Development Kit
FBRT	Function Block Run-Time
FIPA	Federation for Intelligent Physical Agent
FMS	Flexible Manufacturing System
HMI	Human Machine Interface
IEC	International Electrotechnical Commission
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineering
IPMCS	Industrial-Process Measurement and Control Systems
IPT	Industrial Process Terminator
IWSN	Industrial Wireless Sensor Network
IWSAN	Industrial Wireless Sensor and Actuator Network
JADE	Java Agent Development Framework
JRE	Java Run-time Environment
LAN	Local Area Network
LEACH	Low Energy Adaptive Clustering Head
MTE	Minimum Transmission Energy
MAS	Multi-Agent System
NesC	Network Embedded Systems C
OO	Object-Oriented
OS	Operating System
PFIB	Process Interface Function Blocks
PLC	Programmable Logic Controller
SIFB	Service Interface Function Blocks

SPN	Stochastic Petri Net
TinyOS	Tiny Operating System
UML	Unified Modeling Language
XMI	XML Metadata Interchange
XML	eXtensible Markup Language
WSN	Wireless Sensor Network

Chapter 1

INTRODUCTION

1.1 Overview

The Wireless Sensor Network (WSN) is an emerging technology which is newly adopted in a wide range of distributed monitoring and control applications such as: intelligent buildings, environmental monitoring, defense, security systems and industrial automation [1]. The WSN technology is gaining popularity since it is a combined field of mobile computing, wireless communication and microelectronics. WSN is a network of nodes whereby various data can be collected by a single node such as humidity, temperature, pressure and flow. Each sensor node in the network is capable of gathering data from its surrounding environment and processing the collected data locally. The local processing gives an ability to the nodes to perform local decision making independent from the constraints of the overall system. Therefore, each sensor node in a network can be an autonomous and intelligent unit, making decisions with an on-board microcontroller. In a WSN, the overall network of the wireless sensors may also be considered intelligent as the collaborative processing among the sensor nodes provide several features to the system such as: self-organizing, self healing and auto-routing etc. The nodes in a WSN are also capable of communicating together in order to fulfill the overall system requirements. The collaboration makes the nodes more intelligent in terms of discovering the possible faults in the network.

Due to its distributed nature, the WSN has a high potential to be the enabling technology for a variety of industrial monitoring and control applications such as: industrial process monitoring, fault detection, real-time data collection and location tracking. Recent technological developments have proven the potential of merging actuators into the Industrial Wireless Sensor Networks [6-15]. That is, not only the sensors but also actuators can be integrated into a wireless node in order to produce actions to control various operations. The addition of the actuators can make a wireless node capable to monitor and control a mechatronic device such as an industrial robot, a programmable controller or a computer numerical controlled machine. The wireless nodes can act as a gateway to the mechatronic devices in order to collaborate in the wireless domain. In the Distributed Control System (DCS) scheme, the control will take place in the collaborative network of devices, which make their own decision locally by means of the collected real-time data.

The Industrial Wireless Sensor and Actuator Networks (IWSAN) could be essential enablers of the distributed control systems as they provide flexible, wireless real-time data collection for monitoring, diagnostics, and control of the mechatronic devices at industrial automation settings. Hence, the integration of IWSAN into the function blocks framework will enhance the re-configurability and reliability of distributed control systems.

IEC 61499 is a new standard for the industrial-process measurement and control systems which has been adopted recently in distributed manufacturing. The IEC 61499 function blocks have become an industry standard in distributed control for manufacturing automation. However, there is a lack of a structural methodology that captures the user requirements of the development process in a high level of abstraction.

In this regard, IEC 61499 does not support capturing the system requirements and then translating for the design specification. This motivated the researchers to develop various tools in order to combine Unified Modeling Language (UML) and IEC 61499 function blocks. CORFU Engineering Support System (ESS) is an engineering tool compliant with IEC 61499. It captures all the requirements by means of UML and then it is translated to function blocks according to part two of IEC 61499 standard. UML-FB is developed in [68 69] for modeling and implementation of IPMCS. In [70-71] development process of industrial automation systems is addressed. The system is based on UML and IEC 61499 standard from user requirements to implementation domain. However, in all the mentioned works, the design and control architecture of industrial wireless sensor and actuator networks have not been addressed specifically on the basis of IEC 61499. In addition, several challenges exist in integrating the IWSAN into such a combined framework. The WSN technology is new and still so complicated to be implemented into the components of distributed control of manufacturing systems due to the lack of structured modeling methods and effective guidelines.

This thesis presents a methodology to address the issues of integrating IEC 61499 function blocks and the IWSAN for intelligent distributed control of mechatronic devices. UML has been proposed as a solution to overcome the shortcomings of the IEC 61499 standard for capturing the user requirements. A novel 2-dimensional methodology is proposed for modeling IWSAN by considering the requirements of the system. The methodology is structured in three phases, namely: the system requirements, design and implementation. The result of the design phase goes through a verification system in order to check the consistency and the semantic of the scheme. Each phase contains four layers which capture all the requirements from user to physical layer by means of UML diagrams. The performance evaluation of the proposed methodology is presented by Petri net which is a mathematical tool in order to design and analyze the event based systems. The Petri net graphs can be obtained by the UML diagrams developed in the methodology. A prototype tool is developed to translate the UML diagrams into their corresponding model based on IEC 61499 standard.

The major contributions of this thesis can be summarized as follows:

- Modeling sensor networks based on IEC 61499 function block standard
- Integration of industrial wireless sensor and actuator networks into mechatronic devices
- Structural modeling of industrial wireless sensor and actuator networks on the basis of IEC 61499 standard and using UML
- Performance evaluation of the proposed methodology using Petri net
- Developing a prototype software tool in order to implement the methodology

This thesis is structured as follows: in chapter 2 an overview of industrial wireless sensor and actuator networks are presented. Chapter 3 discusses the energy analysis of routing protocols in WSN for industrial applications. Chapter 4 addresses the multi-agent architectures for wireless sensor networks in distributed schemes. Chapter 5 describes the IEC 61499 Function Blocks standard and modeling wireless sensor networks on the basis of this standard. Chapter 6 presents the proposed methodology and its implementation by means of UML diagrams. In chapter 7 the performance evaluation of the proposed methodology is shown using Petri net language and finally in chapter 8, a prototype tool developed for translating the UML diagrams to their function block representation is demonstrated. Chapter 9 concludes the thesis.

Chapter 2

INDUSTRIAL WIRELESS SENSOR NETWORKS

2.1 Introduction

Recently, wireless networks have become very popular in the field of telecommunications. One of the reasons is the use of fixed network topologies in the conventional wired networks. Wireless networks are low cost and easy to install. It also provides portability to devices in the network. Wireless communication is classified based on the applications, rates of data transfer and coverage which is resulted in different IEEE standards.

Wireless Sensor Network (WSN) is an emerging technology that provoked interest in many areas such as environment monitoring, intelligent building, industrial automation and especially in academic research [1]. The network contains many spatially distributed nodes which are independent and are capable of collaborating with other nodes. Nodes can process, save and transfer the processed information rather than the raw data into a special node, so called sink (base station) node. WSNs are implemented in a large scale and they are able to organize the nodes automatically based on the application which is implemented. The collaboration among the nodes allows the network to grow and increase the area of interest from meters to kilometers [2]. In comparison to the conventional communication systems, there is no need to implement an infrastructure for WSN. Resources are limited in WSNs, that is, the size of the nodes, the memory which is used and especially the nodes are energy constraint since they are battery powered. There are differences between WSN and ad hoc networks as follows:

- WSN is a data oriented system where the data are sensitive to time may be sent to not one but many destination.
- WSNs highly depend on the application or they are application-oriented.
- The data are not transferred to a special node but to the nodes in different locations.

Latency and accuracy are the parameters which are required to be considered while designing the WSNs.

In WSNs, nodes are randomly deployed in the area of the interest in order to measure one or more physical phenomenon (Figure 2.1).

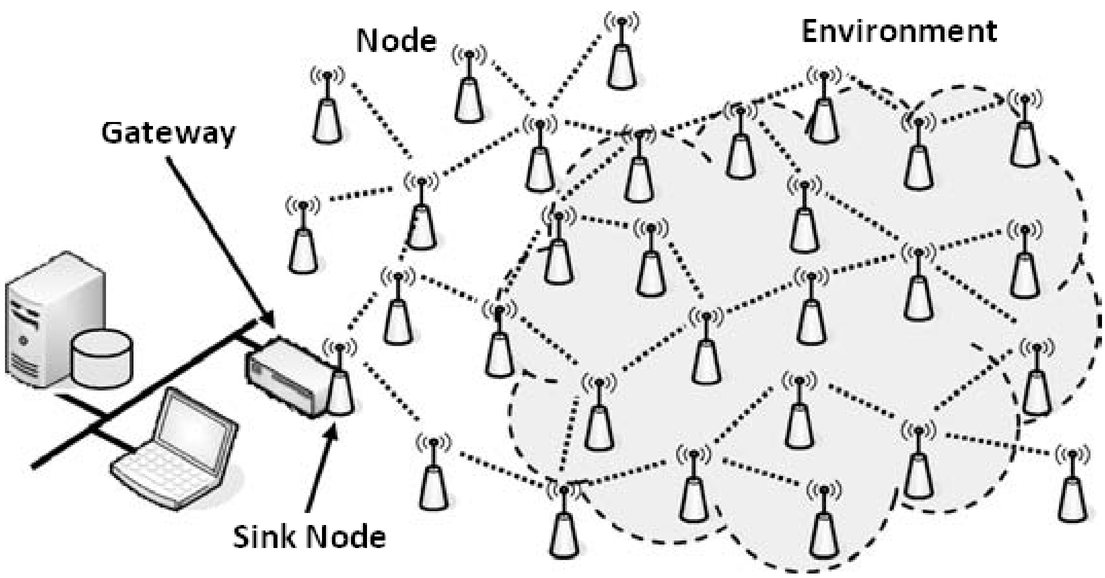

Figure 2-1: Wireless Sensor Network [1]

Each node senses the physical parameter and performs processing on the obtained raw data. Therefore, the extracted information from the data is sent to the base station instead. This is the most important property of the WSN where the data is processed locally. Sink node collects the data from the network and by means of gateway the data are forwarded to a computer or other networks such as Internet for further high-level applications.

One of the applications of WSN is the replacement of the conventional wired sensory systems. The local processing ability of the sensor nodes opens a new door to a wide range of applications. The most common physical parameters which can be measured by nodes are: temperature, light, humidity, acceleration, pressure, magnetic fields, radiation, sound, chemical parameters and geographical locations. Some examples of WSN are: Home Automation: sensor nodes are deployed in the building to measure physical parameters like temperature and humidity and based on the measurement, the actuators are controlled. Environment Monitoring: nodes are randomly deployed on a large-scale area to detect fire, earthquake and collect ambient data. Sensor nodes are used to collect data to be utilized for intelligence and targeting. Traffic Control: sensor nodes are used to monitor traffic and provide a temporary solution. Health Care: Body Sensor Networks (BSNs) are recently applied in the medical measurement systems in order to gather the vital body functions of the patients remotely. Industrial Monitoring and Control: wireless sensors are used to measure the several parameters such as pressure, mechanical stress and radiation in an industrial environment. Nodes also have the ability of not only sensing but also processing and control.

This thesis discusses the control functionality of the WSNs in the industrial systems. The functions performed locally in a distributed fashion so the need to have a central processing unit is avoided. Distributed control is much faster than the centralized approach since there is no delay caused by routing techniques [2].

There are several requirements that needed to be considered for a deployed WSN. They are highly dependent on the application and the area of interest. The requirements are as follows [1]:

- **Fault Tolerance**: the robustness of each node in a WSN plays an important role especially in the harsh environments with a high probability of node failures.
- **Scalability**: sensor nodes are typically deployed with a high density and a large number of nodes. Therefore the design of the network must be in a way that the WSN protocols could handle large amount of data.
- **Network Lifetime**: due to the limited resources in the sensor nodes especially in the energy section, the network design is very significant in order to achieve the maximum lifetime.
- **Security**: in some applications such as defense and health-care, security issues required to be considered in detail when designing the network protocol. Sometimes, high security networks require complex algorithms for implementation which is a major challenge to meet due to the limited resources.
- **Real-time**: depending on the application, the need for real-time data is evident. Therefore, in these applications, sensing, processing and transmitting the data should not cause delay in the overall system's functionality. Real-time is particularly very important in industrial applications.

Section 2.2 presents the WSNs which are implemented in the industrial domain and are called Industrial Wireless Sensor Networks (IWSNs). Later on, actuators are also integrated in the nodes which are called Industrial Wireless Sensor and Actuator Networks (IWSANs).

2.2 Industrial Wireless Sensor Networks

One of the applications of a WSN is in the industrial automation. Traditional sensory systems in a industrial domain are composed of wired communication. Due to the high cost of installation and maintenance of cables, they are not utilized in today's industrial systems. In such systems, tiny nodes are implemented on each automation device in order to measure and monitor critical parameters. The sensory data are then transferred to the base station to report any problem to the operator. This real-time data transmission is caused to detect any failure in advance and, therefore, the operator will realize and repair the device prior to decrease the machines proficiency. It also prevents the cost of replacement. The terms collaborative is also used for WSNs since the nodes are able to establish a communication among them. This brings several advantages with respect to the conventional wired sensors such as flexibility, easy and rapid deployment and smart processing ability. Therefore, the network of wireless sensors provides a high reliable automation system which responds very fast to the events in real-time with a suitable action [3].

The design of a WSN for industrial applications requires knowledge and experience in industrial domain. Understanding the sensory system is also required in order to implement the sensors by an appropriate calibration. The sensors can be implemented in motors, pumps and other inaccessible devices with the ability of not only sensing, but also processing.

Traditional control systems composed of several devices are connected via industrial networks such as Fieldbus and Profibus. Devices can be sensors or actuators. In wireless systems, the communication between sensors is performed by means of industrial wireless networks. The new sensors are replaced conventional wired sensors which are based on 0-5V and 4-20mA. In this thesis, not only sensors, but also actuators are integrated into a wireless node in order to provide more flexibility to the overall system's work. According to IEEE 1451.2 [4]: "a Smart Transducer provides functions beyond those necessary for generating a correct representation of a sensed or controlled quantity. This functionality typically simplifies the integration of the transducer into applications in a networked environment". Wireless sensors are easy to manage since they can be implemented in any network with protocols regardless of the vendors and manufacturer. They are also easy to install and replace with less effort.

Industrial wireless sensor networks are also very popular among the researchers. There are a large number of research activities in this field. Challenges and issues related to IWSNs can be found in [3]. An example of measurement and infrastructure application using wireless sensor networks is given in [6]. Industrial process monitoring is one of the useful applications of wireless sensor networks. Examples can be found in [6]-[15]. Security and safety issues regarding the wireless sensor networks in industrial environment was discussed in [16]. Redundancy plays an important role in industrial automation with respect to the production cost. Multiple usages of sensors for a platform increases the accuracy of the measurement. The connectivity among the grouped sensors is presented in [17]. Platform-based design of industrial applications for an example of manufacturing cell has been discussed in [18]. WINTeR [19] is an architecture and applications of a Wireless Industrial Sensor Network Test bed for Radio-Harsh Environment. It is an open access and multi-user test bed for experimental purposes. Wireless sensors have gained popularity in discrete manufacturing area especially in assembly lines. For instance, application of wireless sensors in a flow-line manufacturing system is given in [20] where sensors are used to monitor the status of machines to enhance the performance of the overall manufacturing system.

In this thesis, not only the sensors, but also actuators are integrated into the wireless node to enhance functionality and controllability. By this integration, the nodes have the authorization of not only sensing, but also processing, deciding and actuating. Figure 2.2 depicts the block diagram of a simple wireless sensor and actuator network. This is the node platform which contains several sections. The most important part is the network capability which allows the nodes to establish a communication link with other nodes or even devices in the network. The node also has a Central Processing Unit (CPU) along with Analogue to Digital Converter (ADC) and Digital to Analogue Converter (DAC) for interface purposes. For instance, the sensors are connected to the ADC in order to convert the analogue signals obtained from the sensors to digital numbers so that the CPU can process it.

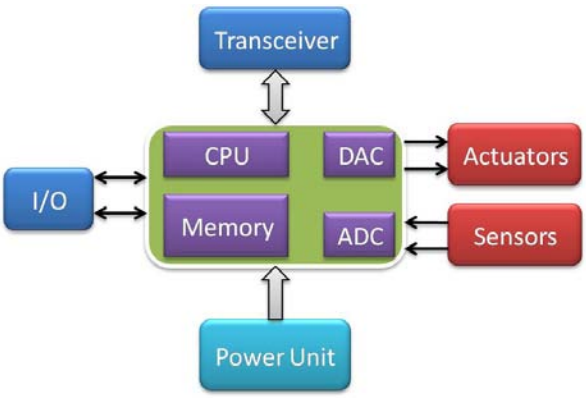

Figure 2-2: Wireless Sensor and Actuator Node

Therefore, the result of the processor is then forwarded to the actuator in order to perform the requested task. The status of the node is also broadcasted to the other nodes in the network. The power unit provides the energy required by the components. There are some input/output ports left spare for further purposes.

In industrial environments, the robustness of the nodes is very vital since there is a strong electromagnetic field generated by the induction motors, welding machines and others. This is caused which decreases the transmission quality and generates error. The performance of the nodes is directly proportional to the capability of transferring data in a short, fixed and specific time [13]. This is a key factor in the wireless sensors when employed in industrial applications. IEC 61784-2 [21] specifies that the real-time capability of a system is a measure of the response time. In some industrial applications like manufacturing, in order to reduce the cost of production, the performance of the control system must be high. That is, information exchange among the nodes must be very fast to avoid any latency.

According to [13] there is an efficient way to avoid latency, which is the traditional centralized system where all the sensors and actuators in the system are directly connected to a central processor. Therefore, the only delay in the system is the time between the sensor reading and actuator output. Consequently, employing the wireless sensors with the platform depicted in Figure 2.2 is not a feasible solution and may result in problems in self calibration and diagnostic.

Distributed scheme is very suitable for modeling wireless sensor and actuator networks. This is because of the network property which allows the nodes to be implemented in a distributed fashion. Distributed architecture has many advantages such as: easy installation and maintenance, flexibility and high performance due to the local processors which increase the real-time property of the over system. Configuration and reconfiguration of the distributed system are easy to perform since they are done locally without affecting the overall system.

Determining a proper routing technique in a wireless sensor network is a key factor since it effects the energy consumption of each node and hence the overall network lifetime. Chapter 3 discusses the routing techniques and the related energy analysis for a die-casting industrial application.

Chapter 3

ROUTING TECHNIQUES AND ENERGY ANALYSIS OF IWSAN

3.1 Introduction

The WSN technology is rapidly becoming a feasible solution for monitoring and control applications at the lowest level of manufacturing automation systems. A network of smart sensors consists of a large number of nodes which are spread out in the area in order to acquire data to monitor and control the system operation. Typically, in all WSNs there is a node called the base station which receives data from the nodes and transmits them to the server or to the network bus. Obviously, the nodes, which are in the coverage area of the base station, perform direct transmission in order to reach the base station, whereas the nodes which are out of the coverage area, transmit their data through the other nodes. The sensor nodes are energy constrained, since the power is supplied from batteries which have limited life time. Transceiver section of the node is the most energy consuming part. Therefore, designing a proper routing protocol will enable efficient utilization of the node circuitry and hence increase the network life time. In this section, different routing protocols have been presented in order to find the optimum transmission path from the nodes to the base station with minimum energy consumption. The design of an energy efficient wireless sensor network is highly application specific. Although there are many energy efficient routing protocols such as LEACH [22-24] which may work successfully in certain applications, they may not perform well for other applications. Therefore, choosing a proper routing communication protocol is highly dependent on the system factors such as the transmission frequency, the number of nodes, the distance between the nodes and the number of bits in the transmitted data.

Energy analysis for establishing a high performance network is necessary since the sensor nodes are battery powered. For this reason, choosing an energy efficient routing protocol technique is a challenging process. In this analysis, two routing protocol techniques, namely, Direct Transmission and Minimum Transmission Energy (MTE) techniques, are used in order to compare the efficiency and network life time of the sensor network for an industrial application such as the die casting manufacturing.

3.2 Problem Definition

There has been a significant amount of research in the design of routing protocols for wireless sensor networks. Different approaches and analyses in this field are given in [23]. Most of the analysis and simulation methods that have been developed are designed for randomly distributed sensor nodes and there is a lack of analytical model for specific industrial applications where the sensor nodes are evenly distributed. For instance, Low-Energy Adaptive Clustering Hierarchy (LEACH) [22] [23] [24] protocol is designed for the case where sensor nodes are randomly distributed in an environment. Several analyses and improvements on LEACH have been proposed in [25-29] however, this random distribution is not suitable for a manufacturing shop floor. In [30], different protocols have been analyzed for industrial applications without considering that the design of network for wireless sensors depends on system and environmental factors. In [31], an analysis for linear wireless sensor networks based on characteristic distance [32], has been presented. However, it does not consider the effect of other factors such as the transmission frequency, the number of nodes and the message length, on the efficiency of the sensor nodes.

In the present analysis, several factors such as the number of nodes, distance between transmitter and receiver nodes, frequency of transmission and message length are considered and, based on these factors, the final energy efficient routing protocol is established.

3.3 Energy Calculation for Radio Energy Model

The power of electromagnetic waves in a wireless channel decrease according to the power law function of the distance between the transmitter and receiver. This function determines the energy dissipation while the signal travels the path toward the receiver in a direct line of sight or multi-path fading channel model. In both models, the energy dissipation follows the power law function [33]. The definition of the power law function depends on a critical distance between the transmitter and receiver which is called $d_{crossover}$ as defined by [23] [33]:

$$d_{crossover} = \frac{4\pi\sqrt{L}h_r h_t}{\lambda} \tag{3.1}$$

where, L is the system loss factor in the propagation model, h_r and h_t are the height of the positions of the receiver and transmitter antenna above the ground, respectively, and λ is the signal wavelength and it is determined by the transmission frequency f_t and the speed of light C ($\lambda = C/f_t$).

Depending on the value of this crossover distance, two different models will be used. The Friis free space model [23] [33] is utilized when the distance between the transmitter and receiver is less than the crossover value. In this case the power law function is defined such that the attenuation in the energy level is proportional to d^2 and if the distance is greater than the crossover value, the two-ray ground propagation model will be utilized, instead. In this case, the attenuation in the energy level is proportional to d^4.

For a typical sensor node such as TelosB [34], the operating frequency varies between 2400 MHz to 2483.5 MHz. We assume that the height of the positions of the transmitter and receiver node is 1m above the ground and the system is assumed to be lossless ($L=1$). Therefore the crossover distance will be calculated using equation (3.1) as $d_{crossover}=104.6m$. Since the distance between the machines in an industrial application, such as in the die casting shop floor, is less than the crossover value and also according to [33] the path loss exponent for obstructed factories is 2 (Table 3.1), therefore, d^2 attenuation will be used in our model.

Table 3-1: Path loss exponent for different environment

Environment	Path Loss Exponent, *n*
Free space	2
Urban area cellular radio	2.7 to 3.5
Shadowed urban cellular radio	3 to 5
In building line-of-sight	1.6 to 1.8
Obstructed in building	4 to 6
Obstructed in factories	2 to 3

Table 3.2 presents the value of crossover distance for different sensor nodes which operate in different transmission frequencies.

Table 3-2: $d_{crossover}$ for different frequencies

Carrier Frequency	$d_{crossover}$
900MHz	37m
1300MHz	54m
1700MHz	71m
2100MHz	87m
2500MHz	104m

Therefore based on the carrier frequency of sensor nodes which are deployed in manufacturing shop floor and the calculated crossover value; the path loss exponent can be determined.

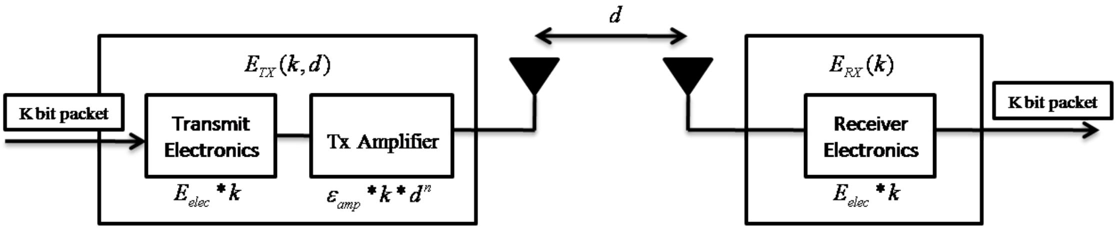

Figure 3-1: Radio Energy Model [23]

The simple radio energy model which has been proposed by [23] is considered in our analysis (Figure 3.1). In this model, d is the distance between the transmitter and receiver. In the transmitter, there are two parts that consume energy. The energy, which is consumed by the electronic circuit in order to process k-bit packet is $E_{elec}*k$ where E_{elec} is the energy consumed to process one bit, and the energy which is consumed in order to amplify and transmit k-bit packet over the distance d is $\epsilon_{amp}*k*d^n$. Therefore the total energy dissipated in the transmitter and receiver, in order to send k-bit packet over the distance d using the assumed transmission model is:

$$E_{TX}(k,d) = E_{elec} * k + \in_{amp} * k * d^2 \qquad (3.2)$$

In this equation the Friis free space model is considered since the distance between the transmitter and receiver is less than the $d_{crossover}$. Similarly, in the receiver part, the energy that is consumed by the electronic circuit to process the k-bit received packet is:

$$E_{RX} = E_{elec} * k \tag{3.3}$$

3.4 Analysis of Routing Protocols in WSNs

Designing routing protocols in wireless sensor networks has been one of the most important research areas that many researchers had studied and developed different techniques. Nodes in wireless sensor networks are energy constrained and that is the challenge which routing techniques have to be optimized in order to conserve the nodes energy and hence the increase the overall network life time. There are different routing techniques that are developed by researchers for different environments [35]. Among those protocols, two methods are used in our analysis, namely, the direct transmission and the minimum transmission energy (MTE) methods. These two are the most suitable methods which can be utilized in industrial applications.

In the direct transmission method, each node transmits its data directly to the so called base station which is in charge of collecting data and transferring them to a PC or Internet for further processing. Therefore, each node consumes some amount of energy in order to transmit data to the base station. Obviously, the nodes which are far from the base station consume more energy in comparison with the nodes which are near to it. Therefore, this method fails when the nodes are spread out in vast environments such as lakes, forests, etc. Moreover, if the base station is out of the coverage area of the nodes, transmission fails. On the other hand, this method may be utilized for small environments such as factory workshops or houses.

Consider the network shown in Figure 3.2 in which a node is located in distance d from the base station:

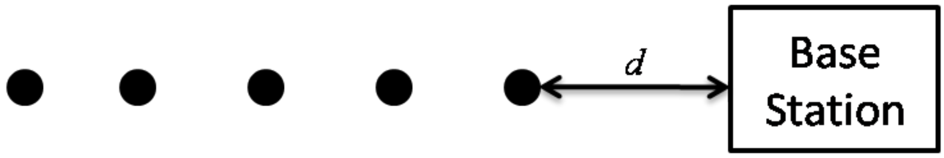

Figure 3-2: Direct Transmission

The energy that a node expends in order to transmit *k* bits to distance *d* consists of the energy that the electronic circuitry and the transmitter amplifier consume [24]. That is:

$$E_{direct} = k(E_{elec} + \in_{amp} d^2) \quad (3.4)$$

This is the amount of energy that each node consumes in direct transmission. This value is proportional to the number of bits and it is also a function of distance d. As a result, the message size also needs to be optimized for energy saving.

In MTE transmission, each node finds the shortest path on the way to the base station and sends its data through the intermediate nodes. Therefore, instead of using a high energy transmission path, the node messages are transmitted using several intermediate low energy transmission paths [23]. For instance, consider the network shown in Figure 3.3:

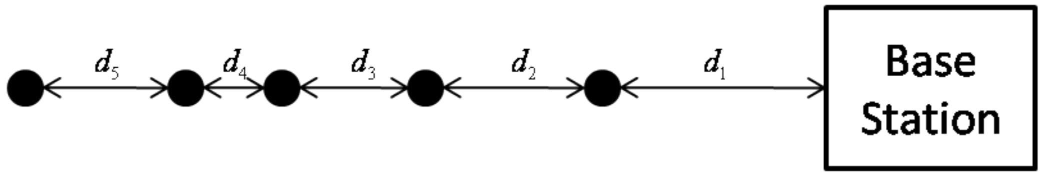

Figure 3-3: Wireless sensor linear network

where the nodes are located along a line with different distances. Consider the transmission of *k* bits from node *n* to the base station. This node sends its data through the other *n-1* nodes. Therefore:

$$E_{n\ to\ B(MTE)} = E_{n\ to\ n-1} + ... + E_{2\ to\ 1} + E_{1\ to\ \mathrm{B}} \tag{3.5}$$

where $E_{n\ to\ B\ (MTE)}$ is the energy that is required for the transmission of *k* bits from node *n* to the base station. In this case, there are *n* transmits and *n-1* receives. Substituting equations (3.2) and (3.3) into equation (3.5), we get:

$$\begin{aligned} E_{n\ to\ B(MTE)} &= [k(E_{elec} + \in_{amp}\ d_n^2)] + ... + [k(E_{elec} + \in_{amp}\ d_2^2 + E_{elec})] \\ &+ [k(E_{elec} + \in_{amp}\ d_1^2 + E_{elec})] \end{aligned} \tag{3.6}$$

where, except the first transmission, all the intermediate nodes consume energy while transmitting and receiving data. Therefore upon simplification, equation (3.6) will be:

$$E_{n\ to\ B(MTE)} = k[(2n-1)E_{elec} + \in_{amp} \sum_{i=1}^{n} d_i^2] \tag{3.7}$$

Now, consider the network in Figure 2 and transmission of a k bit message from node n to the base station using direct transmission. Therefore, by rewriting equation (3.4) for n nodes we get:

$$E_{n\ to\ B(direct)} = k[E_{elec} + \in_{amp} (\sum_{i=1}^{n} d_i)^2] \tag{3.8}$$

Direct transmission expends less energy than MTE communication if [23]:

$$E_{n\ to\ B(direct)} < E_{n\ to\ B(MTE)} \tag{3.9}$$

Applying equation (3.7) and (3.8) into (3.9), we get the condition:

$$[(\sum_{i=1}^{n} d_i)^2 - \sum_{i=1}^{n} d_i^2] < 2(n-1)(\frac{E_{elec}}{\epsilon_{amp}}) \tag{3.10}$$

If the above condition is satisfied, the direct transmission method provides more efficient data transmission than that of the MTE. The right-hand side of equation 3.10 is constant since n is the number of nodes, E_{elec} and ϵ_{amp} are the constant values of circuitry energy for a sensor node. If the left-hand side of equation 3.10 is maximized over a boundary the distance which fall into this boundary will be suitable for direct transmission. Therefore we can define $f(d_i)$ as follows:

$$f(d_i) = [(\sum_{i=1}^{n} d_i)^2 - \sum_{i=1}^{n} d_i^2] \tag{3.11}$$

Our optimization problem will be to maximize equation 3.11 as follows:

$$\begin{aligned} &\max\ f(d_i) \\ &0 < d_i \leq D \end{aligned} \tag{3.12}$$

Where D is the boundary. Solving the maximization problem yields the condition that $f(d_i)$ is maximum when all the distances are equal within the boundary. That is:

$$d_1 = d_2 = ... = d_n = D \tag{3.13}$$

Upon rewriting equation 3.10, we define the critical distance as $d_{critical}=D$ as follows:

$$d_{critical} = \sqrt{\frac{2E_{elec}}{n \in_{amp}}} \tag{3.14}$$

Direct transmission requires less energy in comparison with MTE if the distances between the nodes are between $0 < d_i < d_{critical}$.

3.5 Simulation Result

In this section, a case study related to a die-casting factory (Figure 3.4) [36] is presented. This company, Sahin Metal, is located in Istanbul, Turkey and produces various aluminum-alloy part and light alloy automotive variants. Performances of direct transmission and MTE communication protocols are compared by simulating the energy consumption of the sensor network under variable node numbers, distances and message lengths. On the shop floor, there are die casting machines, trimming presses, calibration presses and vibratory debarring machines. Each machine is assumed to be equipped with a sensor node in order to monitor the machine status as well as other physical characteristics such as die temperature and pressure. For simulation purposes, the following parameter values, which are given in [22] [23] [24], are used: the energy for transmitter/receiver electronics and transmit amplifier are assumed as E_{elec}=50 nJ/bit, ϵ_{amp}=100 pJ/bit/m^2. Transmission of a 100-bit message for 6 nodes which are used for the die casting machines (M1 to M6) are considered. Using the developed equations, the energy that a node consumes when the direct and MTE protocols are implemented in Figure 3.5.

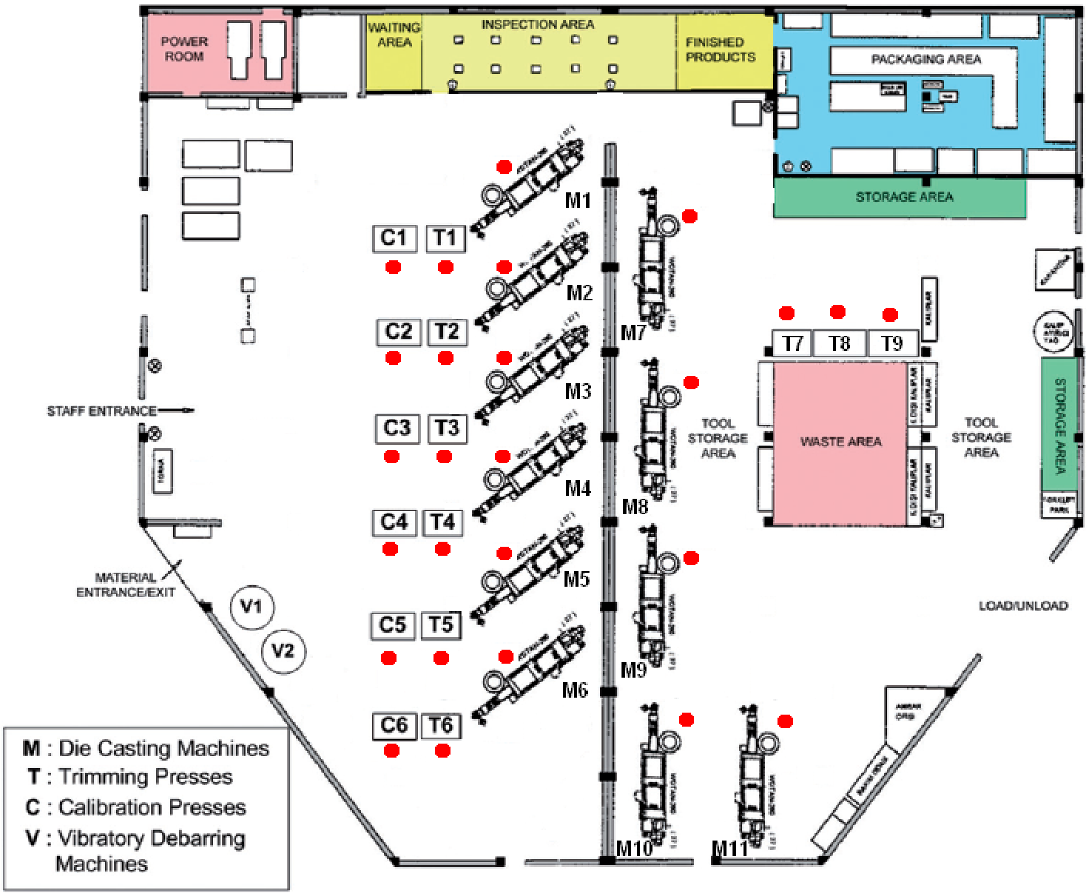

Figure 3-4: Die casting shop floor [36]

It is clear from Figure 3.5 that there is a significant raise in energy in direct method as the distance increases, while in the MTE, the energy increases slightly. The cross-section point two graphs occurs at the critical distance ($d_{critical}$) and in this case the critical distance is $d_{critical}=12.90m$. Therefore if the die machines are located less than critical distance, direct transmission is preferred.

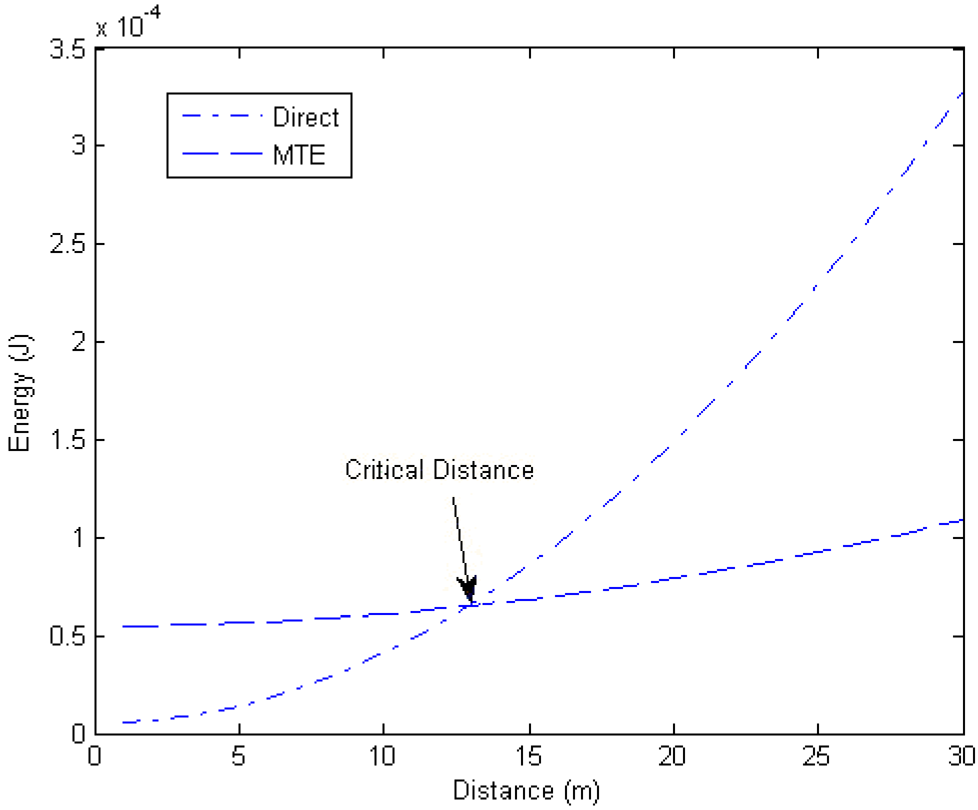

Figure 3-5: Consumed energy of a node versus distance

Figure 3.6 presents the effect of message length and distance on the energy consumption on a 3D graph. It is clear that the critical distance is independent of the message length and it is constant for different values of *k* (consistent with Equation 3.11).

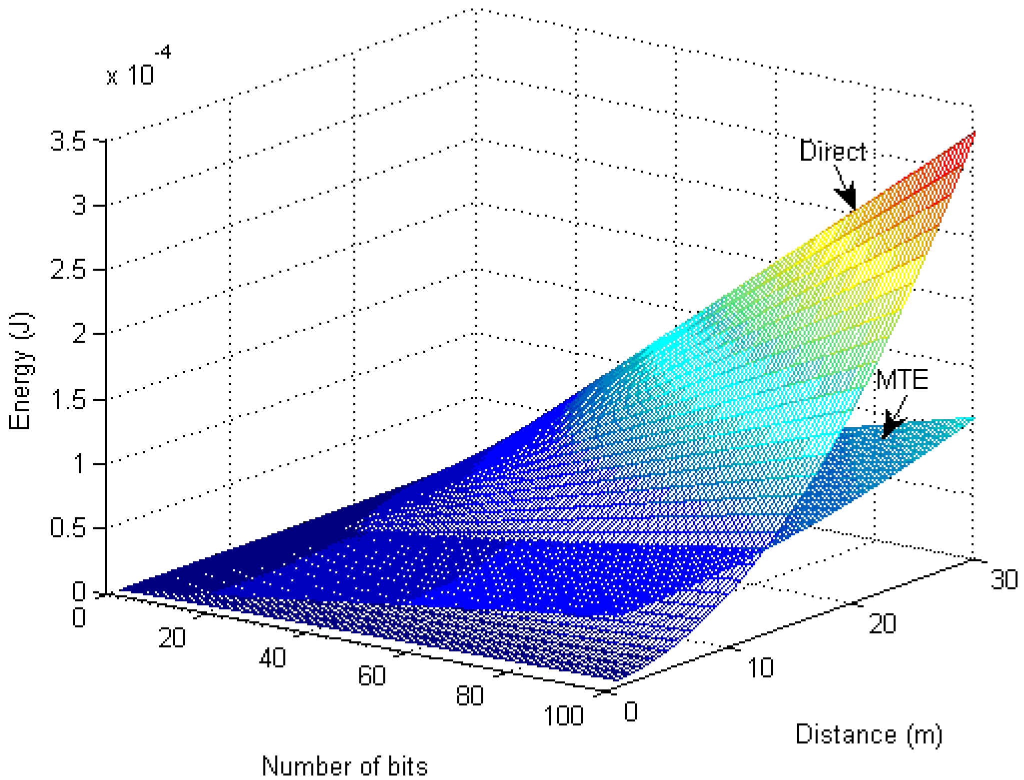

Figure 3-6: Expended energy of a node for different distances and message length

It is clear that for short distances (less than the critical distance) the number of bits of the message effects the dissipated energy of direct transmission. However, as the distance increases, MTE routing performs better, even for different message lengths. Figure 3.7 presents the effect of number of nodes on the total energy dissipation for both direct and MTE transmission.

As the number of nodes increases, for short distances, direct transmission performs better than MTE. The same comment can be made for long distances and less number of nodes. Otherwise MTE routing is preferred.

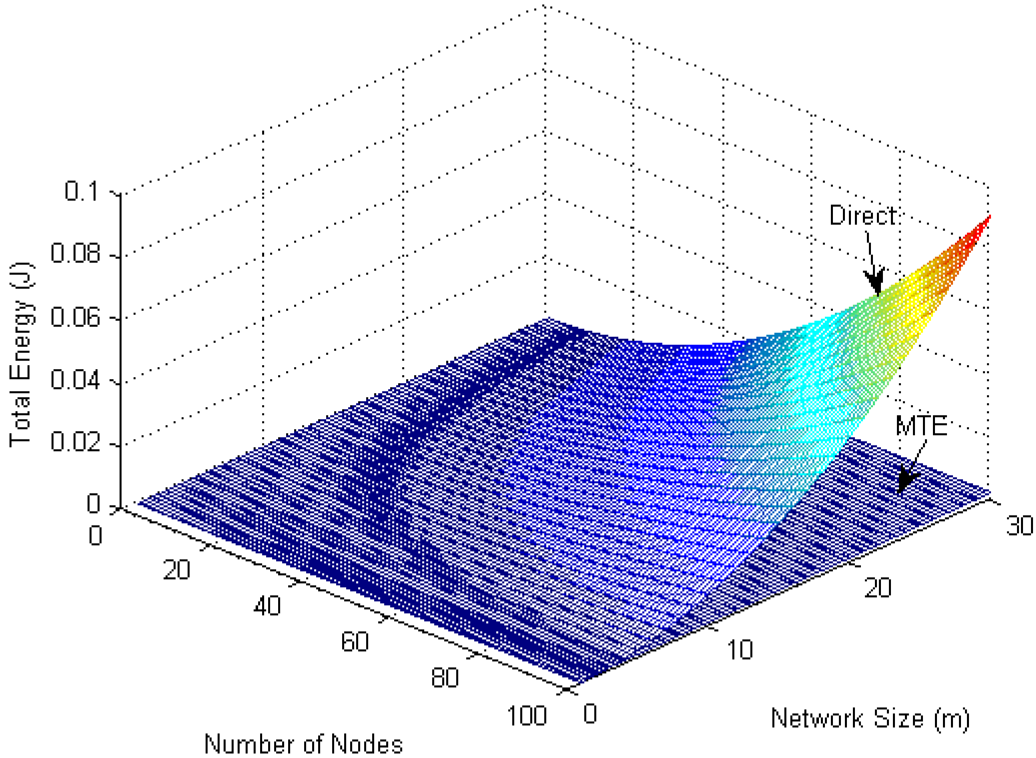

Figure 3-7: Total energy expended for different number of nodes as network size increases

Figure 3.8 represents the total energy expended as the electronic energy increases. The common points of the graphs form a line which indicates the critical distance. The critical distance depends on the energy dissipated by the electronic circuitry of the nodes.

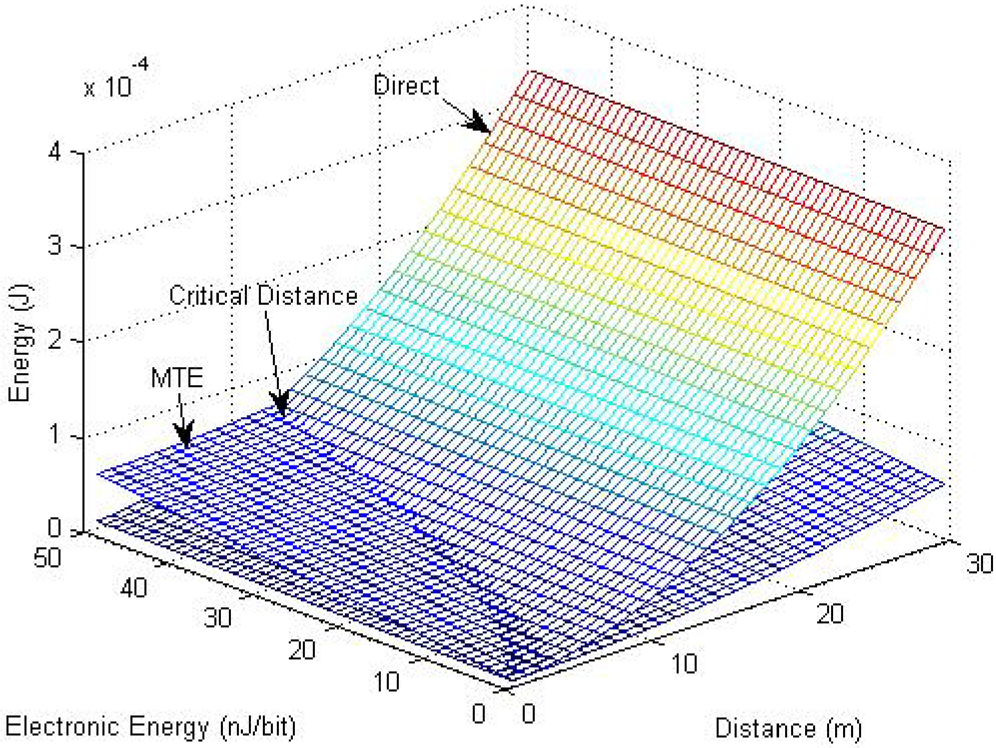

Figure 3-8: Total energy expended versus electronic energy for different distances

Table 3.3 shows the expended energy for both direct and MTE transmission as the path loss exponent is increased. As mentioned before, the path loss exponent can be selected based on the value of $d_{crossover}$. In this table, the values of dissipated energy for transmitting a 100-bit message over a 5m distance are presented. For the distances which are greater than $d_{crossover}$, the energy consumed by the system using direct transmission is almost ten times that of the MTE. Therefore, depending on the frequency of transmission and, hence, the corresponding value of $d_{crossover}$, the path loss can be selected and finally the routing protocol can be established.

Table 3-3: Expended energy for different path loss exponent

Path Loss Exponent, *n*	Direct x 10^{-5} (J)	MTE x 10^{-5}(J)
2	1.40	5.65
2.4	4.01	5.78
2.6	7.43	5.89
3	27.50	6.25
3.4	100	6.92
3.6	210	7.47
4	810	9.25

3.6 Discussion

The analysis of dissipated energy during data transmission in the WSN of a Die Casting shop floor clearly demonstrates that the performances of the Direct and MTE protocols are highly dependent on various factors such as the number of nodes, frequency of transmission, distance between the nodes and the number of bits in the transmitted data. Therefore, a specific method may not perform well for all industrial applications. Although a routing protocol may be generally efficient in terms of energy dissipation (LEACH), however, it may not be useful for a specific application such as a die casting company. For instance, considering the die casting machines (M1 to M6), which are equipped with sensor nodes, the following parameters are required to be considered before deciding on the routing technique:

- *Frequency of transmission*: sensor nodes from different vendors have different transmission frequencies. Based on the value of transmission frequency, the crossover distance and, hence, the path loss exponent can be determined.
- *Message Length*: each node consists of several sensors. Upon finishing the reading process, these data are transmitted for further processing. Transmitting unchanged data should be avoided, which will increase the message length and, hence, the transmission energy.

- *Machines Distance*: the third parameter is the distances between machines. Consider the die machines M1 to M6. If the distance between two adjacent machines is less than the critical value the direct transmission is selected rather than the MTE.

The results have illustrated that the message length and the content of the data to be transmitted in every transmission plays an essential role in the accuracy of the measurement process. Although a reduction in the message length reduces the transmission energy dissipation, on the other hand, the precision of the sensed data is degraded. In fact this energy saving should not have an effect on the overall system performance since the industrial-process measurement requires a robust control architecture in order to avoid any failure in the system. As a consequence, in cases where all the sensed data by the sensors in a node have to be transmitted in every transmission, other factors such as the frequency of transmission and the machine distance should be considered in order to reduce the consumed energy.

Also it should be mentioned that the location of the base station has a pivotal factor in the network life time. For instance, the location of base station for die machines (M1 to M6) can be along the machines line. This will result in transmitting data using a linear topology network which is simple and easy to configure and deploy.

Chapter 4

MULTI-AGENT FRAMEWORK FOR IWSAN

4.1 Introduction

Multi-Agent Systems (MAS) is a framework in order to model and implement the distributed autonomous systems. A MAS contains a network of agents which are assumed to be autonomous, intelligent, self-organizing and collaborative. It has been recognized as one of appropriate schemes for modeling wireless sensor networks since sensor nodes are autonomous and self-organize. This section identifies the modeling approach of WSN based on MAS.

4.2 Intelligent Agent

Sensor nodes are devices which are deployed in a distributed scheme in order to collaborate and achieve the system's goal. Typically sensor nodes are deployed in harsh environments such as lakes, mountains, forests and industrial sections. This will reduce the user's ability to control the functionality of the sensor nodes such as fixing, changing and even charging the batteries. As a result, sensor nodes must be built robust enough to decrease the dependencies of a node on the others. This can be achieved by providing more ability to the sensors or implementing intelligent agents in the sensor network system [37]. For instance, as discussed in the previous chapter, transceiver section of a sensor node is the most energy consuming part. Therefore, by means of an intelligent agent can effectively reduce the usage of the transceiver when the energy is less than a particular threshold.

There are two types of intelligent agents: static and mobile [37]. In static, the agent is inside a sensor node and perform the decision making process based on the available information and resources. In case of mobile, agents can migrate from one node to another in order to collect or analyze the data. This is called logical mobile agent. There is also physical mobile agent where it can move the node and place it in another location. This allows the network to be controlled in a distributed scheme and benefits from its advantages. Apart from the software part, there is a need to have a middleware to support the intelligent agents and performing control and monitoring remotely.

4.3 Intelligent Agent Requirements

In order to implement the agents in the sensor networks, there are some requirements needed to be considered. For instance, the problem of complex computation in distributed systems can be solved by a proper middleware, that is a software to connect the application to the hardware devices. Middleware is based on Application Programming Interface (API) that requires fewer amounts of program codes which can be easily implemented on sensor nodes.

The middleware should provide the communication ability to the agents to establish collaboration among the nodes. This is very important in a distributed system. The middleware should also control the agents in the network in order to consume the node’s resources efficiently.

There are some middleware developed for implementing MAS on wireless sensor networks which will be explained briefly.

4.3.1 LIME

LIME is a middleware that identifies several APIs to be utilized in mobile agents. It is based on JAVA and Linda. Linda is method which uses the shared memory for computing in the distributed networks. Linda is based on the tuple spaces which are used to read, write and delete data. Lime was designed for the mobile sensor nodes where each node can access the other node's tuple space.

4.3.2 Agilla

Agilla [5] is designed for mobile agents in wireless sensor networks. It is implemented in MICA2 sensor node on top of TinyOS (the operating system designed for networked embedded systems). Agilla offers the agents to move the algorithm codes and the execution which is currently performed in a node. This is mobile logical agent which can simplify the network.

4.3.3 Impala

Impala [81] was designed to support the inter node communication such as between sensors, interfaces and other parts. This middleware solves many problems such as correctness, modularity and easy utilization. The applications are implemented on top of Impala in order to provide sensory access and interfaces.

4.3.4 Jade

Java Agent Development Framework (Jade) [38] is another middleware developed for multi-agent systems on the basis of Java and it is fully compliant with Federation for Intelligent Physical Agent (FIPA) [39]. It uses the advantages of Java programming such as portability, interoperability and platform independent. Jade is very useful for data fusion in distributed fashion. A case study on fire detection based on Jade is given in [40].

4.4 Related Research

There are a large number of researches going on the multi-agent systems and wireless sensor networks. Power management is a key factor in WSN since the nodes are battery powered. In [41] an agent based framework for controlling and management power has been proposed. A tiny agent called CSIRO was presented in [42] working in decentralize scheme for controlling th energy resources in wireless sensor networks. A middleware architecture for wireless sensor networks which is a FIPA compliant has been discussed in [43]. Security is another issue that has to be considered while designing intelligent agents in sensor networks. In [44], a secured MAS has been implemented to manage the network behavior. More researches and implementation of intelligent agents and multi-agent systems can be found in [45-52].

There are some concerns relating the intelligent agents in sensor nodes. The most important issue is that the available resources are limited in a sensor node especially in the power section. Sensors are required to perform processing in a minimum time in order to save more energy. This will limit the capabilities of the intelligent agent. When the term intelligent is used, soft computing approaches are required, such as: neural network, fuzzy logic, genetic algorithm and heuristic methods. Unfortunately, these methods require a huge amount of memory and processing ability which is not efficient to be implemented in sensor nodes.

In this context, a new standard has been proposed recently for implementing multi-agent scheme for distributed systems in industrial automation. However, there is no research on implementing this new standard on wireless sensor networks especially for industrial automation. In this thesis, a new modeling approach for development of distributed industrial wireless sensor and actuator networks will be presented. However, before proceeding to the modeling approach, some knowledge and background on the new standard is required. This will be explained in the proceeding chapter.

Chapter 5

IEC 61499 FUNCTION BLOCKS FOR INDUSTRIAL-PROCESS MEASUREMENT AND CONTROL SYSTEMS

5.1 Introduction

IEC 61499 Function block has been recently adopted as a standard in Distributed Control Systems (DCS). It is developed by International Electrotechnical Commission in 2005 [53] for Industrial-Process Measurement and Control Systems (IPMCS). The idea of Function block (FB) was first utilized in IEC 61131 for Programmable Logic Controller (PLC) programming. However due to the lack of flexibility and reusability, it is modified and became as the basic building block and functional software unit in IEC 61499. It tends to be utilized for hardware-independent applications in order to increase the interoperability and configurability between different device vendors. In this thesis, IEC 61499 standard is the basic building block of the proposed methodology. This standard has four parts where each part will be explained in this chapter.

5.2 Evolution of Automation Systems

This section identifies the evolution of the automation systems which has been occurred during the recent years. The evolution will be demonstrated with a simple automation example shown in Figure 5.1. This section has been extracted from [54].

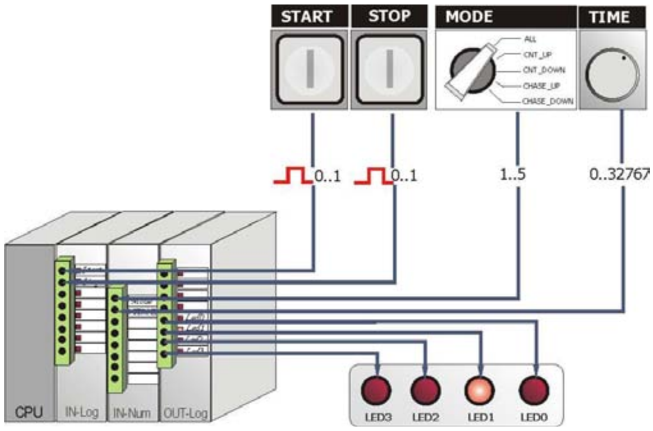

Figure 5-1: A simple automation example called Flasher [54]

In this example there is a PLC device which contains four modules. The first one is the CPU which is used for processing the algorithms and programs. There are two switches namely, Start and Stop connected to the second module. Mode and Time are connected to the third module which defines the system’s mode and its corresponding time. Ultimately, the LEDs are connected the fourth module for demonstrating. The scenario is as follows: use pushes the start button and then based on the selected time and mode, LEDs blink in different shape. The corresponding program is written based on IEC 61131 PLC standard where the program code will be executed sequentially from the start to the end. However there are some problems with this system. Adding a new function requires the integration with the old one. This integration sometimes causes problems due to the compatibility difficulties. Adding new devices require adding new cabling and this will cause difficulties when complex systems are implemented since huge amount of wiring required. Finally, the current system lacks the re-configurability due to the centralized architecture. Configuration requires the whole system to be changed and this will increase the cost of adopting new configurations.

In order to overcome the mentioned problems, flexible distributed scheme has been proposed (Figure 5.2). This scheme is based on networking where each device, sensor and actuator has a special hardware to have network capability. However the system is still based on central processing architecture where the communication between CPU and inputs/outputs is via the special network protocols. Flexible distributed architecture is more flexible due to the distributed scheme since all the devices are connected to the CPU by a single wire and this facilitates the addition or removing devices.

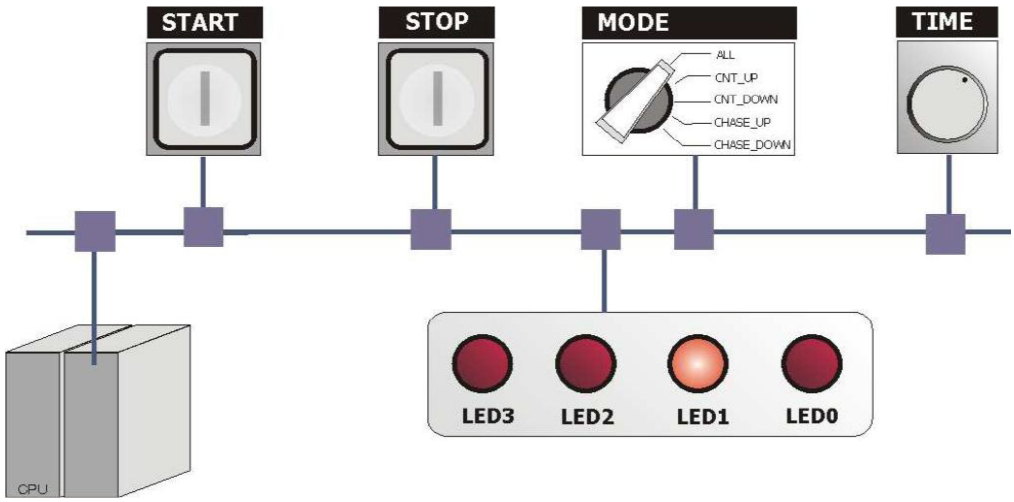

Figure 5-2: Flexible Distributed Configuration [54]

Flexible distributed architecture can be reconfigured very easy with respect to the previous scheme. The network can also be wireless to provide even more flexibility.

In order to implement the fourth generation of the controllers, each device in the network will be a software components and it is implemented by means of object-oriented software. This is shown in Figure 5.3.

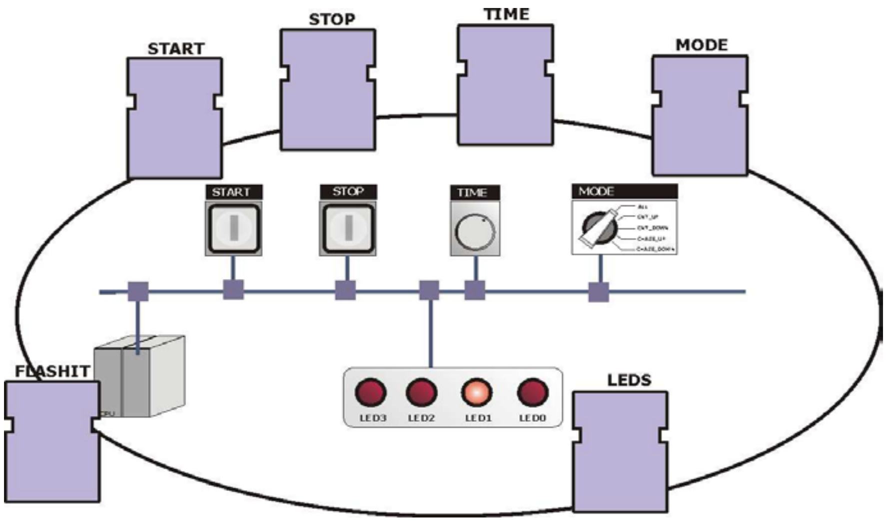

Figure 5-3: Object-Oriented Software implementation [54]

These components are called function blocks in IEC 61499 standard. each function block has the control over its own operation while collaborating with others to achieve the overall system's goal.

The next generation of the controllers is called intelligent control where there is a network of smart objects with not only the network capability, but also processing. In this scheme, there is no central processor but the locals. This technology is called agent based systems which was explained in the previous section. Table 5.1 summarizes the evolutions of the automation systems in 5 different categories.

Table 5-1: The evolution of the automation systems

	Generation 1 Relays	Generation 2 PLCs	Generation 3 Powerful PLCs	Generation 4 Embedded Controller	Generation 5 Intelligent Control
Automated routing operations	X	X	X	X	X
Programmability		X	X	X	X
Configurability			X	X	X
Distributed flexible architecture				X	X
Self Configurability					X

5.3 IEC 61499 standard in Factory Automation

The term Function Block (FB) is a famous and commonly utilized in the field of engineering. As it implies from the name, it is a block capable of performing one or more function. The similar idea has been used in IEC 61499 standard. In the conventional control systems which PLCs was adopted, the program is written based on several function blocks. However, due to the centralized nature of the system, the reusability of each function block in another application is limited. This is resulted in reducing the flexibility of the system as well. One solution is to distribute the function blocks in a network so that all the applications can use the function blocks. This is a key factor in IEC 61499 standard.

IEC 61499 standard has identified the key concepts for design, development and implementing for the industrial-process measurement and control systems. It is mainly focused on the distributed control systems and it illustrates the way for using

FB in complex industrial systems. The standard also benefits from the object-oriented programming to extend the interoperability between devices from different vendors. eXtensible Markup Language (XML) is utilized for the data exchange between software components. Although this standard gains popularity among the researchers recently, it is very difficult to employ this standard in industry. According to [54], “several aspects of this standard are unfamiliar to most practitioners of control systems engineering, especially the idea of distributed applications, event driven execution control and service interface function blocks”. IEC 61499 standard also does not identify the user requirements and design phase. In the next sections the four part of this standard will be presented.

5.4 IEC 61499-1: Architecture

IEC 61499 standard has three levels of abstractions namely, System, Devices and Resource. Function block is the basic building block of the standard [55].

5.4.1 System Model

An industrial-process measurement and control system is composed of several devices interconnected using a communication network. This communication network can be arranged in a hierarchical approach (Figure 5.4)

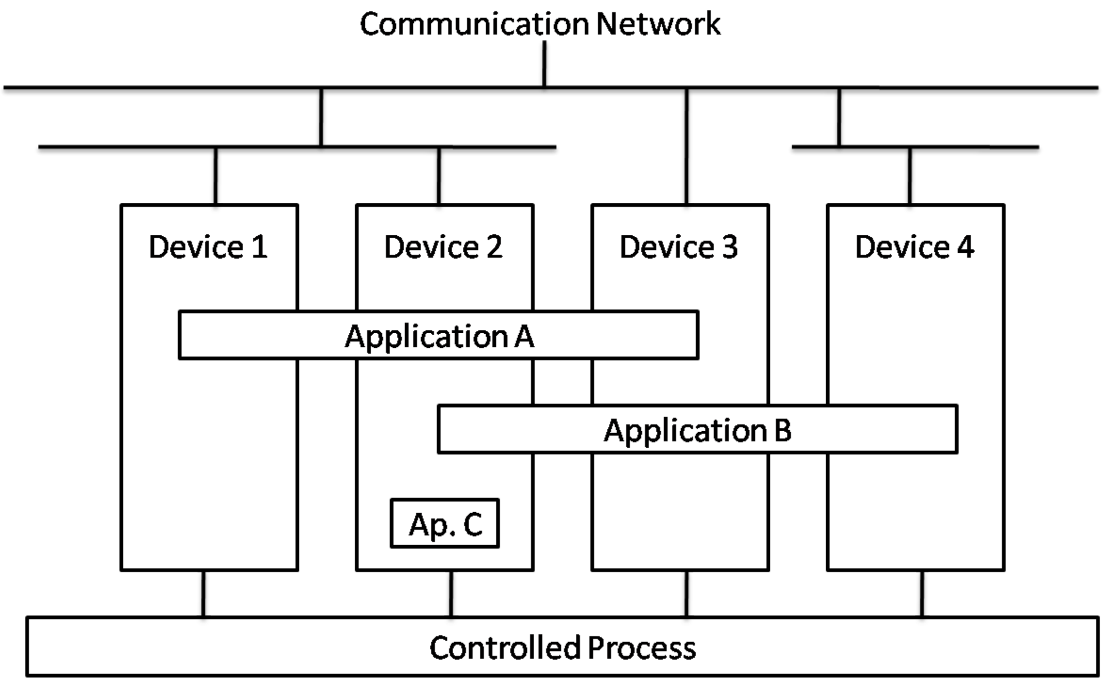

Figure 5-4: System Model [56]

In an IPMCS, an application may use one or more devices. For instance, Application A utilizes device one, two and three whereas Application C utilizes device 2 only. Applications are distributed among the devices. For example in an application, the data reading can be in a device, processing in another and issuing output data in a third. A device should be defined as one of the devices types defined in part 4 of this standard.

5.4.2 Device Model

A device in a system in IEC 61499 standard can have zero or more resources with at least one interface. The interfaces are process interface and communication interface (Figure 5.5). Process interface is in charge of collecting data from the physical entities (Inputs/Outputs). The information exchanged between resources is performed by data and events. Communication interface is used for exchanging information among the resources. Both data and interfaces can be exchanged via communication interface.

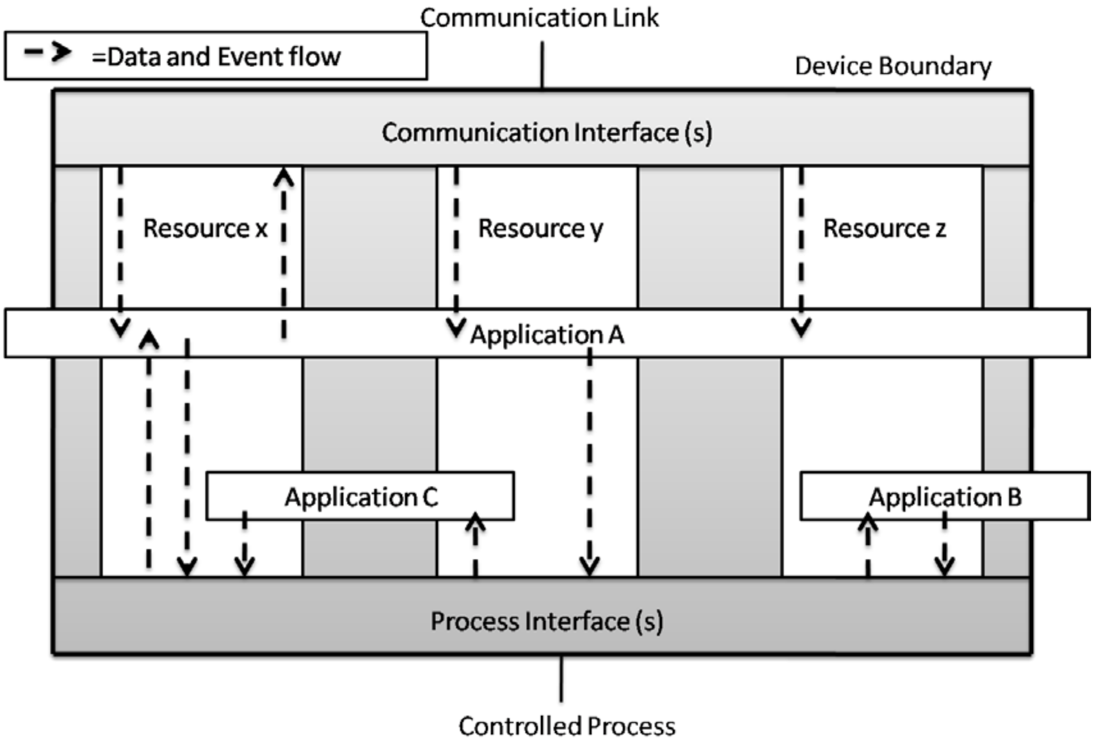

Figure 5-5: Device Model [56]

5.4.3 Resource Model

A resource is a functional unit which has full control over its own operation. The key factor here is that the operation of each resource is independent from the others and any change, modification and (re) configuration does not affect the other resources reside in a device. However, the status of each resource will be shared among the other resources in order to perform collaboration and negotiation to achieve the device's goal.

As it is depicted in Figure 5.6, a resource is composed of a network of interconnected function blocks. As it is mentioned before, there are two types of Service Interface Function Blocks (SIFB) namely, process and communication interface. Communication interface is used for mapping between the resources whereas process interface provides exchanging data between the physical entities.

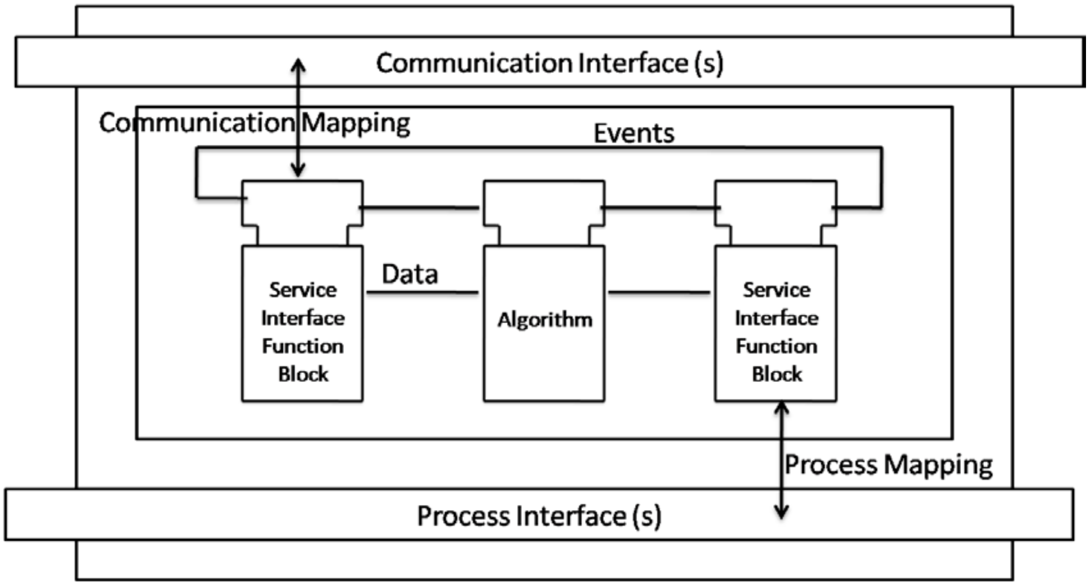

Figure 5-6: Resource Model [56]

Resource is utilized to receive data/events from other resources or from the physical devices by means of service interfaces.

5.4.4 Application Model

An application is network of function blocks with data and event inputs and outputs. It may consist of sub-application which also contains a network of function block. It may also be distributed between one or more resources within the same or different devices.

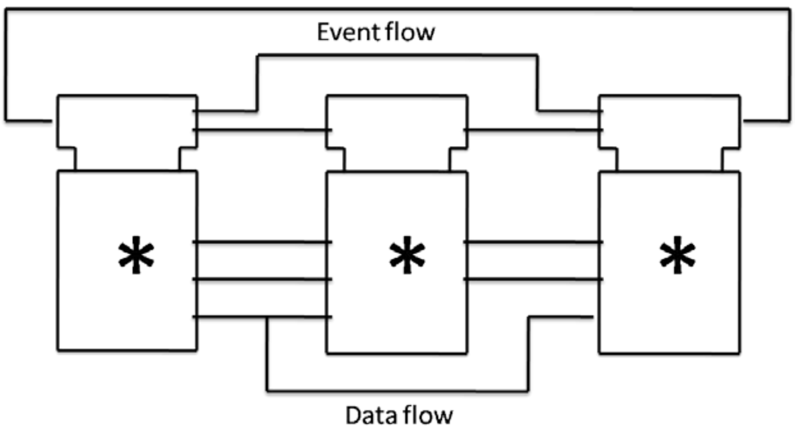

Figure 5-7: Application Model [56]

5.4.5 Function Block Model

A function block is the basic building blocks of IEC 61499 standard and it is the functional unit of the software. As it is depicted in Figure 5.8, it is mainly composed of two parts: head and body. The head is responsible for event flow. It contains a set of event inputs which are received from other function block event connection and can affect on the execution of one or more algorithms reside in the body. Event outputs are issued when the algorithm is completed. The body is composed of a set of data inputs and outputs which receive and transmit the data from/to other function blocks. Execution Control Chart (ECC) resides in the head of function block to schedule the algorithms execution in the body.

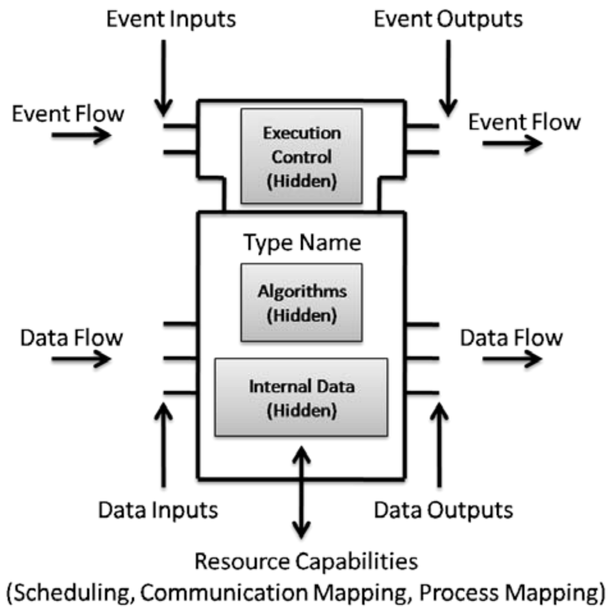

Figure 5-8: Function Block Model [56]

The ECC, algorithms and internal data are hidden from the outside of the function block. Figure 5.9 presents the execution model of the function block.

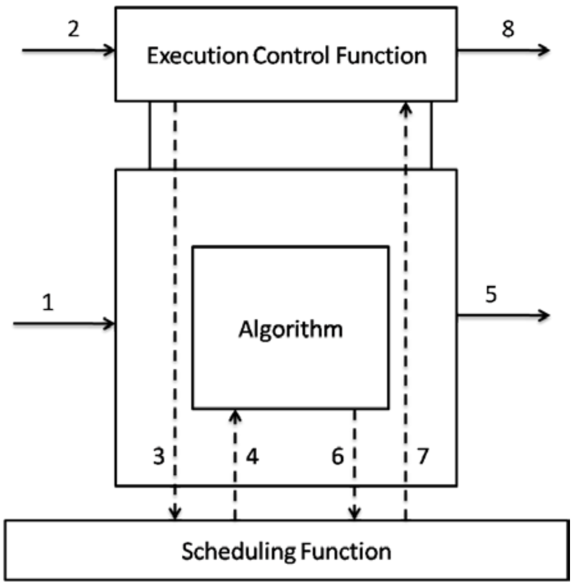

Figure 5-9: Function Block Execution Model [56]

The sequence of the function block execution is as follows:

1. Input data are received (t_1)
2. The corresponding events are made available (t_2)
3. The execution control function informs the scheduling function to schedule the related algorithm (t_3)
4. The algorithm execution starts (t_4)
5. The algorithm is completed and the relevant output data are made available (t_5)
6. The scheduling function is informed by the completion of the algorithm (t_6)
7. The scheduling function notifies the execution control function (t_7)
8. The relevant output events are issued (t_8)

There are time differences defined in this part shown in Table 5.2.

Table 5-2: Function Block Execution Time

Time	Description
$t_2 - t_1$	Setup Time
$t_4 - t_2$	Start Time
$t_6 - t_4$	Algorithm Time
$t_8 - t_6$	Finish Time

The execution control chart's behavior can be presented by finite state machine diagram (Figure 5.10).

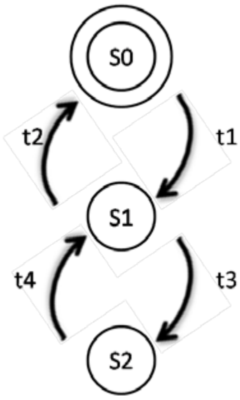

Figure 5-10: ECC State machine diagram [56]

S_0 is the idle state, S_1 is the scheduling algorithm state and S_2 is the state where the FB waits for algorithm to finish.

5.4.6 Distributed Model

Any application can be implemented in a distributed fashion by distributing the corresponding function blocks in different resources. Resources can reside in one device or they can be distributed in different devices. This distributed scheme shall not affect the functionality of the resources and devices. However, the time used for resources to communicate will affect the overall timing schedule.

There are two types of function blocks namely, basic and composite. A basic function block has an ECC to control its own operation whereas in the composite function block, several function blocks reside in one function block.

5.5 IEC 61499-2: Software Tools Requirements

Part II of IEC 61499 standard addresses the software tools requirements. According to [56], the definition of library element is as follows: “The collection declarations applying to data type, function block type, adapter type, sub-application type, resource type, and device type or system configuration”. Therefore a software tool compliant with IEC 61499 standard should support the library element and the data types. It also must be able of exchanging the library element with other software tools. This exchange should be based on Document Type Definitions (DTDs). The DTD is defined based on W3 standard and it is based on XML language. Table 5.3 addresses the DTDs used in IEC 61499 standard.

Table 5-3 Document Type Definitions (DTDs)

DTD	Library Elements
DataType	DataTypeDeclaration
FBType	FBTypeDeclaration
SubapplicationType	SubapplicationTypeDeclaration
AdapterType	AdapterTypeDeclaration
ResourceType	ResourceTypeDeclaration
DeviceType	DeviceTypeDeclaration
System	SystemConfiguration

An Engineering Support System (ESS) for IPMCS shall utilize the DTD library elements which form the XML file for data exchange. All the elements used in the XML file must be compliant with the DTD library element. This will allow the users to exchange the developed system easily.

5.6 IEC 61499-3: Application Guidelines

This part discusses about the application guidelines of IEC 61499 standard along with some tutorials and examples of IPMCS [57].

5.7 IEC 61499-4: Rules for Compliance Profiles

This part identifies the frameworks for design and development of profiles compliant with part I and part II of IEC 61499 standard for promoting interoperability, portability and configurability (Figure 5.11).

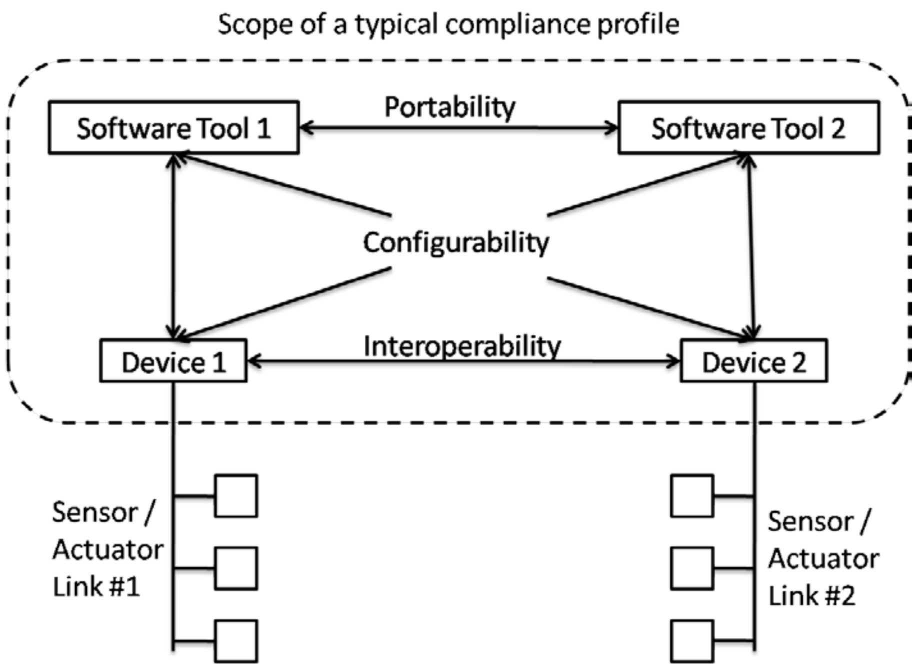

Figure 5-11: Scope of a typical compliance profile [57]

Consider the profile in the above figure; there are two software tools which are connected to two different devices from different vendors. The software tools exchange the information and it is called Portability. The devices shall communicate regardless the device's specifications (Interoperability) while the sensors and actuators may not be interoperable. However, interoperability may be extended to the sensors and actuators level. A software tool from one profile should be able to configure the device under the other software tool’s profile. The rules for achieving these specifications are given in the part 4 of IEC 61499 standards [58].

5.8 Softwares

This section addresses the available software tools for design and development of IPMCS compliant with IEC 61499 standard.

5.8.1 Function Block Development Kit

Function Block Development Kit (FBDK) [59] is an engineering design software based on Java and it is designed for development of IEC 61499 standard systems. It is originally designed by Rockwell Automation Company. FBDK has an editor environment which runs based on the developed Java classes. The Integrated Development Environment (IDE) supports the graphical implementation of function blocks in all level of abstractions. When the design implemented, FBDK uses Function Block Run-Time (FBRT) in order to execute the program. FBRT is a software which is implemented by means of Java virtual machine and therefore it is executable on any machine which supports Java. a screenshot of FBDK editor is given in Figure 5.12.

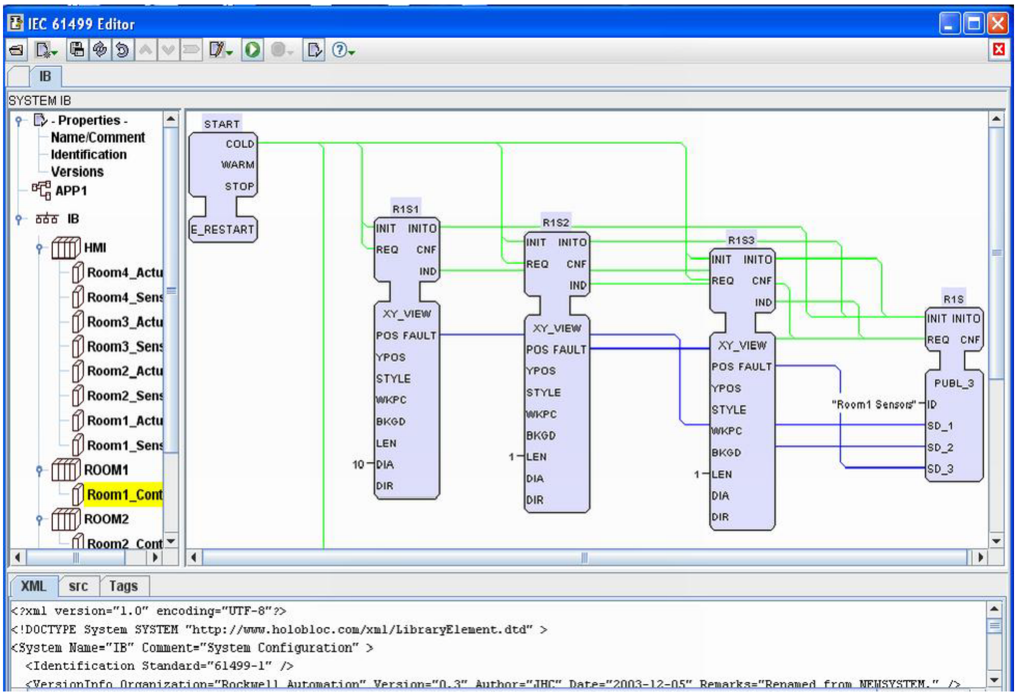

Figure 5-12: Function Block Development Kit Editor (FBDK)

5.8.2 CORFU Engineering Support System

CORFU is designed in University of Patras, Greece [60]. It is compliant with IEC 61499 function blocks and there is a design section integrated based on Unified Modeling Language (UML). CORFU is connected to IBM Rational Rose which is a software for development UML diagrams. Therefore, user will start designing with UML and CORFU will translate the UML to the corresponding function block files. The resulted design file is based on XML that can be utilized in any other software for further processing. The difference between FBDK and CORFU is that CORFU is not implemented on Java. A screenshot of CORFU is given in Figure 5.13.

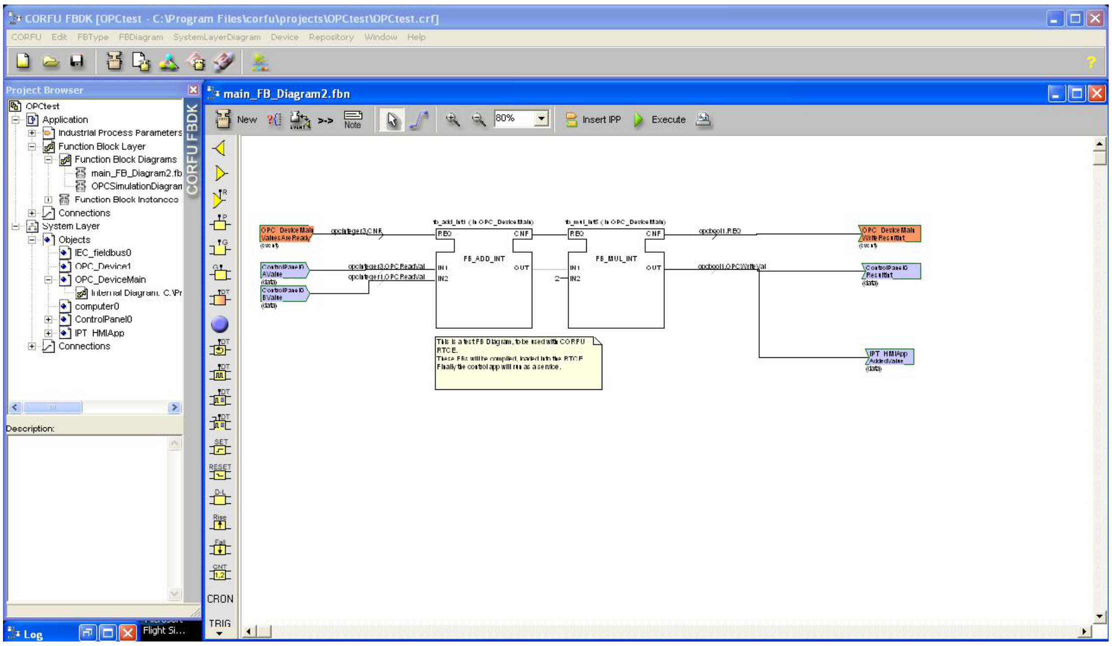

Figure 5-13: Corfu Engineering Support System

5.8.3 ISaGRAF

ISaGRAF [61] was first designed for Programmable Logic Controllers (PLCs) based on IEC 61131-3 standard. it is then extended to support IEC 61499 function block standard as well in 2005 and therefore it is the only commercial software which supports both IEC 61499 and IEC 61131-3 standard.

5.8.4 O^{3}Neida Workbench

It is an open source project developed by OOO Neida group [62] and it is based on Java. The aim is to integrate the concept of Automation Object (AO) with IEC 61499 function blocks standard. More information can be found in [62].

Chapter 6

PROPOSED METHODOLOGY FOR STRUCTURAL MODELING OF IWSAN

6.1 Introduction

Recently the globalization of manufacturing industry systems has led to increase the competition in order to respond to today's demanding market especially in medium-size companies. The global competition requires the manufacturing systems to be flexible and reconfigurable specifically in the shop floor level where mechatronic devices reside. In this chapter, a novel methodology for the development and structural modeling of industrial wireless sensors and actuators is presented in order to provide flexibility and reconfigurability to the mechatronic devices.

Due to the distributed nature of WSNs, it has a high potential to be the enabling technology for a variety of industrial monitoring and control applications such as: industrial process monitoring, fault detection, real-time data collection and location tracking. Recent technological developments have proven the potential of merging actuators into the Industrial Wireless Sensor Networks. That is, not only the sensors but also actuators can be integrated into a wireless node in order to produce actions to control various operations. The addition of the actuators can make a wireless node capable to monitor and control a mechatronic device such as an industrial robot, a programmable controller or a computer numerical controlled machine. The wireless nodes can act as a gateway to the mechatronic devices in order to collaborate in the wireless domain. In this Distributed Control System (DCS) scheme, the control will take place in the collaborative network of devices, which make their own decision locally by means of the collected real-time data.

The Industrial Wireless Sensor and Actuator Networks (IWSAN) could be essential enablers of the distributed control systems as they provide flexible, wireless real-time data collection for monitoring, diagnostics, and control of the mechatronic devices at industrial automation settings. Hence, the integration of ISWAN into the function blocks framework will enhance the re-configurability and reliability of distributed control systems

In this regard, IEC 61499 does not support capturing the system requirements and then translating for the design specification [64] [65] [66]. This motivated the researchers to develop various tools in order combine Unified Modeling Language (UML) [67] and IEC 61499 function blocks. CORFU Engineering Support System (ESS) is an engineering tool compliant with IEC 61499. It captures all the requirements by means of UML and then it is translated to function blocks according to part 2 of IEC 61499 standard. UML-FB is developed in [68] [69] for modeling and implementation of IPMCS. In [70] [71] development process of industrial automation systems is addressed. The system is based on UML and IEC 61499 standard from user requirements to implementation domain. However in all the mentioned works, the design and control architecture of industrial wireless sensor and actuator networks has not been addressed specifically on the basis of IEC 61499. In addition, several challenges exist in integrating the IWSAN into such a combined framework. The WSN technology is new and still so complicated to be implemented into the components of distributed control of manufacturing systems due to the lack of structured modeling methods and effective guidelines.

The proposed methodology is based on the implementation of the IEC 61499 standard for the distributed control of mechatronic systems. The methodology also addresses the existing problems of this standard for capturing the system requirements in the development process. For this reason, the Unified Modeling Language (UML) is used in order to overcome this problem. A case study is presented in order to demonstrate the operational aspects of the proposed methodology.

6.2 Modeling Language

The Unified Modeling Language (UML) is a graphical language which is used for general-purpose modeling within the Object-Oriented (OO) software engineering framework. UML is for visualizing and specifying the conceptual things in a system as well as describing the system's blueprints within particular stakeholders in order to implement the system's architecture. Specifically, UML is utilized for describing both structural and behavioral model of the system from high level to low level of abstraction. Structural models highlight the static organization of the things which form the system, whereas in the behavioral model, the activities and the dynamic models are expressed. In both cases, the relations among things or objects are depicted by means of special relationship symptoms. There are several UML diagrams each of which represents the system from a particular point of view not only in structural but also in behavioral. This will help to better understand the control architecture at any level of abstraction from enterprise to physical layer. In the proposed methodology, the system contains three phases, namely system requirements in which use cases emphasize what a user requires, design phase and implementation which addresses the physical realization of the system

The structural diagrams which are used in this thesis are as follows:

- **Class Diagram:** it pictures the static aspects of the system's basic building block.
- **Object Diagram:** it consists of instances of things defined in class diagram. It demonstrates the static view of the objects which collaborate without presenting messages passed between them.
- **Deployment Diagram:** demonstrates the runtime processing configuration nodes. It also shows the components which live in a node. Deployment diagram is utilized for modeling the static view of the system. It is actually a class diagram focuses on systems hardware level.
- **Component Diagram:** it shows the static implementation view of the system. The physical things which are defined in a node, are modeled by component diagram. It is actually a class diagram which focuses on the system's components.

  The behavioral diagrams which are used are as follows:
- **Use Case Diagram:** contains a set of acts, use cases and relations among them.
- **Sequence Diagram:** it demonstrates the message ordering while time passes.

- **Activity Diagram:** it presents the activities which are performed in the system over the time. It also depicts the control which flows from one activity to another.

6.3 IWSAN based on IEC 61499 Standard

Industrial wireless sensor and actuator networks can be utilized in distributed sensing and actuating systems. Due to its distributed nature, the IWSAN has a high potential to be the enabling technology of distributed sensing and control for various applications in distributed control of manufacturing systems based on the IEC 61499 Function Blocks. According to what have discussed in the previous section, a sensor and actuator node contains electronic components which are interconnected by means of a data bus and each of which has control over its own operation. Thus, in the proposed methodology, each node and its electronic components corresponds to a device and resources, respectively (Figure 6.1).

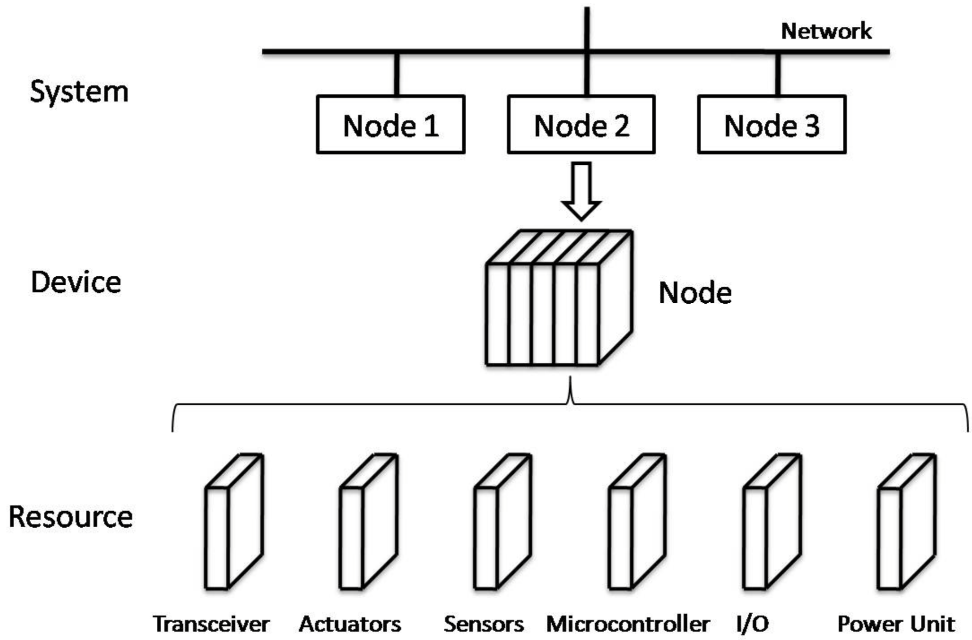

Figure 6-1: IWSAN modeled based on IEC 61499 standard

A system is a network of several nodes (Devices) which are connected by means of a wireless communication network. This network enables the node for sharing data among other nodes and it is also used for conveying data to the gateway or base station. Electronic components are the node’s resources which are connected in a distributed fashion. An application can utilize one or more resources in order to perform a particular task. Specifically, data are read from the physical entities by process interface and it is, then, processed in the corresponding resources. Data exchange among resources is performed by communication interface.

6.4 Overall Methodology

The proposed methodology illustrates the development process from system requirements to implementation and on the other hand, the four layers which describe the system level from presentation to sensing and actuating. Figure 6.2 represents the methodology. In the system requirements phase, all the system specifications are captured from the user for all four layers. These requirements are, then, utilized for design and development phase. In this phase, the system will be designed in both structural as well as behavioral domain. These are corresponding to static and dynamic behavior of the system. Ultimately, the designed structure and behavior is implemented from presentation to sensing and actuating. The details of the proposed control architecture will be explained using an example of manufacturing process which is composed of several mechatronic devices. This system is located in the Flexible Manufacturing System (FMS) Laboratory of Eastern Mediterranean University (Figure 6.3). The aim is to design wireless sensor and actuator networks for distributed monitoring and control of the mechatronic-devices in this setting.

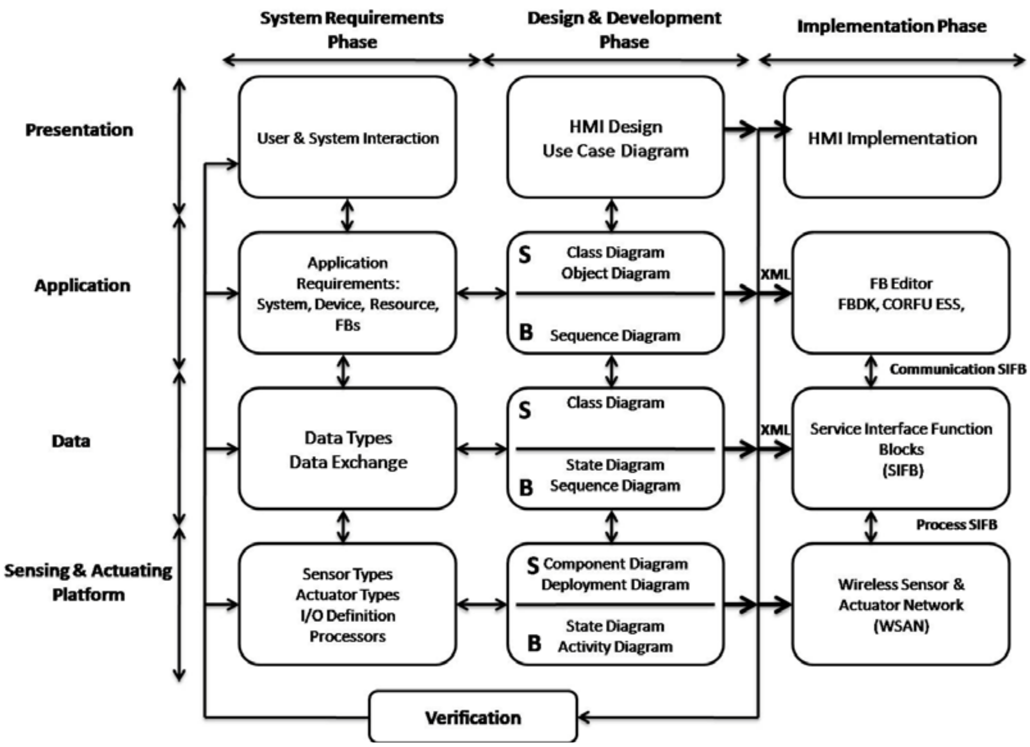

Figure 6-2: Proposed Methodology for Modeling IWSAN

Figure 6-3: Flexible Manufacturing System (FMS) Laboratory of Eastern Mediterranean University

The description of the prototype system is as follows:

“The system consists of three workstations, namely Automated Storage and Retrieval System (ASRS), CNC Machine station and an assembly station each of which contains several mechatronic devices. The system operates as a typical flexible manufacturing system, where the work-pieces travel among these workstations in an order determined by a production control unit. The flow of work-pieces is among workstations and is performed with a conveyor system. Initially, when a work order is released, a work piece is placed on the conveyor by means of a robot. It is, then, moved to CNC machining workstation which comprises of a sliding robot and a CNC machine for further operations. The work piece is conveyed to assembling workstation afterwards. Width and height testing, assembling, gluing and filling are performed in this part and eventually the work piece will be placed in ASRS."

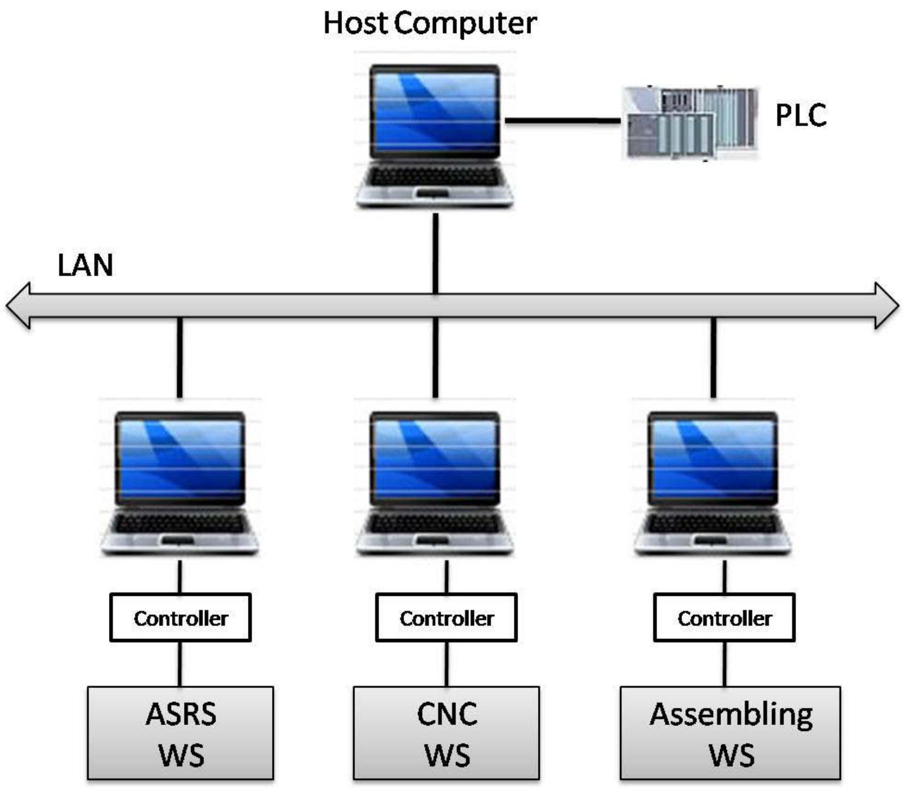

Figure 6-4: Flexible Manufacturing System Laboratory system architecture

Figure 6.4 depicts the system architecture which is currently implemented in the FMS Lab. There are three workstations each of which connected to a controller. There are also three PCs which, in one hand, manage the controllers and, on the other hand, connect to a Local Area Network (LAN) in order to establish communication among the other workstations. Host computer controls all the commands which are sent to each PC. The conveyor is controlled by PLC which is connected to Host computer. The problems with the current architecture can be addressed as follows:

1. The system is controlled by a central processing unit which is located in the host computer. All the control signals and commands are issued by this unit.

2. The workstations have restricted capability of controlling over their own operation.

3. The system makes use of wired sensors and actuators.

4. The system lacks the real-time reconfiguration at any level of abstraction such as device, resource.

6.5 System Requirement Phase

In this phase, the requirements of the system in different layers are captured at a very low level of details. Thus, the overall requirements and the scheme regardless of any structural and behavioral diagrams are expressed.

The ASRS workstation described in the system contains a 6x6 storage totally 36 cells considered for storage and retrieval each of which contains a sensor in order to report the occupancy of each cell. The status of each cell needs to be monitored by user. There is a robot (actuator) which has the ability to place or take a work piece from conveyor into a specific cell. Also there is a sensor placed on the conveyor near the ASRS in order to detect the work piece to be taken. Therefore there will be a device node which contains 37 sensors and an actuator.

Similarly, for the CNC machining workstation, there is a CNC machine which is equipped with a sensor in order to report to the status the machine. A sliding robot which moves along a line on a path is responsible for taking the part from conveyor and placing it into the CNC machine and vice-versa. The robot is also equipped with a sensor for status monitoring. Two sensors are placed on the robot path in order to detect the position of the robot. Therefore, there are totally five sensors and two actuators in this workstation.

In assembling workstation, there are testing, gluing and a machine which fills the work piece with some balls and each of which equipped with a sensor. The robot moves the part from conveyor and based on the testing results, it will be placed in either trash box or for further assembling operation. In this work station, there are totally five sensors and four actuators.

In most cases the sensors are subject to detect work pieces and therefore a digital signal which has two values (True or False) is sent to the processor. Thus, the binary data type will be used on the data tire. For testing machine in assembly workstation, a real or integer number is sent to processor. Based on the number of sensors and actuators the devices and resources will be determined.

6.6 Design and Development Phase

6.6.1 Presentation Layer

The system requirements which are described will be used for design and development phase. The outcome of this phase goes through a verification process in order to check the logical behavior whether it is consistent. The successful design scheme will be overtaken to the implementation phase.

Design phase begins with Human Machine Interface (HMI) with the help of UML use case diagrams. UML use case diagrams describe the system at high level of abstraction that hide a lot of details. The requirements which are captured in the previous phase are employed to design the presentation layer. Figure 6.5 presents the overall system interaction with operator.

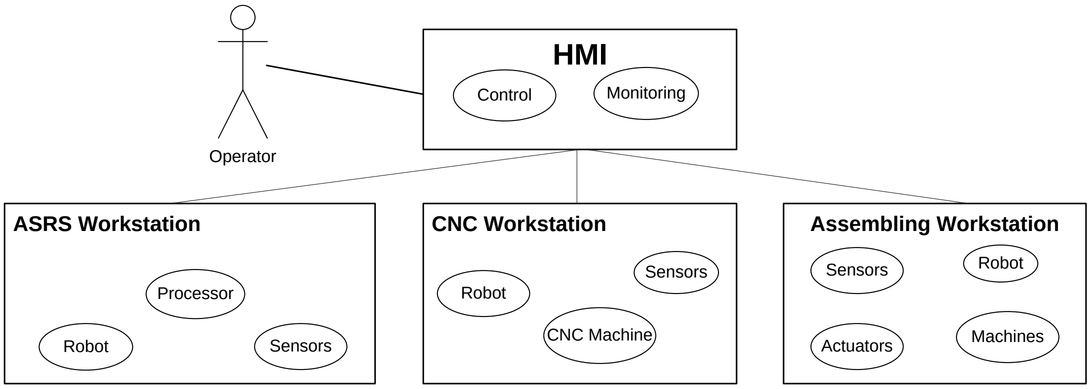

Figure 6-5: use case diagram for user and system interaction

The operator interacts with system via HMI. Control and monitoring use cases are located in HMI which interact to workstations.

6.6.2 Application Layer

In the application layer, class and object diagrams are utilized in order to model the structural part. The class diagram of the IEC 61499 standard is shown in Figure 6.6. This diagram illustrates the hierarchical model and it is fully compliant with IEC 61499-2, part two of the standard which addresses the software tools requirements.

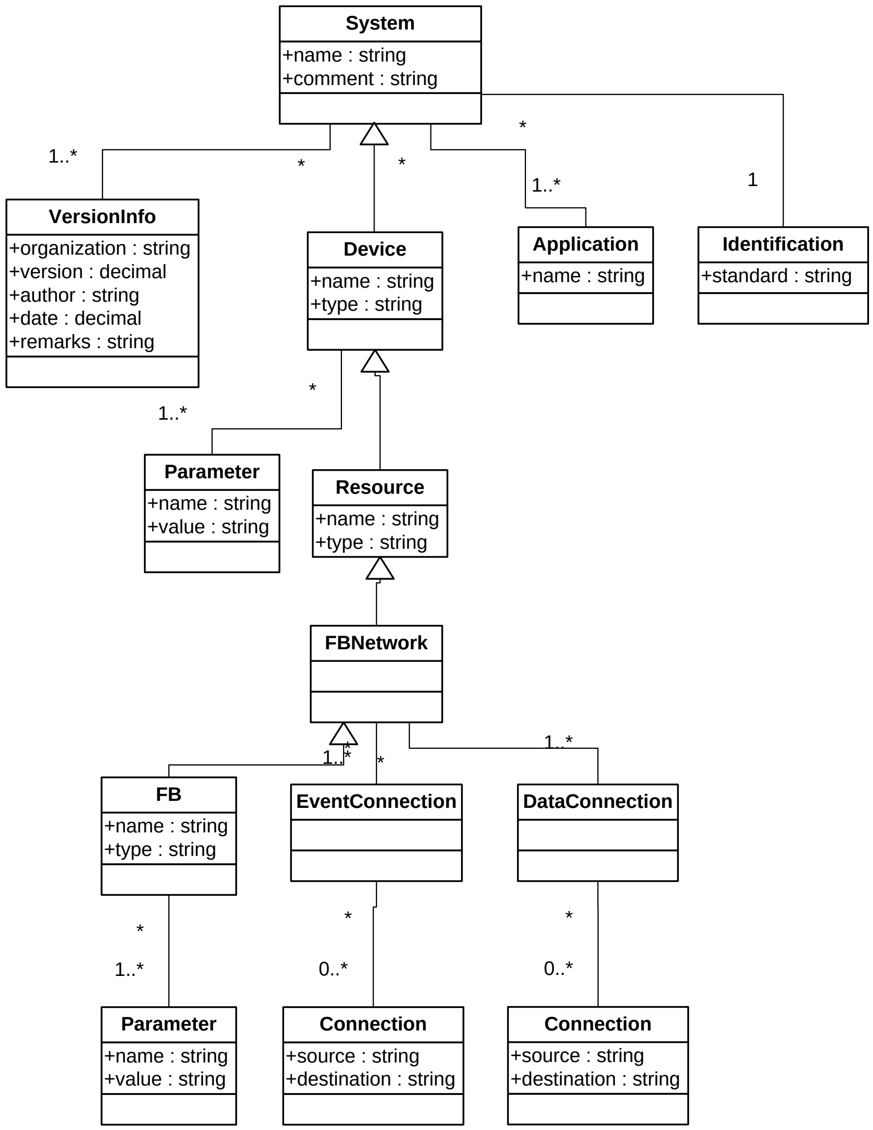

Figure 6-6: UML Class diagram for hierarchical model of IEC 61499 standard

These classes are the system's building blocks and it is the reference class diagram which will be used for developing object diagrams for different workstations. Each class has a name and attribute. The attributes are made public and they are accessible through all the system.

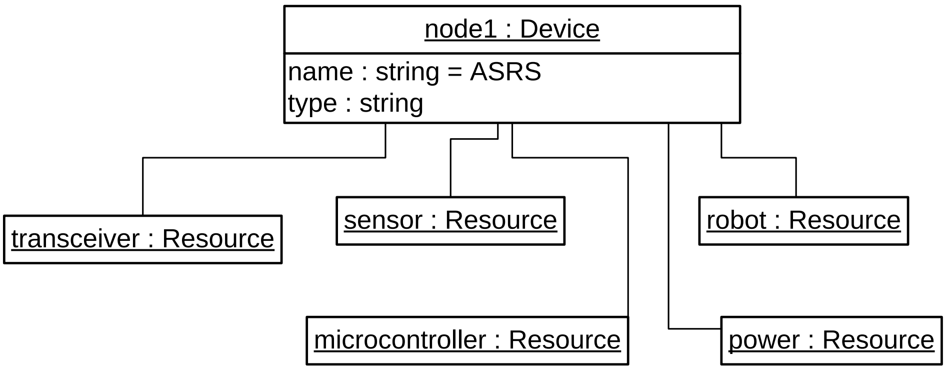

Figure 6-7: UML object diagram for ASRS

Figure 6.7 illustrates the object diagram which is developed ASRS based on class diagram. The node is an instance of the device and each sensor and actuator is an instance of the resource. Microcontroller, transceiver and power system are also instances of the resource. Similarly, for assembling workstation, sensors, actuators and other electronic components are instances of the resource (Figure 6.8).

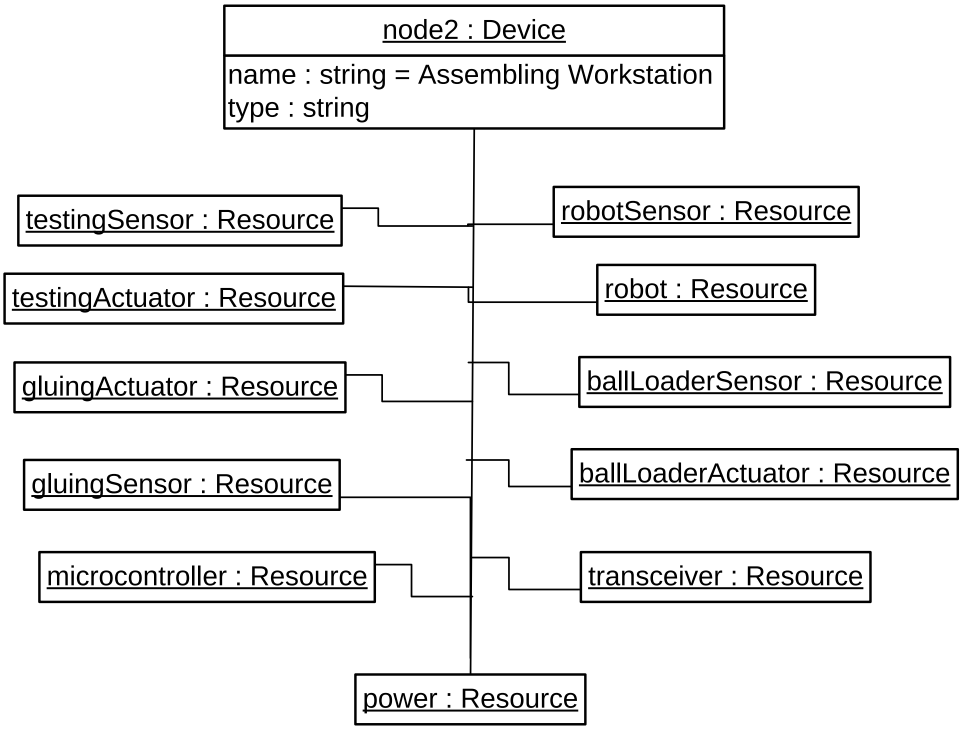

Figure 6-8: Node platform using UML object diagram for assembling workstation

Figure 6.9 presents the node platform for CNC machining workstation.

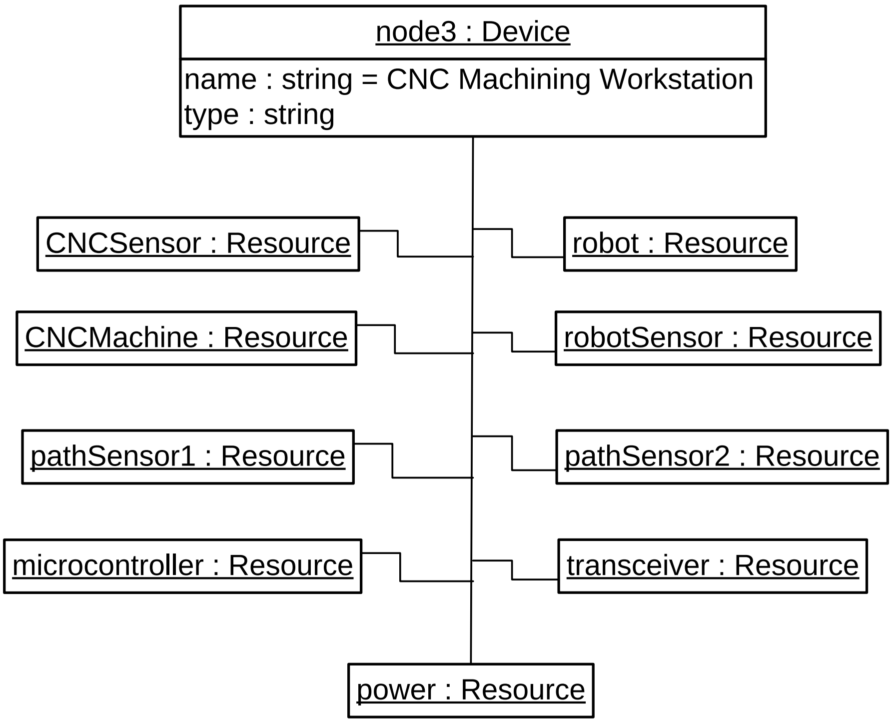

Figure 6-9: Node platform using UML object diagram for CNC machining workstation

Class and object diagrams demonstrate the static aspects of the system building blocks. The dynamic aspect of the application layer is presented by UML sequence diagram. A node contains several resources which represent the electronic components and they are connected by means of a data bus. All the nodes in a system form a network and this network is connected to the HMI layer. By receiving a message from the HMI layer in order to perform a task, the message will be broadcasted to the network and the corresponding node will be informed. The transceiver resource receives the message and based on its content, the proceeding operations will be performed. For instance, Figure 6.10 illustrates the time ordering of the messages for ASRS node's resources in order to execute the requested task from HMI layer.

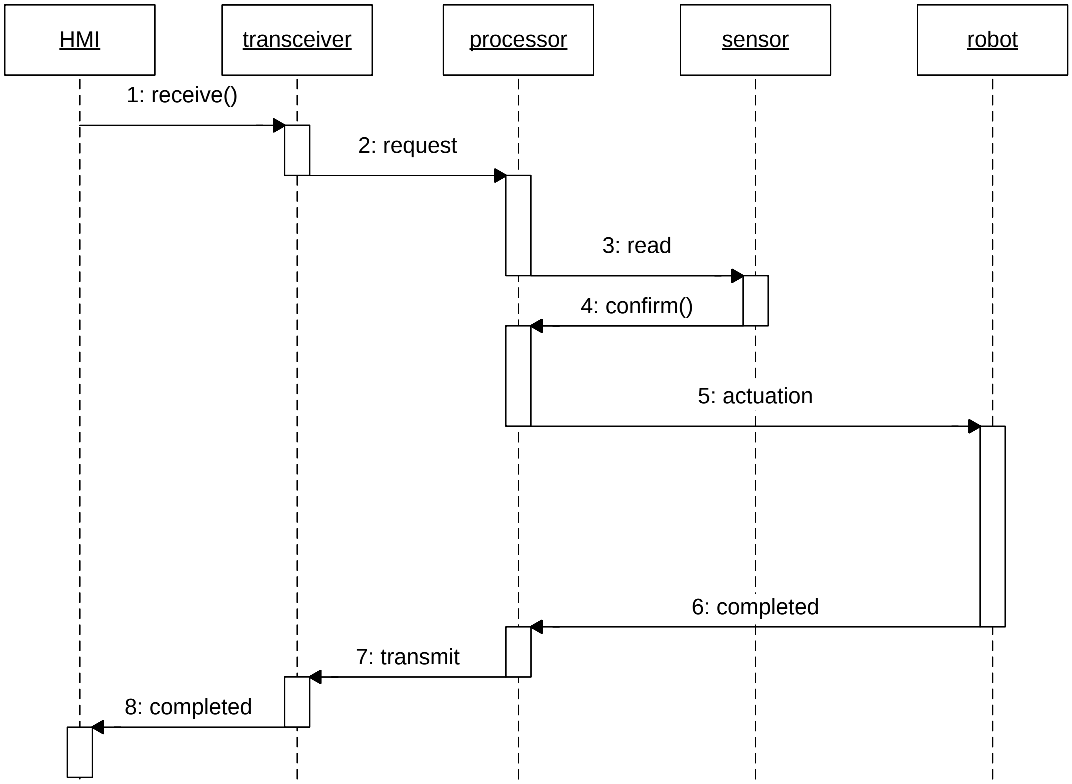

Figure 6-10: UML sequence diagram for ASRS node platform

All the electronic components in Figure 6-10 are objects which are instances of a resource. Several operations based on the scenario can be executed in the assembly workstation as well as CNC machining. Therefore, a simple operation for assembly work station is presented in Figure 6.11. Any other operations which are received from the HMI layer can be designed and executed in a similar manner.

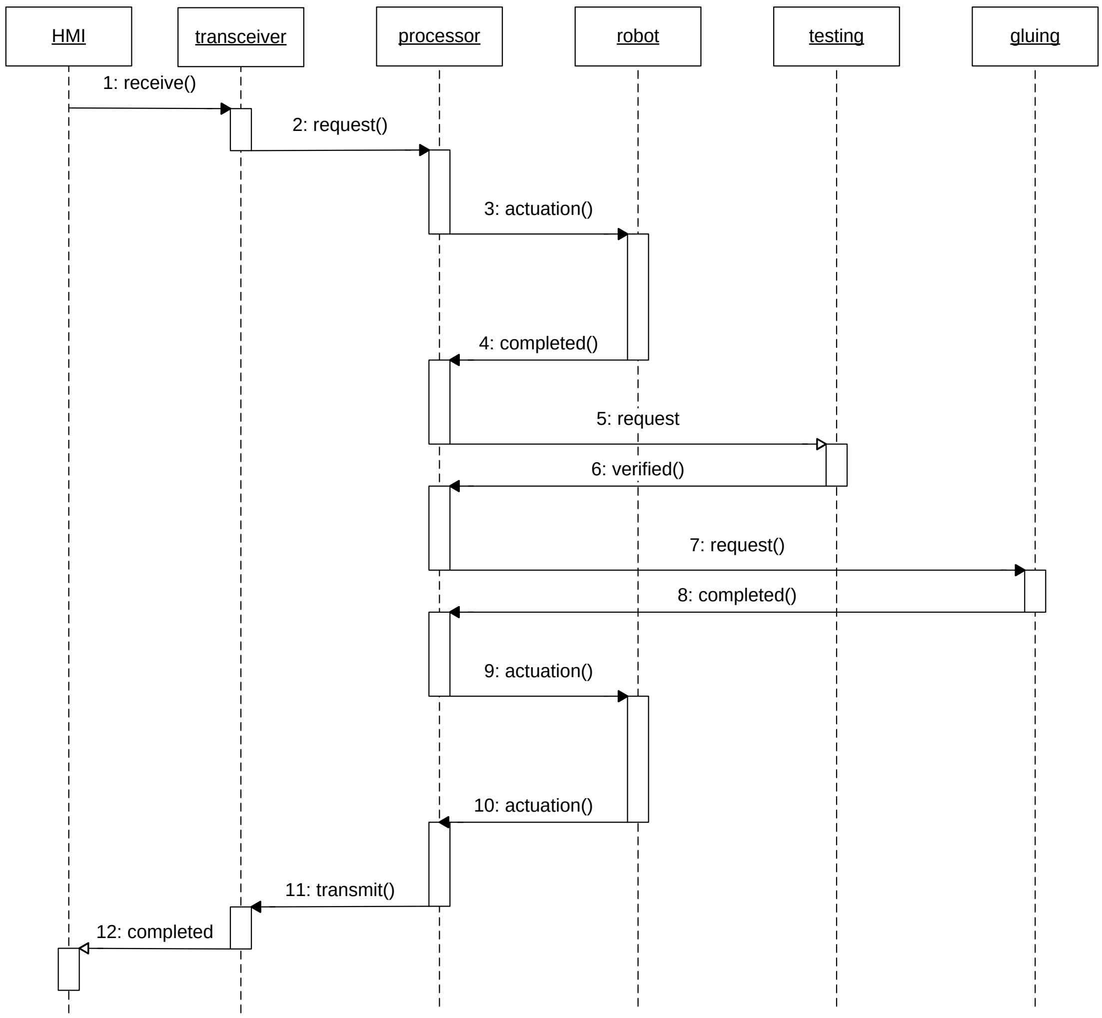

Figure 6-11: UML sequence diagram for assembling node platform

6.6.3 Data Layer

This layer also contains structural as well as behavioral part. In the structural part, UML class diagram presents the class hierarchy of function block types. Three function blocks are stated in IEC 61499 standard, namely Basic, Composite and Service Interface Function Blocks (SIFB). Figure 6.12 represents the UML class diagram for different types of function blocks. In the data layer, SIFBs are utilized for data exchange. Typically dealing with data communication and implementation requires hardware understanding and this is the reason that it is not developed in the application layer. Specifically, these are function blocks that provide services to the resources for data movements. Process Interface Function Blocks (PIFB) collects data from the physical entities whereas Communication Interface Function Block (CIFB) is responsible for data movements and management between resources or devices. In the proposed methodology, node to node communication in a system (which is device communication in IEC 61499) is performed by special CIFB called PUBLISH and SUBSCRIBE [54]. In this regard, each node has a dedicated IP address in the network. Communication among the electronic resources in a device will be performed by means of CIFBs which are dedicated for local communications.

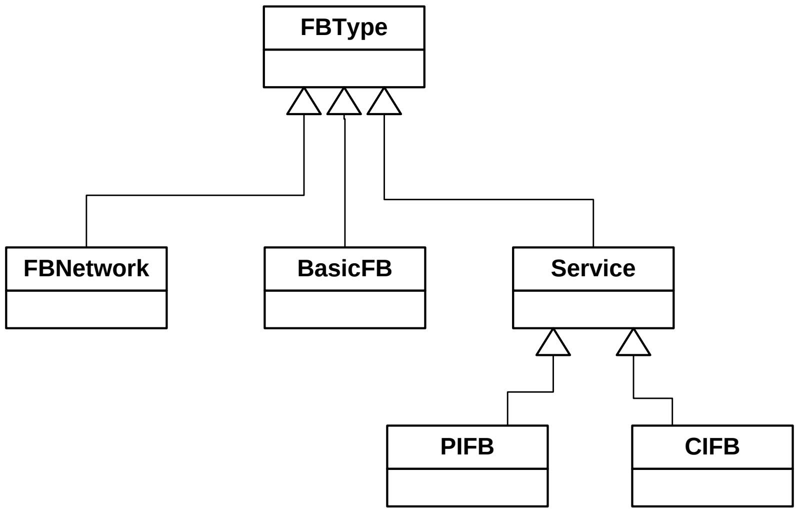

Figure 6-12: UML Class Diagram for different types of function blocks

For instance consider the ASRS or other workstations node platform which contain several electronic resources. In the resources in a node, there will be an instance of CIFB for local communication and one PUBLISH and SUBSCRIBE for node to node communication.

The dynamic behavioral of each SIFB is specified by service primitives which are identified in the part one of IEC 61499. An example of a SIFB is shown in Figure 6.13.

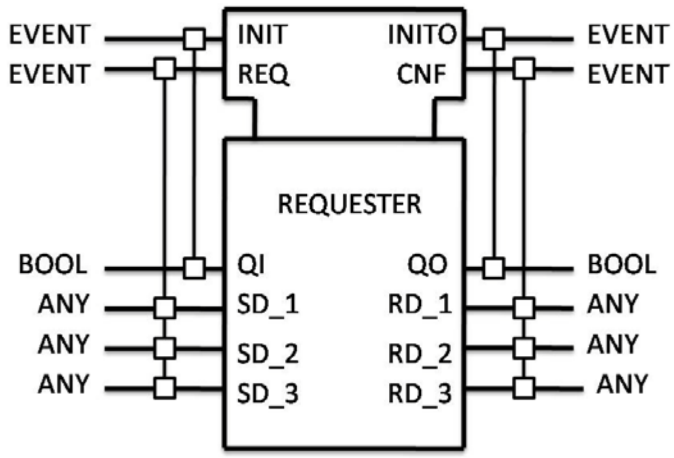

Figure 6-13: An example of Service Interface Function Blocks

Upon receiving event at INIT along with data input QI, the initialization will be done and the INITO event with QO data output will be issued. The REQ event input is associated with SD_1 to SD_3 data inputs which mean that, when an event is received at REQ, the associated data will be read and prior to completion of the service, the corresponding event which CNF and its associated data will be available. This dynamic behavior is specified by service primitives which can be illustrated using UML sequence diagram (Figure 6.14).

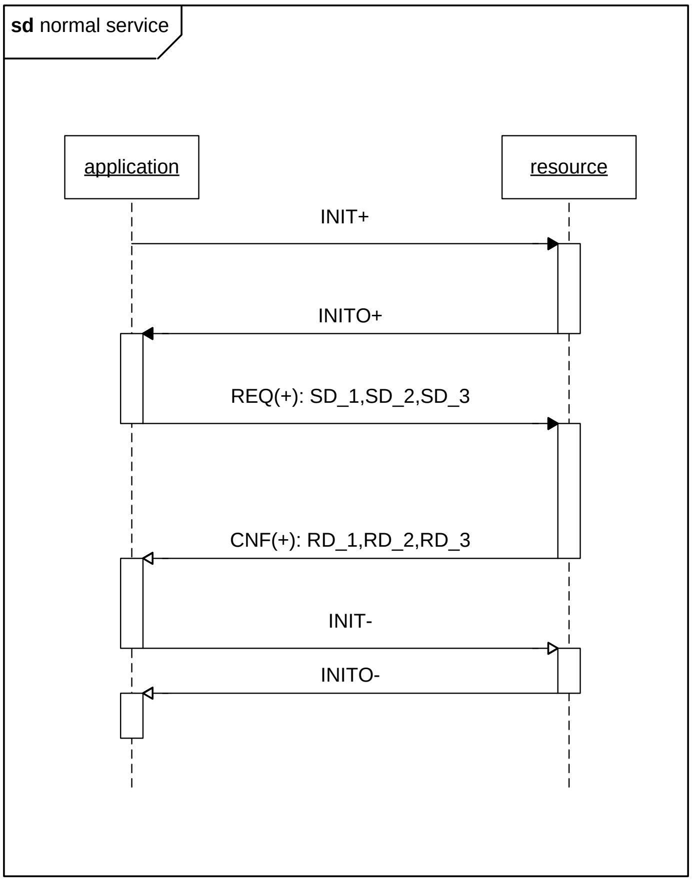

Figure 6-14: UML Sequence diagram of normal service establishment

In the example of a SIFB, the application requests service from SIFB by sending an event to the event input. Initialization is performed while the value of *QI=true*. This is indicated by plus sign near each event inputs i.e. INIT+ and INITO+. The service execution starts upon receiving signal to the REQ along with *QI=true* which will be REQ+. The REQ event is associated with the data inputs, that is, SD_1 to SD_3. Prior to completion of the service, the CFN+ event will be issued with its associated data which are RD_1 to RD_3. Receiving event in INIT with *QI=false* will terminate the service of the SIFB.

6.6.4 Sensing and Actuating Platform

This is the physical layer and similar to the previous layers, it contains structural as well as behavioral part. In the structural part, both UML component and deployment diagrams are utilized in order to specify the static implementation view of the WSAN node and system. Component diagram specifies the components which form a node in deployment diagram. In component diagram, electronic resources are modeled by components in UML. For instance one or more sensors and actuators with processing, power and transceiver units form a node. This is represented in Figure 6.15.

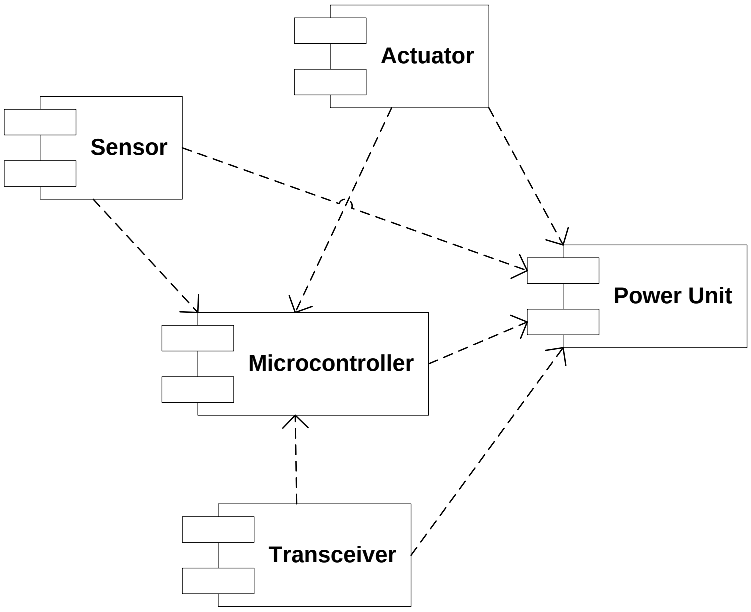

Figure 6-15: UML component diagram for WSAN node platform

In this figure, there are five components (resources) each of which has dependencies relationship with one another. The power unit component is required by all the other components. Sensors, actuators and transceiver components are dependent on microcontroller component. These components are all instances of a resource in IEC 61499 function blocks. In fact, this diagram illustrates the resources in a device and their dependencies relation. For instance, for ASRS work station, the components diagram is the same as diagram in Figure 6.15 and thus more sensor components will be shown rather than one and the actuator component will be the ASRS's robot. Likewise, for the other workstations, the diagrams are similar and simply the names of actuators are different. In the assembly workstation, the actuator components are robot, ball loader, gluing and testing. Due to the similarities in the diagrams for workstations, they are omitted. UML component diagram specifies the static implementation view of the node platform. However, deployment diagram identifies the implementation view of the system. Figure 6.16 presents the deployment diagram which addresses the static realization of the system. In this figure, each WSAN node is represented by its corresponding representation in UML deployment diagram. A node consists of several components which are the instances of the component diagram shown in Figure 6.15. The nodes are communicated via a wireless communication link. Typically, low power and small radio standards such as ZigBee (IEEE 802.15.4) [72] and wireless HART [73] are utilized. The latter is newly adopted in industrial-process measurement and control and it is highly reliable, secure and effective in power management. The result of sensing and actuating is shared among the other nodes and it is, then, forwarded to the base station node. Base station is a node itself and has capabilities of processing as well as transmitting and receiving. Base station can be connected to either a server or directly to the internet for remote controlling via TCP/IP protocol. The component diagram models each individual node while the deployment diagram demonstrates the whole system. In both cases the static models are presented.

The dynamic behavior of this layer is illustrated by state machine and activity diagram. State machine diagram models the behavior of individual nodes. It specifies the sequences of the states that a node goes through during its life time. Figure 6.17 presents the life cycle of a node. It can be extended for any workstation's node such ASRS, CNC machining and assembling. Typically, a node has four states, namely initialization, idle, sleep and active. Upon node's startup, the initialization is performed. This includes microcontroller's startup as well as sensors and actuators initialization. For instance, each robot in all workstation requires going to its home position. Network identification is essential in this state. That is, identifying all the nodes which are in the coverage area and broadcasting its ID to neighborhood nodes. Prior to completion of initialization, the node is ready and remains in the idle state. If no event is received after a specific time, it goes to sleep mode in order to save energy and upon receiving any event it will go back to idle and then active mode. An event can be either a message which receives from the other nodes or sensor readings. In both cases, the event is processed and the appropriate signal is sent to the actuators such as robot, gluing, testing or CNC machine. The result of the processing may be broadcasted to the network and the node departs from active state and remains in idle afterwards. This is the life cycle of a WSAN node in the system.

The activity of the whole system by UML activity diagram is depicted in Figure 6.18. This figure represents the scenario which is demonstrated in the previous sections.

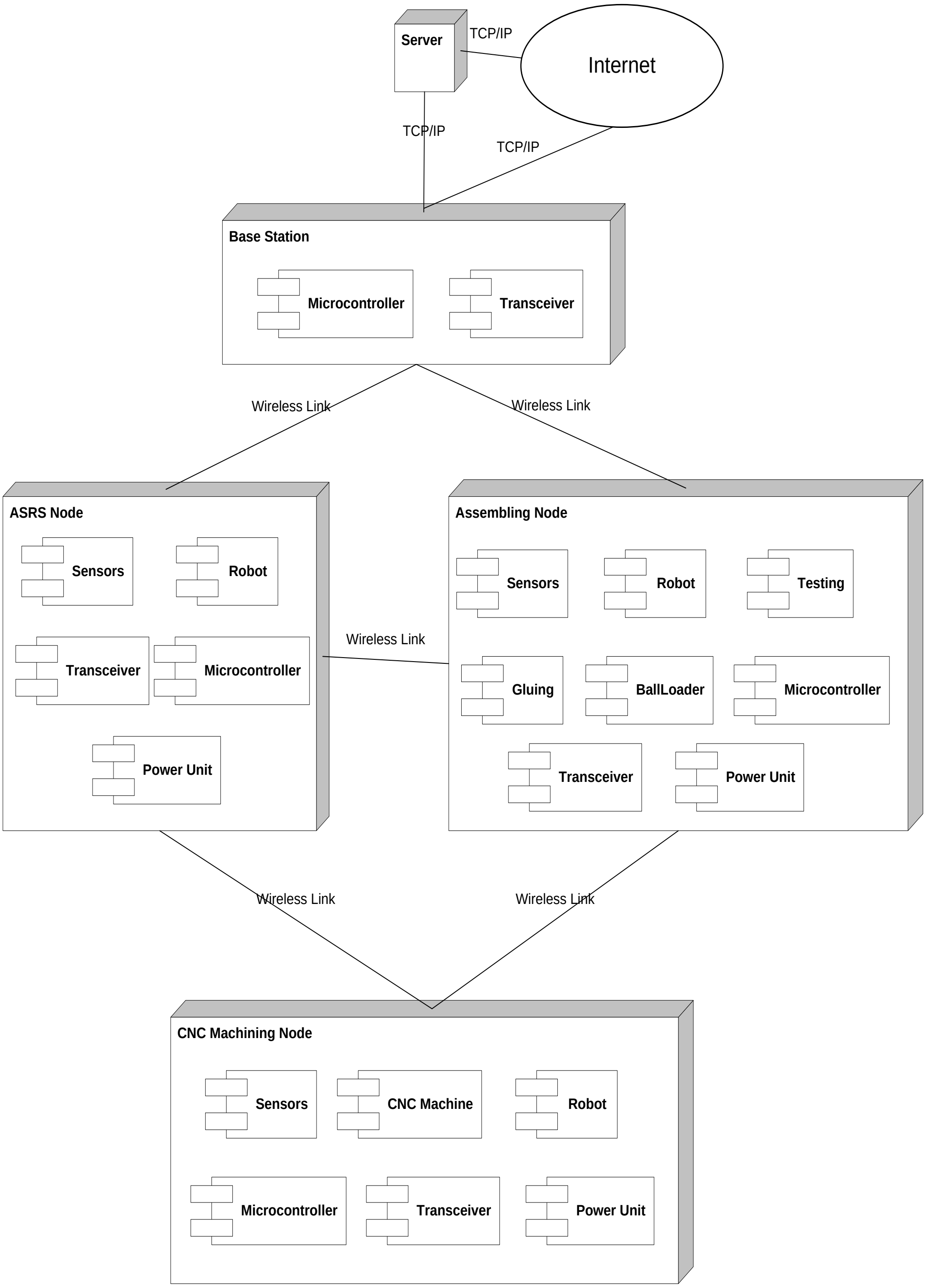

Figure 6-16: UML Deployment diagram for static realization of the system

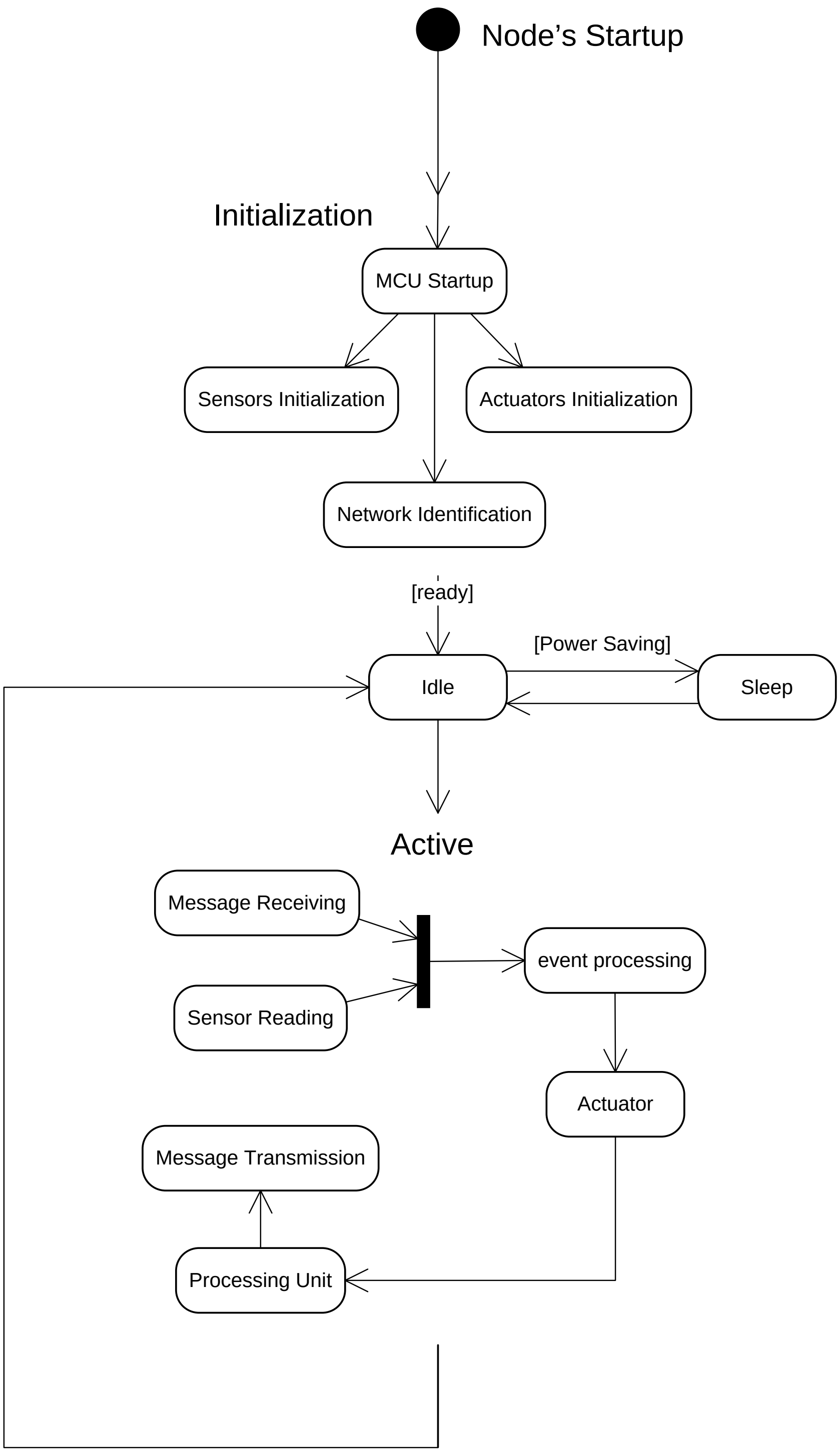

Figure 6-17: UML State machine diagram for WSAN node’s life cycle

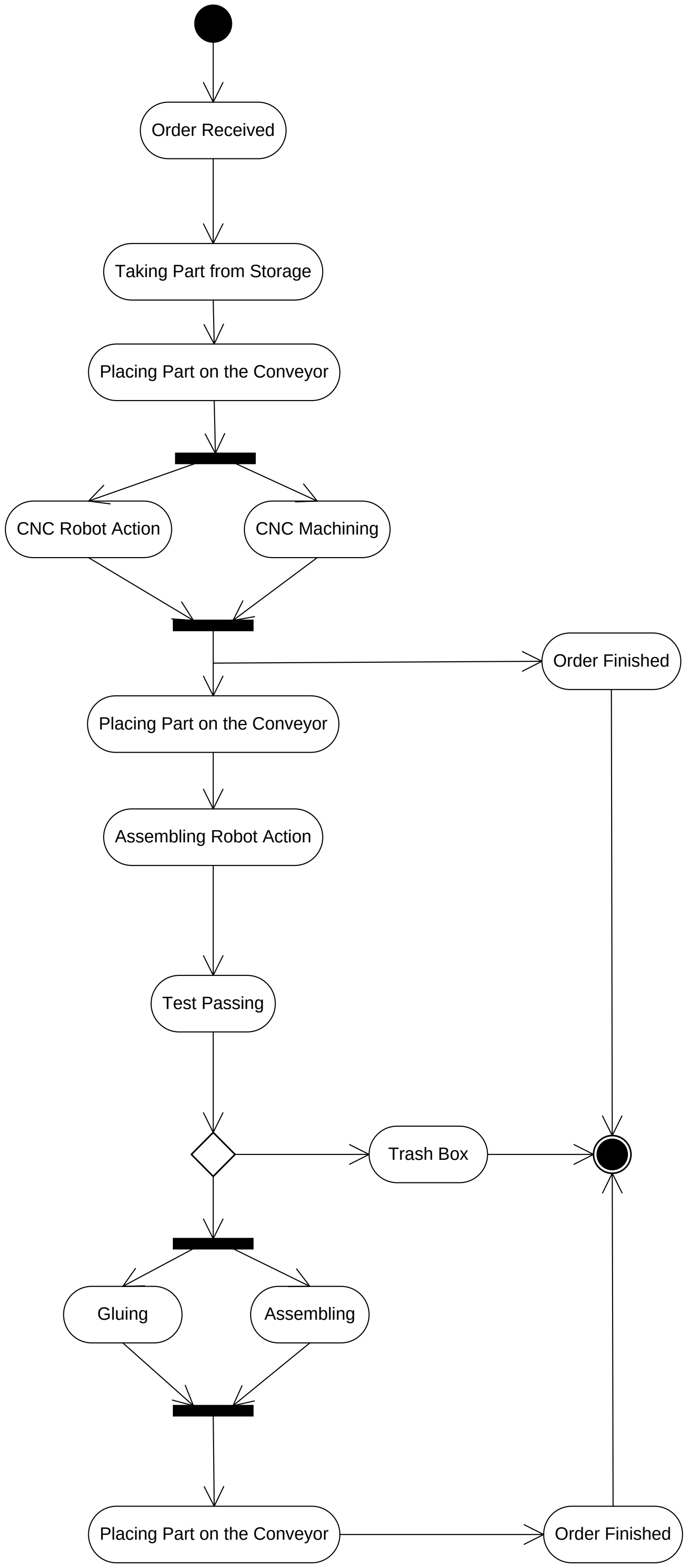

Figure 6-18: UML Activity diagram for the proposed system

6.7 Verification Phase

The designed system is required to be verified before proceeding to implementation phase. This can be performed in both system requirements as well as design and development phase and also in each level of abstraction. The requirements are captured and based on them the expert designs and models the system by means of UML diagrams. Consistency check must be performed for all the UML diagrams which are utilized in the design and development phase. For instance, in the application layer in design phase, inconsistencies between UML class, object and sequence diagrams may result to misplace the devices and resources in the system.

The proposed methodology is based on IEC 61499 standard and all the developed UML diagrams in the design phase can be converted to the XML format files. XML is the encoding mechanism for the exchange of data between systems and device models. The XML files which are the result of the design phase must be compatible to the part two of IEC 61499 standard which defines the DTDs for the exchange of IEC 61499 library elements. There are many XML and DTD validators available which can be used for verifying the XML's elements [74].

6.8 Implementation Phase

6.8.1 Presentation Layer

For HMI implementation several object-oriented languages are available. For instance java can be used for developing the graphical representation as well as the real time control by means of java run time environment (JRE). The reason for using java is that there are some processors which have hardware implementation of run time environment.

6.8.2 Application Layer

Tools are available in order to implementing the XML files. For instance, Function Block Development Kit (FBDK) is built up by Holobloc for development of systems based on the IEC 61499 standard. It is a java based tool which contains Function Block Run-time (FBRT) environment for real time execution. By converting UML diagrams in this layer, the XML files can be imported easily at any level of abstractions such as, system, device, resource and function blocks. CORFU ESS can also be utilized from the design phase to implementation. It has the ability to convert the UML diagrams to the corresponding function block type.

6.8.3 Data Layer

The SIFBs are also implemented in FBDK and the XML files which presents the sequence and the semantic of the communication can be easily imported in FBDK or any FB editor.

6.8.4 Sensing and Actuating Platform

TinyOS [75] is an open source operating system (OS) which is developed for low power wireless devices such as sensor networks. It is a component-based and event-driven OS and all the codes are written in NesC which is Network Embedded Systems C. TinyOS is very appropriate for the physical programming and implementation of the proposed methodology. Component-based property of the TinyOS is used for the electronic components such as sensors, actuators, transceiver and processing unit. An application in TinyOS, which is written in NesC, contains one or more components that are connected by interfaces. According to [75], a component provides and uses interfaces. The provided and used interfaces are utilized for presenting the functionality of its specification and the functionality which is used to perform the intended task in implementation, respectively. Thus, each electronic resource in WSAN is a component in NesC. UML component diagram which is developed in design phase is used to model the components and interfaces in NesC.

As mentioned before, TinyOS is an event-driven OS and due to the naturally event-driven structure of IEC 61499, TinyOS is very appropriate in order to implement the function blocks. For instance, consider the node platform in Figure 6.19.

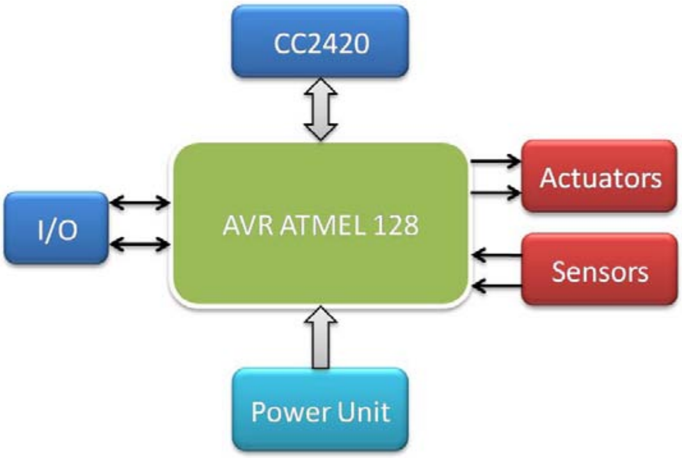

Figure 6-19: A Typical Sensor Node Platform

In this node, the microcontroller unit is based on Atmel AVR128 [76] and the transceiver resource is CC2420 from Texas Instruments [77]. Depending on the application, the sensors and actuators are chosen. All the components regarding this node platform is provided in the TinyOS library and the user may employ these components in the NesC code.

6.9 Performance Evaluation for Case Study

In this section, the performance of the proposed distributed methodology is evaluated and compared with the current centralized system. The system contains sensors and actuators where n_s denotes the number of sensors and n_a presents the number of actuators. S_{A_i} indicates the effected sensors by A_i actuator and A_{S_j} is the actuators which are in charge of S_j sensor [78]. For instance, in ASRS device, there are one robot and 37 sensors, therefore one actuator will be responsible for all the sensors, that is, $S_{ASRS}^{A_1} = 37$ and $A_{ASRS}^{S_j} = 1$, where A_1 is the ASRS robot and S_j are the sensors and $j = 1, \ldots, 37$. The same notation will be applied for the CNC and Assembly node devices.

Figure 6.20 presents the time scheduling of the centralized system which is currently operating. This is for step m which covers a product life cycle taking raw material from ASRS and placing the final product back in the ASRS. PLC as a central controller along with host computer manage the tasks and time scheduling. Therefore, a product requires T amount of time to be completed. In this period of time, each device expends a specific amount of time to operate and the rest will be idle. This is the main problem of the PLC-based centralized control systems where the tasks are performed sequentially rather than operating concurrently. As a result:

$$T_C = t_{ASRS}^C + t_{CNC}^C + t_{Assembly}^C \quad (6.1)$$

where C denotes centralized in all the terms.

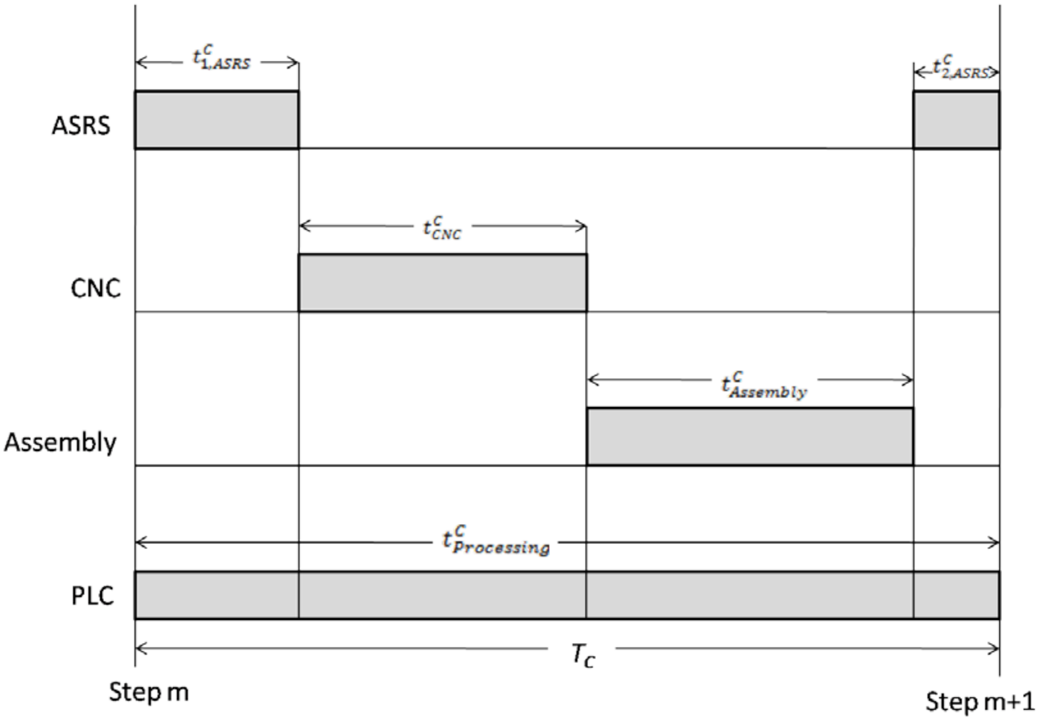

Figure 6-20: Centralized Time scheduling for a product life cycle

In the proposed distributed scheme, for every device a local processor is dedicated and they are deployed in a distributed fashion. For instance, the time scheduling for the ASRS node device is shown in Figure 6.21.

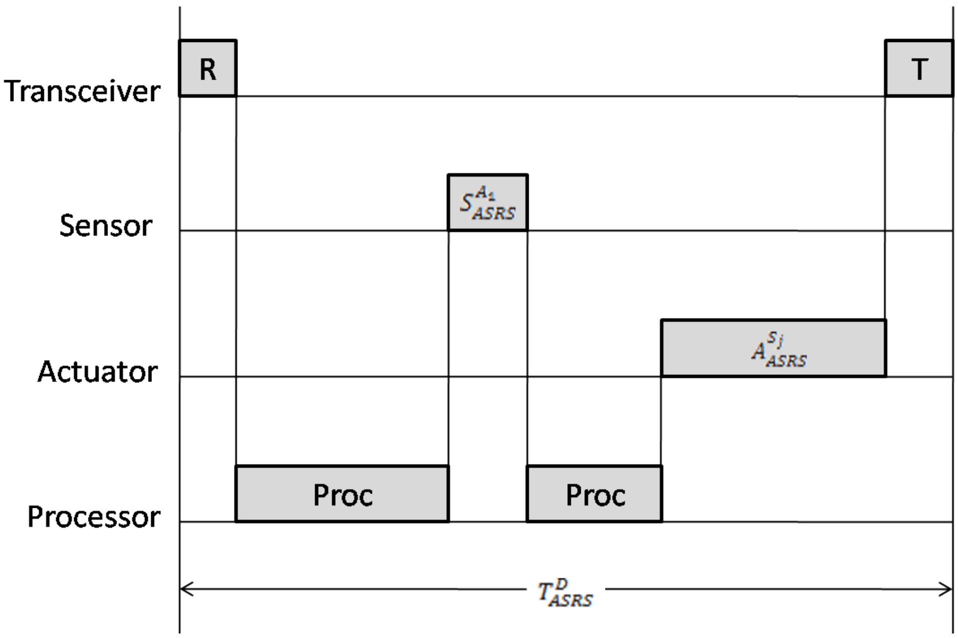

Figure 6-21: Time scheduling for ASRS node device

In this Figure, ASRS starts by receiving a message from the network by Transceiver and it is denoted by “R” and the result is also broadcasted to the network (“T” in the Figure). $S_{ASRS}^{A_1}$ presents the number of sensors to be read for a specific task. There is one actuator for ASRS node device and hence $A_{ASRS}^{S_j} = 1$. Figure 6.21 is compliant with the UML sequence diagram obtained in the previous sections (Figure 13). Therefore, same diagrams can be achieved for other node devices based on their UML sequence diagram. Therefore the total time for a product to be completed is:

$$T_D = t_{ASRS}^D + t_{CNC}^D + t_{Assembly}^D \tag{6.2}$$

Equation (6.1) and (6.2) present the time which both centralized and distributed systems expended in order to complete a product. However, the rate of completing a product in centralized system is every T_C and in distributed scheme is every $t_{Assemb\ ly}^D$ due to its concurrent nature. It is obvious that $t_{Assembly}^D < T_C$ and hence in distributed system the productivity is more than the centralized one, although both T_C and T_D may be equal. This will be proven by the simulation results as well.

In order to measure the performance parameters of a node device such as arrival distance, throughput rate, waiting and service time, Petri net [79] as a mathematical and graphical tool has been used. Petri net provides an environment to design and analyze the discrete event systems. The reason for choosing Petri net is that both wireless sensors and IEC 61499 function blocks are event-based systems, that is, upon occurring an event, one or more algorithms will be invoked and also event occurrences are completely discrete in time.

Petri net consists of a finite set of places (P), finite set of transitions (T), set of arcs (A) from places to transitions and from transitions to places and the weight function. Using the UML sequence diagrams developed for each node platform in the proposed system, the corresponding Petri net graph can be obtained as Figure 6.22 shows, for instance.

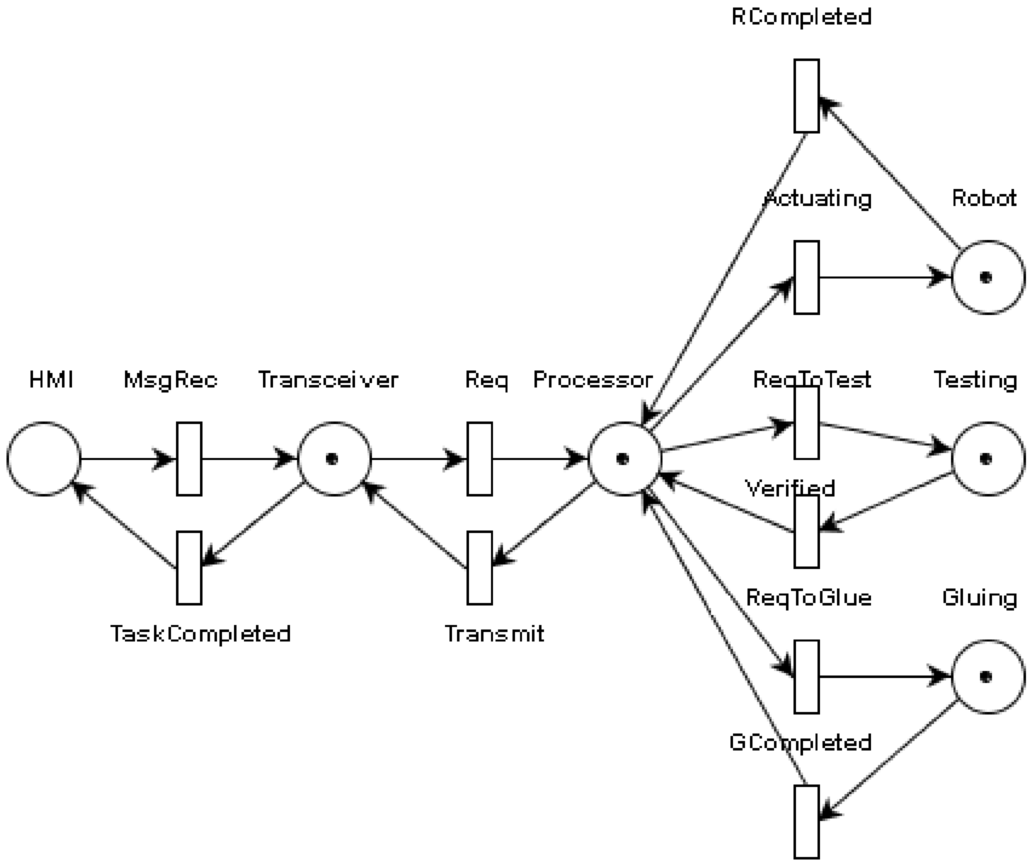

Figure 6-22: Petri net graph for assembling node platform

This figure is compliant with Figure 6.11 which shows the UML sequence diagram for the assembling node platform and presents all the steps. The circles present the places and the rectangular indicate the transitions from one place to another and present the events in the system. Tokens are the dots in every place and present the number of resources in a node platform. Similarly, for other node platforms (ASRS and CNC), the corresponding Petri net graph can be obtained based on the UML sequence diagrams.

MATLAB Petri net toolbox [80] has been used in order to simulate and analyze the Petri net graphs which developed for both centralized control (CC) and the proposed distributed control (DC) system. The systems have been simulated for 100 events and 400 units of time and the statistical results are presented in Table 6.1.

Table 6-1: Statistical Results for both CC and DC systems

Node Platforms	Arrival Distance		Throughput Rate		Waiting Time (unit)		Service Time (unit)	
	CC	DC	CC	DC	CC	DC	CC	DC
ASRS	30.71	3.06	0.032	1.4	30.71	0.02	6.24	5.63
CNC	33.27	8.20	0.030	0.52	33.27	0.01	15.24	13.97
ASSEMBLY	33.27	4.94	0.031	2.91	65.99	0.03	6.01	5.56

Four parameters have been considered to discuss the performance of the proposed model. Arrival Distance is the mean time between two successive instants when a token arrives in a place, that is, the mean times of two successive instants for all the devices are less than the one in CC system. Throughput Rate is the rate of tokens which are departed from a place and indicates the throughput of each device. In CC, the rate of departed tokens in the ASRS, CNC and assembly device is less than the proposed DC system, that is, in the CC; the devices spend most of their time in the idle mode which will result in decreasing the rate of production. This has been proved by Waiting Time as well. Waiting Time is the time where each device spends and waits for the next operation. Service Time is the time where a device expends in order to perform a specific task. There is a slight difference and this is due to the local processors in the DC which is higher than the conventional centralized PLC system.

Table 6-2 Comparison between Centralized and Proposed Distributed Control System

Parameters	Centralized	Distributed
Initialization	Slow	Fast
Flexibility	No	Yes
Reconfigurability	Offline - Static	Online – Dynamic
Robust Against Failures	No	Yes

As a summary, Table 6.2 outlines the comparison between the centralized and proposed distributed control system. The initialization of the distributed control system is very fast comparing with the centralized system since the local processors will be initialized faster than the PLC. The centralized systems suffer from the flexibility whereas in the distributed systems, flexibility is achieved by employing IEC 61499 function block standard. Real time reconfiguration is also one of the most important characteristics of the IEC 61499 standard where system reconfiguration can be performed online whereas in the centralized PLC-based systems, reconfigurations will be done offline. This will decrease the robustness of the system against failures. The mechatronic devices are very prone to failure especially in the level of sensors and actuators. Therefore, the control system should be robust enough in order to handle any failure. This has been achieved by employing IEC 61499 standard where the devices can respond to the failures without any disruption in other device’s task.

This section evaluates the performance for the discussed case study. The overall performance of the proposed modeling scheme will be studied in the next chapter.

Chapter 7

PERFORMANCE EVALUATION OF IWSAN ON THE BASIS OF IEC 61499 STANDARD

7.1 Introduction

In the previous chapter industrial wireless sensor and actuator networks have been modeled based on IEC 61499 standard. UML has been used as a modeling language and the performance for a case study has been presented. In this chapter, the performance of IWSAN will be evaluated generally. Petri net as a modeling tool for Discrete Event Systems (DES), is used to evaluate the performance of the proposed system since both IEC 61499 function blocks standard and WSN are event-driven and events occur discretely. Several system parameters such as: coverability, reachability, transient and steady state analysis will be discussed. This chapter also identifies the utilization of each electronic resource in node so that one can come up with an energy-efficient design paradigm.

7.2 Petri Net Model

Petri Net is a tool for modeling and analyzing a Discrete Event System (DES). It was first invented by C. A. Petri in 1960 as a model for describing behavioral as well as performance evaluation of the event-based systems. It is also used for modeling asynchronous and concurrent systems. As a mathematical tool, it can be used to evaluate the performance of the system using deterministic or stochastic parameters.

Petri Net exhibits a number of properties that make it very popular in the context of discrete manufacturing systems to represent machines, robots and conveyors in a production line, for instance. The system is then described and analysis by various factors such as, controllability, reachability, liveness, boundedness and etc.

A Marked Petri Net graph is a 5-tuple function which is defined as follows:

$$PN = (P, T, A, w, M_0) \qquad (7.1)$$

where $P = \{P0, P1, \dots, Pm\}$ is the finite set of places, $T = \{T0, T1, \dots Tn\}$ is the finite set of transitions ($P \cup T \neq \emptyset$ and $P \cap T = \emptyset$), $A \subseteq (P \times T) \cup (T \times P)$ is the set of arcs from places to transitions and from transitions to places, $w: A \rightarrow \{0,1, \dots\}$ is the weight function on the arcs and M_0 is the initial marking. A can be re-written in terms of inputs and outputs as follows [81]: $I: (P \times T) \rightarrow N$ is a subset of A which contains only the arcs from places to transitions and $O: (T \times P) \rightarrow N$ is the set of arcs from transitions to the places. In both cases N is a non-negative integer number.

Figure 7.1 presents a simple resource in a node device according to the proposed model implemented in Function Block Development Kit (FBDK). This resource is corresponded to the sensor resource shown in Figure 6.1. It is a network of function blocks each defined to perform a specific task and interconnected through the event and data interfaces. Upon starting, the “START” function block issues event to all the blocks in order to perform initialization and then it will be shown by “INITDone” function block as a simple Boolean variable. The "Read” is basic function block aiming to read the data from the sensor and activating by either a sensor reading or a request from the other resources. The latter will be performed by “Receive” function block connected to the REQ event of the “Read” function block. When the reading is finished it will be broadcasted to other resources such as MCU by activating the

REQ event of the "Transmission" function block for further processing. The "Receive" and "Transmission" function blocks are service interface function blocks and their task is to publish and subscribe data among the resources (SUBL_0 and PUBL_1).

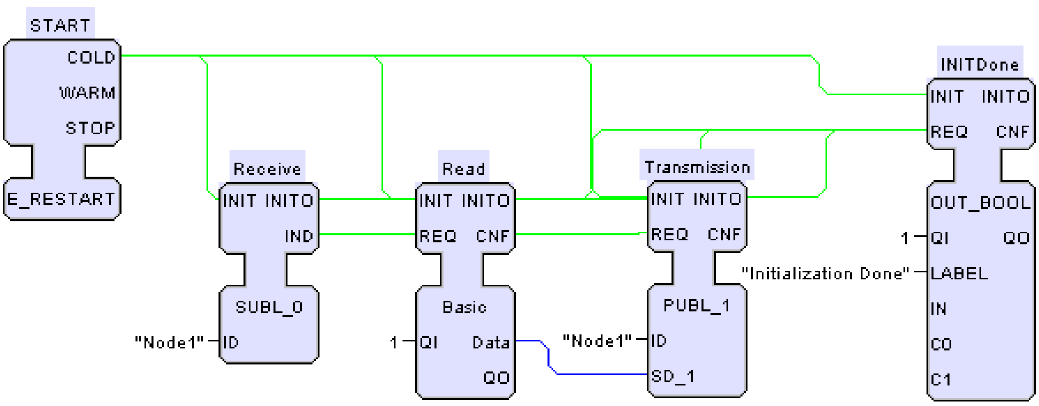

Figure 7-1: Sensor Resource in a Node Device

Similarly for other resources in a node device, there will be a network of function blocks each of which intended to perform a specific task. The corresponding Petri Net graph of a node device is presented in Figure 7.2 and it is developed using PIPE [82]. The timed-Petri net model has been used since each event expends a specific amount of time. Table 7.1 presents the parameters which are used in Figure 7.2.

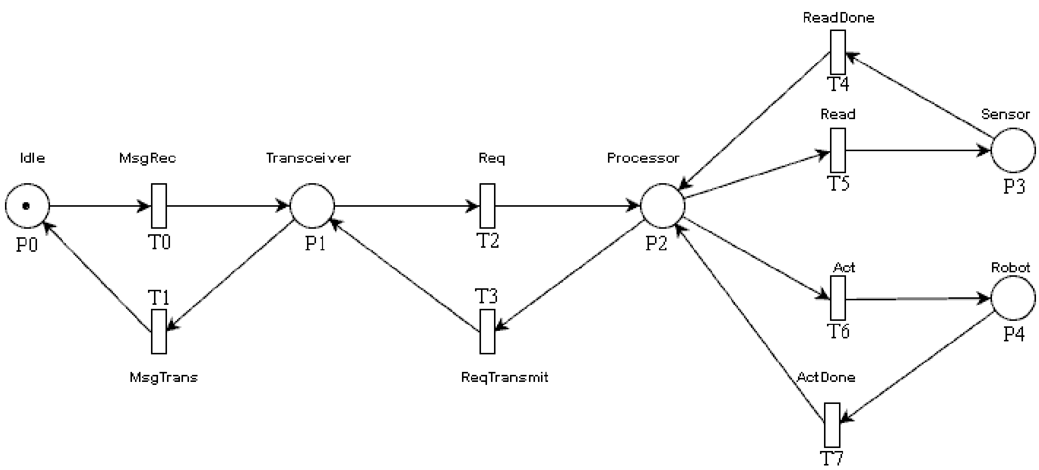

Figure 7-2: General Petri Net Model of a Node Device

This is a general Petri net model for a wireless sensor and actuator node. For the sake of simplicity, a sensor and an actuator are considered. It is also assumed that, the node has been initialized and it is now in the idle mode. This has been shown with a token in the $P0$ place. Places are corresponded to the node's state and the events in the function block model are presented by transitions. Table 1 identifies the places and transitions.

Table 7-1: Places, Transition and their notations

Places and Transitions	Notation
P0	Idle Mode
P1	Transceiver
P2	Processor
P3	Sensor
P4	Robot
T0	Receiving a Message
T1	Transmitting a Message
T2	Request to Process
T3	Request to Transmit
T4	Read Done
T5	Sensor Reading
T6	Actuating
T7	Actuation Done

The scenario is as follows: when a message from the other nodes received by the transceiver, a request will be sent to the processor unit to process. In order to perform the requested task, the data will be read by the sensor and hence a proper signal will be issued to the robot to act accordingly. Upon finishing, the result will be transmitted to the other nodes in the network.

7.3 Transient Analysis

In this section the performance analysis of the proposed mode by state space equations will be discussed. This is an alternative approach to demonstrate the dynamic behavior of the Petri net model. Given the initial state, the next state of the system can be determined by the following state space equation [83]:

$$x_{k+1} = x_k + uA \qquad (7.2)$$

where x_{k+1} is a $1 \times m$ row vector presents the next state and u is a $1 \times n$ row vector $u = [0,0, \dots ,1, \dots ,0]$ with all entries zero expect the jth element as one which presents the jth transition currently firing. Matrix A is called the incidence matrix which is a $n \times m$ matrix presenting the difference between the output and input matrices. Each entries of A will be obtained as follows:

$$a_{ji} = w(t_j, p_i) - w(p_i, t_j) \quad (7.3)$$

where $w(t_j, p_i)$ and $w(p_i, t_j)$ are the weight functions as well as the entries of output (A_O) and input (A_I) matrices, respectively. As a result, the Petri net graph shown in Figure 7.2, contains 5 places and 8 transitions, that is, $n = 8$ and $m = 5$. The corresponding input, output and incidence matrices are obtained as follows:

$$A_I = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_O = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$A = A_O - A_I = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix}$$

The state of each node can be obtained at any time by the developed state space equation. One of the properties of Petri net is the ability of analyzing the system in terms of stability, reachability and boundedness. Coverability tree can be used in order to demonstrate the reachability and boundedness. Figure 7.3 shows the coverability tree for the tangible states obtained for the Petri net graph (Figure 7.2)

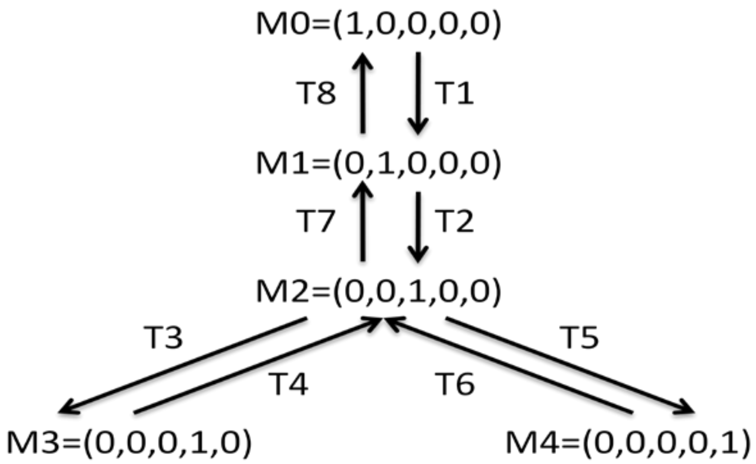

Figure 7-3: Coverability (Reachability) Tree for Wireless Sensor & Actuator Node

Coverability tree shows all the tangible states which are reachable from the initial state $M0$. That is, $RS(M_0) = \{M_1, M_2, M_3, M_4\}$. The proposed system contains 4 tangible states excluding the initial state as shown in Table 7.2.

Table 7-2: The set of all tangible states

	P0	P1	P2	P3	P4
M0	1	0	0	0	0
M1	0	1	0	0	0
M2	0	0	1	0	0
M3	0	0	0	1	0
M4	0	0	0	0	1

The average numbers of tokens have been calculated for 100 events firing and 1043 time units (Table 7.3). The node has the maximum number of tokens in place $P2$ which is the processing mode and the minimum number of token in sensor place $P3$.

Table 7-3: Average number of tokens

Places	Average Number of Tokens
Idle (P0)	0.06232
Transceiver (P1)	0.11505
Processor (P2)	0.73442
Sensor (P3)	0.04218
Robot (P4)	0.04602

Therefore by inspecting the results obtained, it can be seen that the node with the proposed model, is bounded, reachable and also there is no deadlock in the Petri net graph and hence it is live. The system is conservative since the total number of tokens is constant for all the tangible states. That is, the summation of the numbers in each tangible state is constant meaning that token neither created nor destroyed.

7.4 Steady State Analysis

Stochastic Petri Net (SPN) is a Petri net model where the firing rates are random and are not constant. This can be used to evaluate the Petri net models stochastically and obtain the steady state probabilities. For this purpose Markov chain will be utilized where the places are replaced by the states (Figure 7.4). The firing rate of a transition T_i is denoted by λ_i and it is shown on each arc.

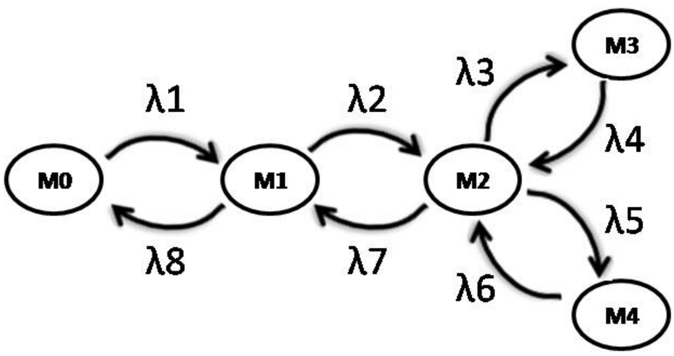

Figure 7-4: Markov chain graph for the proposed Petri net model

The transition rate matrix for the Markov chain model can be obtained as follows:

$$Q = \begin{bmatrix} -\lambda_1 & \lambda_1 & 0 & 0 & 0 \\ \lambda_8 & -\lambda_2 - \lambda_8 & \lambda_2 & 0 & 0 \\ 0 & \lambda_7 & -\lambda_3 - \lambda_5 - \lambda_7 & \lambda_3 & \lambda_5 \\ 0 & 0 & \lambda_4 & -\lambda_4 & 0 \\ 0 & 0 & \lambda_6 & 0 & -\lambda_6 \end{bmatrix}$$

To obtain the steady state probabilities(π_i), the following equation can be used [79]:

$$(\pi_0\ \pi_1\ \pi_2\ \pi_3\ \pi_4)Q = 0$$

$$\pi_0 + \pi_1 + \pi_2 + \pi_3 + \pi_4 = 1 \quad (7.4)$$

Solving equation 7.4 and substituting the firing rates given in Table 5, the following steady state probabilities will be achieved:

$$\pi_0 = 0.0321,\ \pi_1 = 0.0385,\ \pi_2 = 0.0561,\ \pi_3 = 0.1122,\ \pi_4 = 0.7611.$$

Table 7-4: Transitions and their corresponding firing rates

Transitions	Firing Rates (Time unit)
$\boldsymbol{\lambda_1}$	5
$\boldsymbol{\lambda_2}$	4
$\boldsymbol{\lambda_3}$	10
$\boldsymbol{\lambda_4}$	4
$\boldsymbol{\lambda_5}$	50
$\boldsymbol{\lambda_6}$	4
$\boldsymbol{\lambda_7}$	4
$\boldsymbol{\lambda_8}$	5

Each probability is associated with a tangible state belongs to the reachability set. Therefore based on these probabilities, one can determine the place (s) which a node stays with high probabilities. That is, M_4 is a place with high probability and this confirms that a node will expend most of its time in actuating.

7.5 Discussion

Wireless sensor and actuator node has been modeled using IEC 61499 function blocks where both are event based systems. Petri net as an analytical tool has been used for demonstrating the structural as well as the dynamical behavior of the proposed model. Performance of the proposed model has been measured by using the incidence matrix where the state of the system at any time instance can be analyzed using initial state. The system contains 5 marking places including the initial state. Theses analysis can be used when designing wireless sensor and actuator node platforms. Each place is corresponding to a resource in IEC 61499 function block. Therefore, the performance of each resource as well as the resource state and other parameters can be determined at any time using the developed analysis. For instance, table 6 depicts the global statistics of each resource in a wireless sensor and actuator node obtained by MATLAB Petri net Toolbox [80] for 100 events and 1043 units of time.

Table 7-5: Global Statistics of Places

Places	Arrival Sum	Arrival Rate	Arrival Distance	Waiting Time	Energy mJ
Idle	12	0.011505	86.9167	5	0.567
Transceiver	27	0.025887	38.6296	4.4444	469.463
Processor	38	0.036433	27.4474	20.7027	31.054
Sensor	11	0.010547	94.8182	4	4
Actuator	12	0.011505	86.9167	4	40

Arrival sum and Arrival rate are the total number of arrived tokens and the rate of arrival, respectively. Therefore the rate of using of each resource in a node can be analyzed. Arrival Distance shows the mean time between two successive instants when a token arrives in a place. This can be utilized to measure the performance of each resource in the proposed modeling. Waiting Time demonstrates the mean time where a token spends in each place. This parameter may be used to measure the time that a resource expends to perform a specific task. For instance, the Processor expends the maximum time unit among the other resources. Energy is the power consumption times the expended time in a place. For instance, the energy consumption of a typical sensor node such as TelosB has been obtained. The maximum amount of energy is expended by the transceiver place which is the most consuming resource in a sensor node and it is compliant to [84].

As a result, the design of wireless sensor and actuator node should be in such a way that, the usage and the performance of each resource is efficiently organized and this will help the users to design the node platform more proficiently. Any additional sensor or actuator may be amended as a place in the developed Petri net graph with the corresponding transition events.

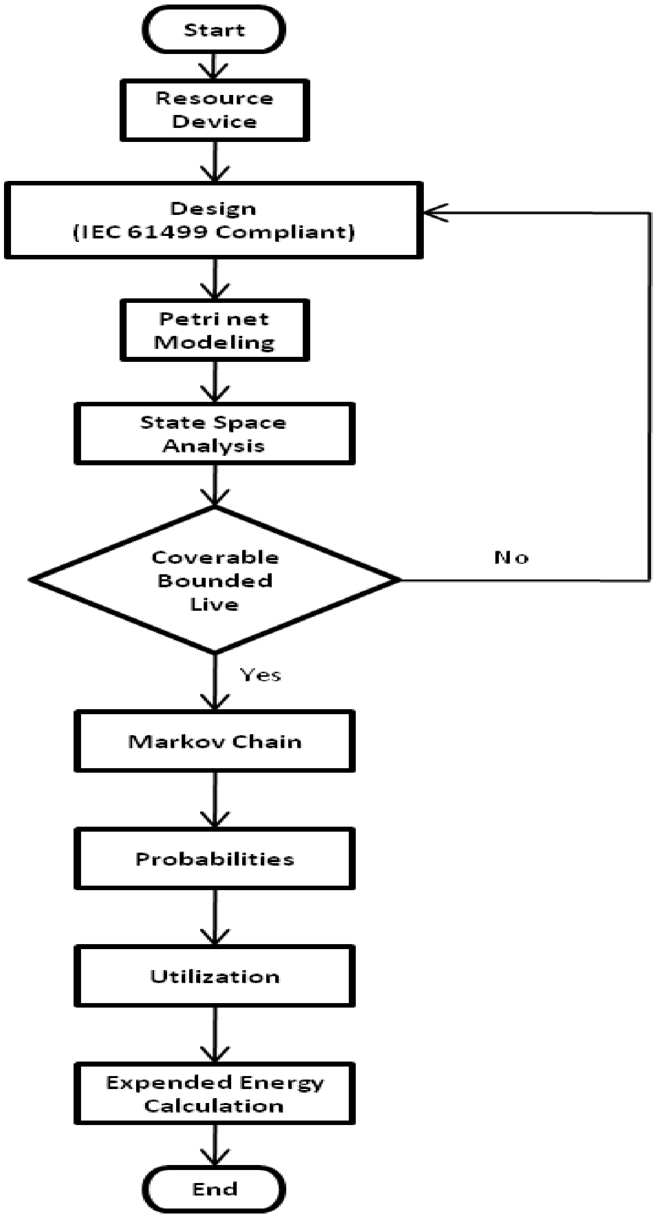

Figure 7-5: Design and Analysis Flowchart

Figure 7.5 summarizes the design and analysis flowchart where user will first start determining the number of resources and devices for a specific node. The design should be IEC 61499 compliant. The designed system will be modeled using Petri net tool for both transient and steady state analysis where coverability, boundedness and liveness of the system will be obtained. The utilization of each resource and hence the energy consumption will be achieved by Markov chain model and the corresponding probabilities.

Chapter 8

A PROTOTYPE SOFTWARE TOOL

8.1 Introduction

In this chapter, the prototype software tool which is developed for transferring the UML diagrams to the corresponding IEC 61499 file. This chapter is presented though an example of a simple mechatronic device and using the simplified version of the proposed methodology in chapter 6.

8.2 Modeling Methodology

This section outlines the modeling methodology for a mechatronic device with the following specification.

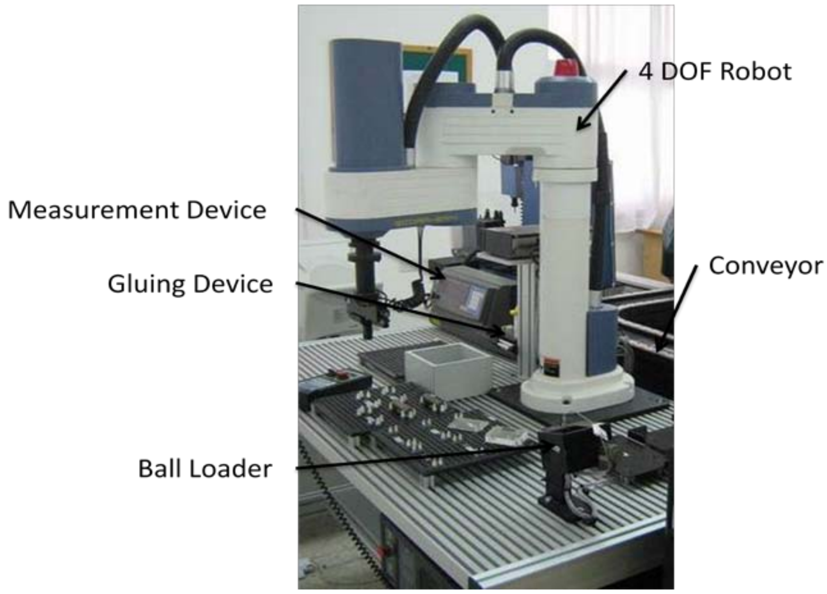

Figure 8-1: Mechatronic device with different resources

This is the flexible manufacturing system laboratory in Eastern Mediterranean University (Figure 8.1). The system contains several mechatronic devices with different competences. Figure 8.2 presents one of the devices with the following operations:

There is a 4 Degree of Freedom (DOF) robot which takes the part from the conveyor and places on the measurement device to check the dimension. If verified, it is then taken to the ball loader to load some balls. Gluing is the last operation which is performed on the part and then it will be placed back on the conveyor to transfer to the storage.

As it is also mentioned in [85] this system is equipped with wired sensors and Programmable Logic Controller (PLC). It also suffers from the lack of flexibility and reconfigurability. The aim to redesign the system based on the IEC 61499 standard and wireless sensors and actuators.

The proposed methodology is based on three layers namely, System Requirements, Design and Implementation (Figure 8.2).

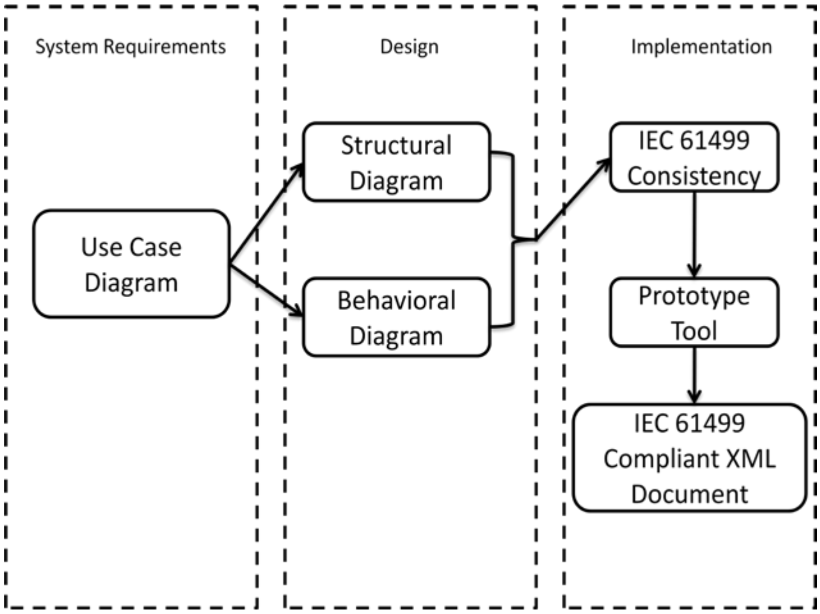

Figure 8-2: Proposed Methodology utilized for Software Tool

UML Use case diagram is utilized in order to capture the requirements defined by user. It contains an actor along with a use case which defines the scenario of the system by plain text (the operation of mechatronic device). Design phase includes two parts namely, structural and behavioral of the system. UML class diagram is employed to demonstrate the static view of the system whereas UML interaction diagram is utilized to present the dynamic behavior of the system. A prototype tool is designed for the implementation phase with the following architecture (Figure 8.4).

IBM Rational Rose [86] is utilized as a CASE tool for UML diagrams such as use case, class and interaction diagrams. Rose does not generate the corresponding XML file and for this reason, Unisys Rose XML Tool [87] as a software add-ins will be integrated to Rose in order to generate the XMI documents. XML Metadata Interchange (XMI) is a new standard developed by Object Management Group [88] for metadata information exchange by means of XML documents. XMI is widely used for saving UML models in XML. This will give the ability to transform UML models between tools. The generated XMI documents will be parsed by the proposed tool. The generated XML document is fully compliant with IEC 61499 standard, since it is validated by the given Document Type Definition (DTD) library element provided by part II of IEC 61499 standard [56]. Ultimately, there will be a database of XML files generated for each wireless sensor and actuator node that can be used for further processing such as other CASE tools or for exchanging information between mechatronic devices. Figure 8.3 demonstrate the overall structure of the tool.

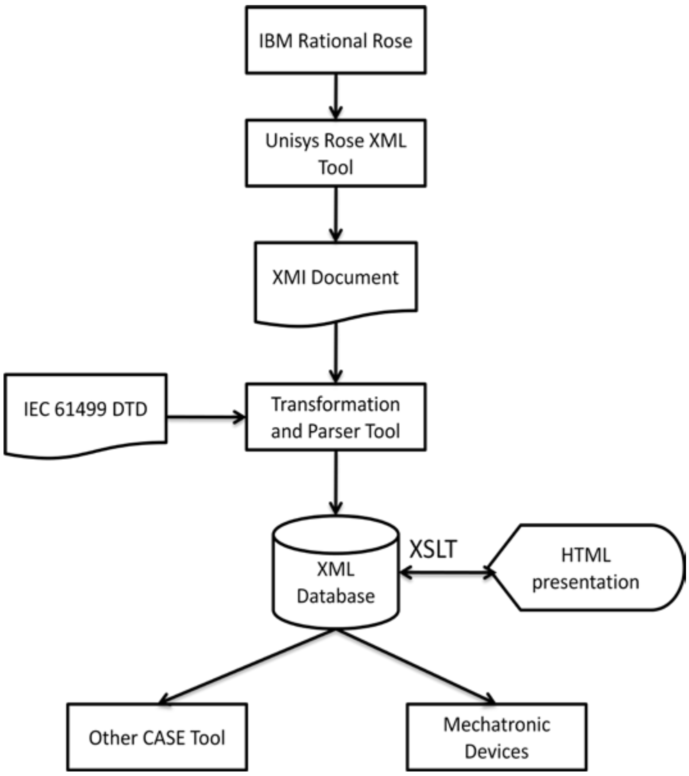

Figure 8-3: Overall structure of the proposed tool

Consider the mechatronic device presented in Figure 8.1. In order to design the wireless sensor and actuator, a transceiver, a processor, robot, testing and gluing are required where each of which are capable of performing several operations.

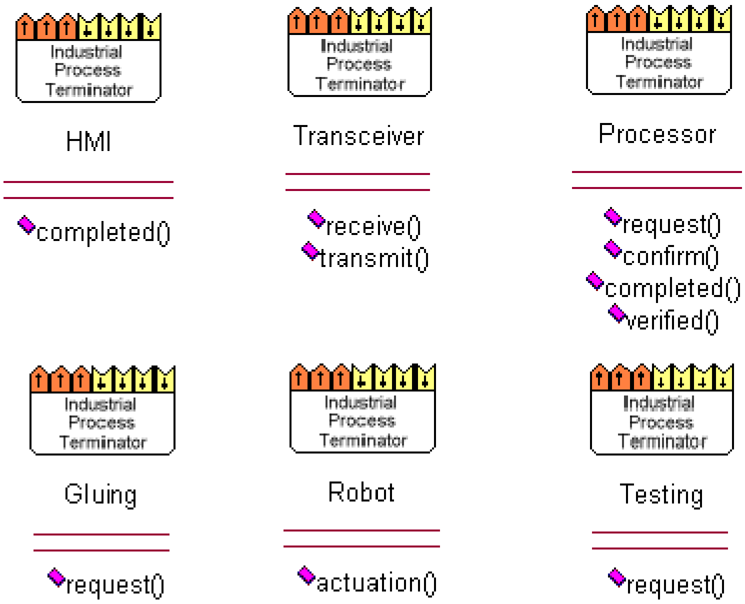

Figure 8-4: UML Class diagram

Figure 8.4 depicts the UML class diagram developed in IBM Rational Rose which is integrated in CORFU ESS. There are six classes defined for designing the wireless sensors and actuators node with different operations. Human Machine Interface (HMI) class is developed for monitoring and interaction purposes between the user and system. Transceiver is used to establish a communication among the nodes. The stereotype of the classes is Industrial Process Terminator (IPT) which is used for the mechatronic devices [61].

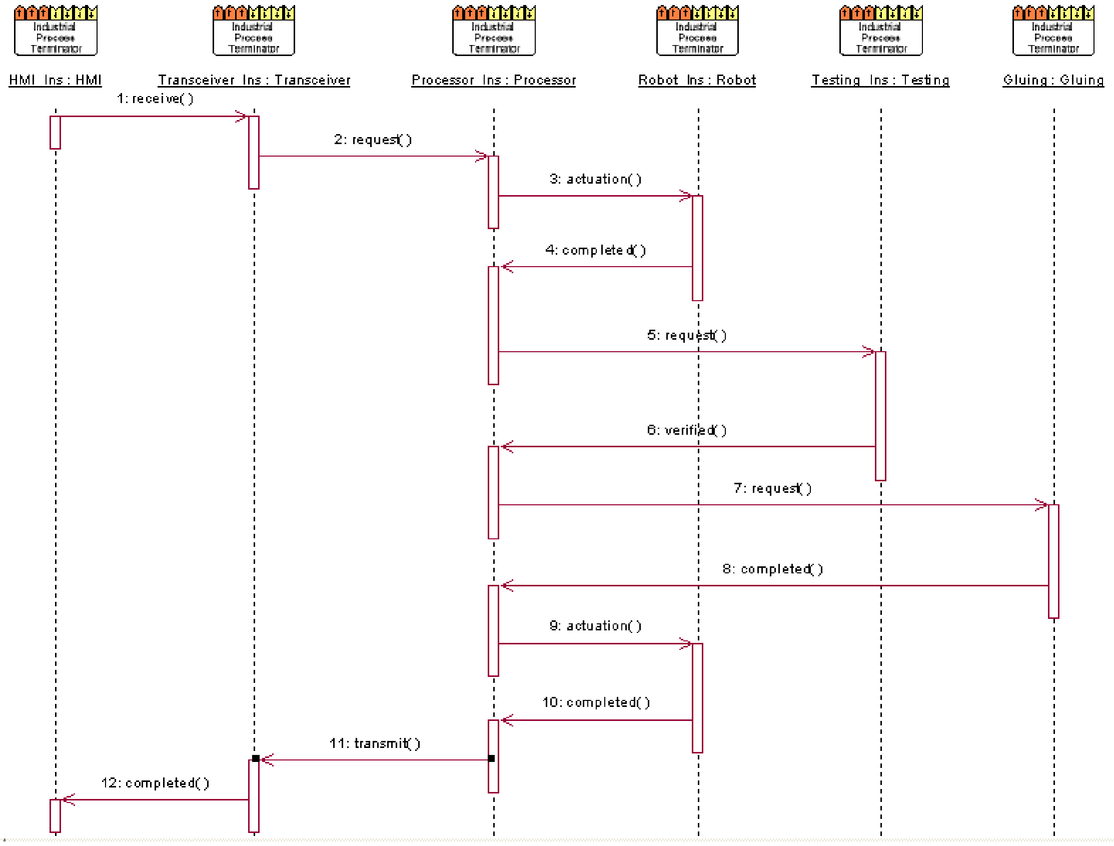

Figure 8-5: UML Interaction diagram

Figure 8.5 presents the activities in a node for an operation. A command from HMI or a message from other nodes is received by the transceiver and sends the request to the processor in order to perform the tasks which are robot operation, testing and gluing.

Unisys Rose XML Tool transfers the UML diagrams to the corresponding XMI document. XMI stands for XML Metadata Interchange which utilized for information exchange via XML. Both UML Class and Interaction diagrams are transferred to XMI format (Figure 8.6). There are some tags and elements which are important for transferring the XMI documents to the corresponding XML file which is compliant to IEC 61499 standard. For instance, UML:Model identifies the whole model and its specification. Table 8.1 depicts the tags and their explanations.

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--  <!DOCTYPE XMI SYSTEM 'UMLX13-11.dtd' >  -->
- <XMI xmi.version="1.1" xmlns:UML="href://org.omg/UML/1.3" timestamp="Fri Apr 27 21:32:38 2012">
 - <XMI.header>
  - <XMI.documentation>
     <XMI.exporter>Unisys.JCR.2</XMI.exporter>
     <XMI.exporterVersion>1.3.6</XMI.exporterVersion>
    </XMI.documentation>
    <XMI.metamodel xmi.name="UML" xmi.version="1.3" />
   </XMI.header>
 - <XMI.content>
    <!--  ==================== IWSAN    [Model] ====================  -->
  - <UML:Model xmi.id="G.0" name="IWSAN" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false">
   - <UML:Namespace.ownedElement>
      <!--  ==================== IWSAN::IndustrialProcessTerminator    [Actor] ====================  -->
      <UML:Actor xmi.id="S.117.2132.24.1" name="IndustrialProcessTerminator" visibility="public" isSpecification="false" isRoot="true" isLeaf="true"
        isAbstract="false" namespace="G.0" />
      <!--  ==================== IWSAN::HMI    [Class] ====================  -->
    - <UML:Class xmi.id="S.117.2132.24.2" name="HMI" visibility="public" isSpecification="false" isRoot="true" isLeaf="true" isAbstract="false" isActive="false"
        namespace="G.0">
     - <UML:Classifier.feature>
        <!--  ==================== IWSAN::HMI::completed    [Operation] ====================  -->
      - <UML:Operation xmi.id="S.117.2132.24.3" name="completed" visibility="public" isSpecification="false" ownerScope="instance" isQuery="false"
          concurrency="sequential" isRoot="false" isLeaf="false" isAbstract="false" specification="">
       - <UML:BehavioralFeature.parameter>
           <UML:Parameter xmi.id="XX.27.2132.24.4" name="completed.Return" visibility="public" isSpecification="false" kind="return" type="G.28" />
         </UML:BehavioralFeature.parameter>
        </UML:Operation>
       </UML:Classifier.feature>
      </UML:Class>
      <UML:Stereotype xmi.id="S.117.2132.24.24" name="IndustrialProcessTerminator" visibility="public" isSpecification="false" isRoot="false" isLeaf="false"
        isAbstract="false" icon="" baseClass="Class" extendedElement="S.117.2132.24.2 S.117.2132.24.4 S.117.2132.24.7 S.117.2132.24.12 S.117.2132.24.14
        S.117.2132.24.16" />
      <!--  ==================== Byte    [DataType] ====================  -->
      <UML:DataType xmi.id="G.28" name="Byte" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
```


Figure 8-6: XMI Document generated by Unisys Rose

Table 8-1: XMI Tags

Tag	Explanation
UML:Model	Identifies the whole system
UML:Class	Defines the UML classes
UML:Operation	Identifies the operations which are defines for each class
UML:Message	Identifies the messages utilized in interaction diagrams

As a result, tags and their corresponding values will be utilized in order to generate the XML. Part 2 of IEC 61499 standard outlines the software requirements and therefore the generated XML file must be compliant to this part. The standard tags and elements are defined in a file called Document Type Definition (DTD). As a result, after generating the XML file, it must be validated with the DTD file provided by IEC 61499 standard.

The developed tool parses the XMI file and generates the corresponding XML file which will be validated using the DTD file.

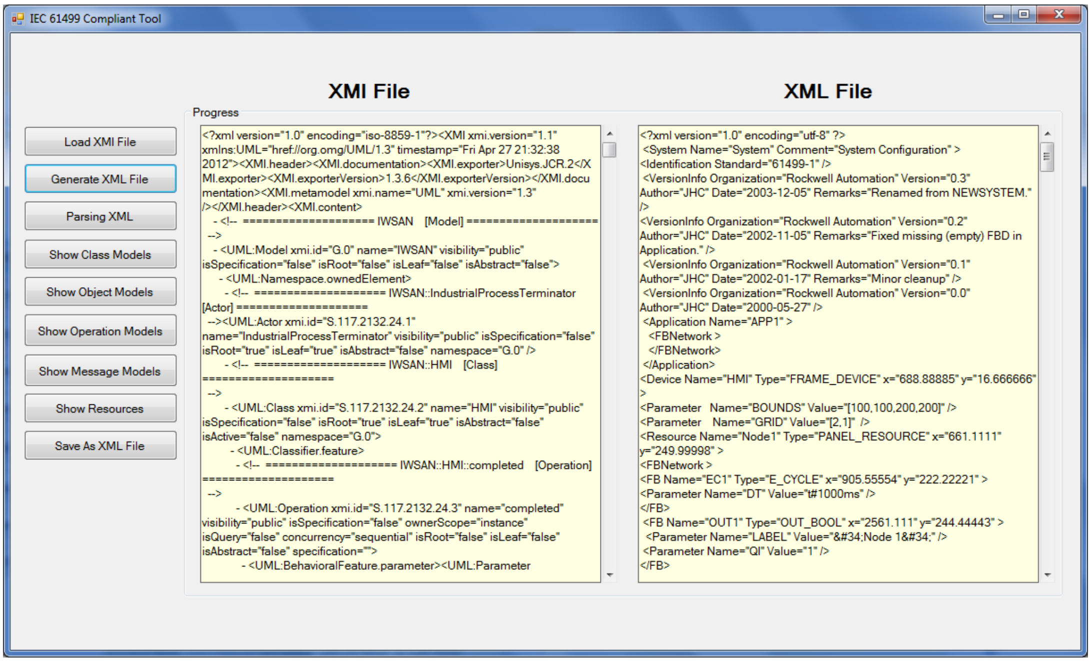

Figure 8-7: IEC 61499 Compliant software tool

Figure 8.7 presents a snapshot of the developed tool. User loads the generated XMI file from the UML diagrams into the tool. In the left pane the XMI file is shown. All the classes and their operations are given in the XMI file. There is a button called Parsing XMI which parses the XMI file. This is presented in Figure 8.8.

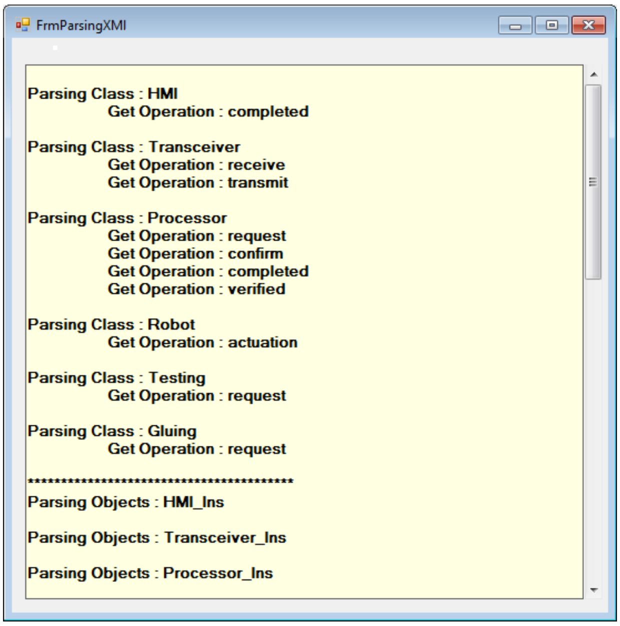

Figure 8-8: XMI Parsing

All the classes and their corresponding operations are extracted. By clicking "Show Class Models" button all the classes will be shown. "Show Object Models" will demonstrate the instances of the classes which are used in the UML diagrams and also called Automation Objects. Each class has one or more operations which will be shown by clicking "Show Operation Models" (Figure 8.9).

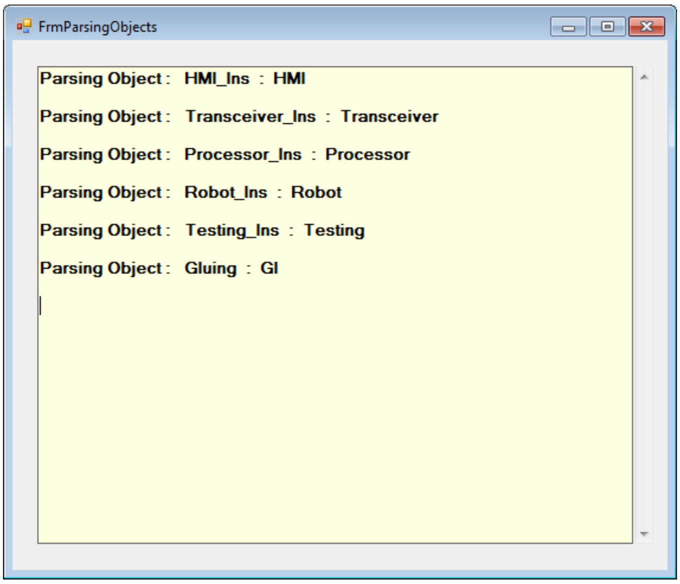

Figure 8-9: Objects and their corresponding classes

XMI file also contains messages which are given in the UML interaction diagrams. These messages are indeed the events in function blocks which are utilized to establish the collaboration. In the generated XML file by this tool, messages will be shown in the event connections of each device. As it is shown in figure 8.10, the sender and receiver of each object is also extracted.

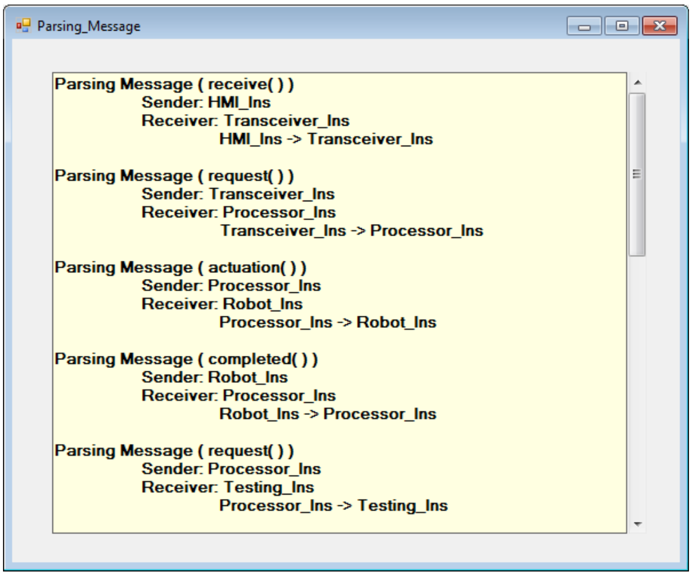

Figure 8-10: Extracted Messages

“Show Devices” illustrates the IEC 61499 standard devices which will be used in the XML file. As a result, each class in the XMI file will be a device in the generated XML file. By clicking “Generate XML File” software will generate a XML file and validates it by the given DTD file of IEC 61499 standard in to check consistencies. The generated XML file is fully compliant with IEC 61499 standard and therefore can be used in any mechatronic devices which support XML files. It also can be imported into other CASE tools for further processing.

Chapter 9

CONCLUSIONS AND FUTURE WORKS

In this thesis, a novel methodology for the development of industrial wireless sensor and actuator network has been presented. This methodology is based on IEC 61499 standard which is utilized recently in the context of distributed control systems to give flexibility and reconfigurability to a mechatronic devices. This standard does not capture the user requirements in a high level of abstraction and therefore UML has been used to overcome this problem. The proposed methodology contains three phases, namely system requirement, design and implementation. The result of design phase goes through a verification step to check the consistencies. The current problems of a case study have been addressed and the system has been redesigned based on the proposed methodology to achieve the following advantages:

- Conventional wired sensors are controlled by means of a central processor unit. This reduces the flexibility and configurability. Multi-agent systems and distributed control systems are employed to provide these characteristics for wireless sensor networks.
- Energy is a key factor in wireless sensor networks and therefore the network should be designed in a way to achieve the maximum network life time.
- The user requirements are captured in a high level of abstraction with the usage of UML use case diagram at machine level.

- The industrial wireless sensors and actuators are modeled with the help of IEC 61499 function block standard.

Any reconfiguration in any level is performed without disruption in other mechatronic devices on the basis of IEC 61499 function block standard. The performance evaluation shows that distributed scheme performs better with respect to centralized system. This has been shown by Petri net models and MATLAB simulation. A prototype software tool has been designed in order to translate the UML models to XML files on the basis of IEC 61499 standard.

REFERENCES

[1] F. Su, W. Sankarasubramaniam, Y. Cayirci, E. Akyildiz, "Wireless Sensor Networks: a survey," *International Journal of Computer and Telecommunications Networking*, vol. 38, pp. 393-422, 2002.

[2] M. Hannikainen, H. Amalainen, Kuorilehto M, "A middleware for task allocation in wireless sensor networks," in *Proc. 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'05)*, Berlin, Germany, 2005, pp. 821-826.

[3] V.C. Gungor and G.P., Hancke, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches," *IEEE Transaction on Industrial Electronics*, vol. 56, no. 10, 2009.

[4] Institute of Electrical and Electronic Engineers, "standard for a smart transducer interface for sensors and actuators- transducer to microprocessor communication protocols and transducer electronic data sheet (TEDS) formats," IEEE Std, 1451.2, 1997.

[5] Agilla, http://mobilab.wustl.edu/projects/agilla/.

[6] J. Koskinen, P. Kilpelainen, J. Rehu, P. Tukeva, and M., Sallinen, "Wireless Sensor Networks for infrastructure and industrial monitoring applications," in *ICTC*, 2010, pp. 250-255.

[7] M. Antoniou, M.C. Boon, P.N. Green, P.R. Green, and T.A., York, "Wireless sensor networks for industrial processes," in *IEEE Sensors Applications Symposium*, 2009, p. 13.

[8] K.S. Low, W.N.N. Win, and M.J., Er, "Wireless Sensor Networks for Industrial Environments," in *International Conference on Computational Intelligence for Modeling, Control and Automation*, 2005, pp. 271-276.

[9] Xingfa Shen, Zhi Wang, and Youxian Sun., "Wireless sensor networks for industrial applications," in *Fifth World Congress on Intelligent Control and Automation, (WCICA),* vol. 4, 2004, pp. 3636-3640.

[10] Li Zheng., "Industrial wireless sensor networks and standardizations: The trend of wireless sensor networks for process automation," in *Proceedings of SICE Annual Conference*, 2010, pp. 1187-1190.

[11] K. Khakpour and M.H., Shenassa, "Industrial Control using Wireless Sensor Networks," in *3rd International Conference on Information and Communication Technologies: From Theory to Applications, (ICTTA),* , 2008, pp. 1-5.

[12] M. Shanmugaraj, R. Prabakaran, and V.R.S., Dhulipala, "Industrial utilization of wireless sensor networks," in *International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT),* 2011, pp. 887-891.

[13] P. Ferrari, D. Marioli, E. Sisinni, A. Taroni A. Flammini, "Wired and wireless sensor networks for industrial applications," *Microelectronics Journal*, vol. 40, no. 9, pp. 1322-1336, 2009.

[14] F. Salvadori, "Monitoring and Diagnosis in Industrial Systems Using Wireless Sensor Networks," in *IEEE International Symposium on Intelligent Signal Processing, (WISP)*, 2007, pp. 1-6.

[15] Peng Jiang, Huijin Ren, Lei Zhang, Zhi Wang, and Anke Xue., "Reliable Application of Wireless Sensor Networks in Industrial Process Control," in *The Sixth World Congress on Intelligent Control and Automation (WCICA)*, 2006, pp. 99-103.

[16] S. Peter, O. Stecklina, and P., Langendoerfer, "An engineering approach for secure and safe wireless sensor and actuator networks for industrial automation systems," in *IEEE Conference on Emerging Technologies & Factory Automation (ETFA)*, 2009, pp. 1-8.

[17] J. H. Lee, T. Kwon, J. Song, "Group Connectivity Model for Industrial Wireless Sensor Networks," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 5, pp. 1835-1844, 2010.

[18] A. Nivento, L.P. Carloni, and A., Sangiovanni-Vincentelli, "Platform-Based Design of Wireless Sensor Networks for Industrial Applications," in *Proceedings of Design, Automation and Test in Europe (DATE)*, 2006, pp. 1-6.

[19] J. Slipp, "WINTeR: Architecture and Applications of a Wireless Industrial Sensor Network Testbed for Radio-Harsh Environments," in *6th Annual Communication Networks and Services Research Conference (CNSR)*, 2008, pp. 422-431.

[20] S. Kamarthi, S. Berry, N.M. Vadde, "Smart sensors in a flow-line

manufacturing system: an agent-based simulation," in *International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, 2005, pp. 635-640.

[21] IEC, "Industrial Communication Networks-Profiles-Part2: Additional Fieldbus profiles for real-time networks ," International Electrotechnical Commision, ISO/IEC 8802-3, 2007.

[22] A Chandrakasan, and H. Balakrishanan W.B. Heinzelman, "An Application-Specific Protocol Architecture for Wireless Microsensor Network," *IEEE Transaction on Wireless Communication*, vol. 1, no. 4, pp. 660-670, October 2002.

[23] W. Heinzelman, *Application-specific protocol architectures for wireless networks*, PhD Thesis ed.: Massachusetts Institute of Technology, 2000.

[24] W. Heinzelman, "A. Chandrakasan, and H. Balakrishnan. Energy efficient communication protocol for wireless microsensor networks," in *Hawaiian International Conference on Systems Science*, 2000, pp. 1-10.

[25] D. Song, "Probabilistic Modeling of Leach Protocol and Computing Sensor Energy Consumption Rate in Sensor Networks," Texas A&M University, 2005.

[26] Mudi Xiong, Ying Wang, "Monte Carlo Simulation of LEACH Protocol for Wireless Sensor Networks," in *Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2005, pp. 85-88.

[27] Y. Jianjun F. Yiming, "The Communication Protocol for Wireless Sensor Network about LEACH," in *Proceedings of the 2007 International Conference*

on Computational Intelligence and Security Workshops, 2007, pp. 550-553.

[28] S. Sivathasan, "RF/FSO and LEACH wireless sensor networks: A case study comparing network performance," in *IFIP International Conference on Wireless and Optical Communications Networks*, 2009, pp. 1-4.

[29] A. Al-zoubi, Y. Khamayseh, W. Mardini M. BaniYassein, "Improvement on LEACH Protocol of Wireless Sensor Network (VLEACH)," *International Journal of Digital Content Technology and its Applications*, vol. 3, no. 2, pp. 132-136, 2009.

[30] W. Jeong, S. Y. Nof, "Performance evaluation of wireless sensor network protocols for industrial applications," *Journal of Intelligent Manufacturing*, vol. 19, no. 3, pp. 335-345, 2008.

[31] W. Dargie, J. M. Reason M. Zimmerling, "Energy-Efficient Routing in Linear Wireless Sensor Networks," in *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2007, pp. 1-3.

[32] T. Garnett, and A. Chandrakasan M. Bhardwaj, "Upper boundson the lifetime of sensor networks," in *IEEE International Conference on Communications (ICC 2001)*, 2001, p. 785–790.

[33] T. Rappaport, *Wireless Communication: Principles & Practice*. New Jersey, USA: Prentice-Hall, 1996.

[34] Crossbow. Cross Bow Co. http://www.xbow.com

[35] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor

networks: A survey," *IEEE Wireless Communication Magazine*, vol. 11, pp. 6-28, December 2004.

[36] M. Hashemipour M. Bal, "Implementation of holonic scheduling and control in flow line manufacturing systems: die-casting case study," *Production Planning & Control*, August 2010.

[37] A. J. Younge, "Using Intelligent Agents in Wireless Sensor Networks," Rochester Institute of Technology, New York,.

[38] Java Development Framework (JADE). http://jade.tilab.com

[39] Federation for Intelligent Physical Agent (FIPA). http://www.fipa.org

[40] G. Roman, C. Lu, C. Fok, "Mobile agent middleware for sensor networks: an application case study," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, p. 2005.

[41] D. Marsh , A. Ruzzelli, R. Tynan G. M. P. O'hare, "Agents for wireless sensor network power management," in *In Proceedings of International Workshop on. Wireless and Sensor Networks* , 2005.

[42] J. Wall, P. Valencia, J. Ward G. Platt, "Tiny Agent Wireless Sensor Networks Controlling Energy Resources," *Journal of Networks*, vol. 3, no. 4, pp. 42-50, 2008.

[43] P. Guerrero, A. Buchmann K. Nawaz, "Towards a FIPA Compliant Multiagent Based Middleware Architecture foe Sensor Networks," in *International Symposium on Wireless Pervasive Computing*, 2008, pp. 339-343.

[44] H. Nishiyama, F. Mizoguchi M. Obayashi, "Secured Cooperative Multi-Agent System in Limited Resources for Intelligent Sensor Networks," in *31st Annual Conference of IEEE Industrial Electronics Society*, 2005.

[45] C. Muldoon, A. Schoofs, T. Campana, G. Hare, R. Tynan A. Ruzzelli, "Autonomous Management and control of Sensor Network-based Applicarion," in *International Conference on Computational Science and Engineering*, 2009, pp. 885-890.

[46] G. Hare, J. Grady, R. Tynan C. Muldoon, "Agent Migration and Communication in WSNs," in *Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2008, pp. 425-430.

[47] M. Ketel, "Agent Based Technology for Distributed Wireless Networks," in *40th Southeastern Symposium on System Theory*, 2008, pp. 79-83.

[48] E. Shakshuki, A. Matin S. Hussain, "Agent Based System Architecture for Wireless Sensor Networks," in *Proceeding of the 12th International Conference on Advanced Information Networking and Application*, p. 2006.

[49] G. Hare, M. Graddy, C. Muldoon R. Tynan, "Virtual Sensor Networks: An Embedded Agent Approach," in *International Conference on Parallel Distributed and Processing with Apllications*, 2008, pp. 926-932.

[50] D. Jain R. Patel, "A Multi agent System for Distributed Sensor Networks," in *International Conference on Advances in Computing, Control and Telecommunication Technologies*, 2009, pp. 823-826.

[51] H. Chan, "Agent-Based Factory Level Wireless Local Positioning System with ZigBee Technology," *IEEE Systems Journal*, vol. 4, no. 2, pp. 179-185, 2010.

[52] F. Lauri, A. Koukam, E. Tonye E. Tange, "The Coverage Problem in Wireless Sensor Networks by Holonic Multi-agent Approach," *International Journal of Computing and ICT Research*, vol. 3, no. 1, 2009.

[53] IEC, "IEC 61499: Function Blocks," International Electrotechnical Commission, Geneva, International Standard, First Edition, 2005.

[54] V. Vyatkin, *IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design*.: O3 Neidia and Instrumentation Society of America, 2007.

[55] J. H. Christensen, "Design Patterns for System Engineering in IEC 61499," Otto-von-Guericke-Universitat Magdeburg, Germany, 2000.

[56] IEC, "IEC 61499-1: Function Blocks - Part 1 Architecture," International Electrotechnical Commission, Geneva, International Standard, First Edition, 2005.

[57] IEC, "IEC 61499-2: Function Blocks - Part 2 Software Tools Requirements," International Electrotechnical Commission, Geneva, International Standard, First Edition, 2005.

[58] IEC, "IEC 61499-3: Function Blocks - Part 3 Application Guidelines," International Electrotechnical Commission, Geneva, International Standard, First Edition, 2005.

[59] IEC, "IEC 61499-4: Function Blocks - Part 4 Rules for Compliance Profiles,"

International Electrotechnical Commission, Geneva, International Standard, First Edition, 2005.

[60] HOLOBLOC. (2008) Function Block Development Kit. http://www.holobloc.com

[61] University of Patras. CORFU Engineering Support System (ESS). [Online].

[62] ISaGRAF. http://www.isagraf.com

[63] OOO Neida Group. http://www.oooneida.info

[64] S. Panjaitan and G. Frey, "Combination of UML Modeling and the IEC 61499 Function Block Concept for the Development of Distributed Automation Systems," in *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, 2006, pp. 766-773.

[65] Thramboulidis, "IEC 61499 in Factory Automation," in *Proceedings of IEEE International Conference on Industrial Electronics, Technology and Automation*, Bridgeport, USA, 2005.

[66] R. Tracht, R. Hirschfeld Torsten Heverhagen, "A Profile for Integrating Function Blocks into the Unified Modeling Language," in *Proceeding of Works on Specification and Validation of UML models for RT and Embedded Systems*, San Francisco, 2003.

[67] Unified Modeling Language. http://www.uml.org

[68] V. Vyatkin V. Dubinin, "UML-FB A Language for modeling and Implementation of Industrial-Process Measurement and Control Systems on the

Basis of IEC 61499 Standard," in *Proceeding of 6th International Conference of Science and technology, New Information Technology and Systems*, 2004, pp. 77-83.

[69] V. Dubinin, V. Vyatkin, and T. Pfeiffer, "Engineering of Validatable Automation Systems Based on an Extension of UML Combined With Function Blocks of IEC 61499," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*, 2005, p. 3996 – 4001.

[70] G. Frey S.D. Panjaitan, "Development process for distributed automation systems combining UML and IEC 61499," *International Journal of Manufacturing Research*, vol. 2, no. 1, pp. 1-20, 2007.

[71] T. Hussain and G. Frey, "UML-based Development Process for IEC 61499 with Automatic Test-case Generation," in *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, 2006, p. 1277 – 1284.

[72] ZigBee Alliance. http://www.zigbee.org

[73] HART Communication Protocol. http://www.hartcomm.org

[74] World Wide Web Consortium (W3C). http://www.w3.org

[75] TinyOS. http://www.tinyos.net

[76] Atmel Co. http://www.atmel.com

[77] Texas Instruments (TI). http://www.ti.com

[78] J. Chen, Y. Xiao, Y. Sun X. Cao, "Building-Environment Control with Wireless

Sensor and Actuator Networks: Centralized Versus Distributed," *IEEE Transaction on Industrial Electronics*, vol. 57, no. 11, pp. 3596-3605, 2005.

[79] M. Zhou R. Zurawski, "Petri Nets and Industrial Applications: A Tutorial," *IEEE Transaction on Industrial Electronics*, vol. 41, no. 6, pp. 567-583, 1994.

[80] MATLAB Petri Net Toolbox. http://www.ac.tuiasi.ro/pntool

[81] T. Liu, M. Martonosi. "Impala: A Middleware System for Managing Autonomic Parallel Sensor Systems", *PPoPP'03,* San Diego, CA, USA, June 2003.

[82] Platform Independent Petri Net Editor. http://pipe2.sourceforge.com

[83] S. Lafortune C. G. Cassandras, *Introduction to Discrete Event Systems*.: Kluwer Academic , 1999.

[84] R. Abrishambaf , S. N. Bayindir, M. Hashemipour, " Energy Analysis of Routing Protocols in Wireless Sensor Networks for Industrial Applications" *Journal of Systems and Control, Part I of Proceedings of the Institution of Mechanical Engineers*, vol. 226, Issue. 5, pp. 678-684, May 2012.

[85] R. Abrishambaf, M. Hashemipour, M. Bal, "Structural Modeling of Industrial Wireless Sensor and Actuator Networks for Reconfigurable Mechatronic Systems," *International Journal of Advanced Manufacturing Technology*, 10.1007/s00170-012-4070-y. 2012.

[86] IBM Rational Rose. http://www.ibm.com

[87] Unisys Rose XML Tool. http://www.ibm.com

[88] Object Management Group (OMG). http://www.omg.org