TECHNOLOGICAL CONSIDERATIONS FOR DEVS-BASED SYSTEM CONTROL

A thesis presented to the Department of Computer Science

African University of Science and Technology, Abuja

In partial fulfilment of the requirements for the award

MASTER OF SCIENCE DEGREE IN COMPUTER SCIENCE

By

IFEBUDE, BARNABAS CHUKWUDIKE

Supervised by

Professor Mamadou Kaba Traoré

African University of Science and Technology
www.aust.edu.ng
P.M.B 681, Garki, Abuja F.C.T
Nigeria

June 2016

TECHNOLOGICAL CONSIDERATIONS FOR

DEVS-BASED SYSTEM CONTROL

By

Ifebude, Barnabas Chukwudike

A THESIS APPROVED BY THE COMPUTER SCIENCE DEPARTMENT

RECOMMENDED:

____

Supervisor, Professor Mamadou Kaba Traoré

____

Head, Department of Computer Science

APPROVED:

____

Chief Academic Officer

____

Date

ABSTRACT

System control has grown from manual control where humans sense the environment, analyse the result and determine control actions to semi-automated control. In semi-automated control sensing is done by sensing devices and with the help of communication technologies, humans at a control centre view the environment, analyse it and determine control action. However, advances in computational and processing technologies has now shifted the most important role of making control decisions away from humans, to microcontrollers/microprocessors. The problem now is how to make these devices intelligent enough to make the best control decision. Different control methods have been used in system control they include, the proportion integral derivative (PID) control, Model-Based control and control based on Artificial Intelligence (AI). We proposed an architecture and explored technologies that can be used to realize Discrete Event System Specification (DEVS)-Based Control, which is a Model-Based approach. The proposed approach is based on a simulation model rather than the optimization model used in other Model-Based system control approaches. Using DEVS in system control, we can model physical systems to be controlled, run a fast simulation of the system based on the current state of the system and determine control actions. Two-fold Communication issues were identified and the research addressed different methods of solving the two-fold communication problems.

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest gratitude to God Almighty who provided me with His abundant blessing and opportunity to successfully conclude this programme.

I would like to express my appreciation to my supervisor and Head of Department, Prof. Mamadou Kaba Traoré for his scholarly assistance, constant support, and most importantly his words of advice. Even when research got boring and uninteresting at times, listening to him encouraged me to continue. I thank him for having given me the opportunity to undertake such a wonderful research. I am also grateful to Hamzat, Doyin, Ignace and all members of GReP for their assistance in this research work.

In addition, I would like to thank the Nelson Mandela Institute (NMI) and Huawei Technologies for their scholarship award to undertake the 18 month Master's programme. I would also like to acknowledge Prof Ousmane Thiare, Prof Lehel Csato, Dr Ekpe Okorafor, Prof Ben Abdallah, Prof Mohamed Hamada and other lecturers at African University of Science and Technology (AUST). Also, many thanks to David, Isima, Bolade, Buchi, Bidemi and all other staff at AUST.

I would also like to thank Prof. Eric Okoli, the Director Result Database of Ebonyi State University, and all the entire staff of the office for their support during my Master's program. I am equally grateful to the Deeper Life Campus Fellowship for keeping up spiritually: Pst. Gilbert and wife, Pst. Igwe, Bro Peter, Obinna, Ogonna and Monday, and Deeper Life Bible Church, Galadimawa District. Special thanks to my colleagues for their assistance in one way or the other: Monsur, Mubarak (roommate), Isa, Christian, Tanko, Yemisi, Wumi, Mubaraka, Nkiru, Barikisu, Safiyat and my partner Joy.

Finally, and very importantly, I acknowledge my sincere indebtedness and gratitude to my parents, Pst & Late Mrs Ifebude Joseph for their parental support and encouragement up to this point in life. I must not forget my dearest siblings for their love, support and sacrifice throughout my life. Words are not enough to express my appreciation for everything my family has done for me. Many thanks to my siblings: Serah, Elizabeth, Izu and Victor. This acknowledgement will not be complete without acknowledging my Fiancée Chikaodiri Amarachi Janet, for her love, patience, support, comments and encouragements which were indeed very valuable and crucial for successful completion of my degree. Thank you Beautiful.

DEDICATION

I dedicate this thesis to God Almighty, family and friends …

TABLE OF CONTENTS

1.0 INTRODUCTION 1
1.1 Context 1
1.2 Research Objectives 3
1.3 Approach Adopted 3
1.4 Organization of Work 3
2.0 LITERATURE REVIEW 4
2.1 System Control 4
2.1.1 Conventional PID Control 5
2.1.2 Model-Based Control 6
2.1.3 Artificial Intelligent Based Control 7
2.2 Traffic System Control 8
2.3 DEVS-Based System Control 9
3.0 SYSTEM ARCHITECTURE AND COMMUNICATION TECHNOLOGIES 11
3.1 Model-Based System Control Architecture 11
3.2 SURVEY OF TECHNOLOGY FOR DEVS-BASED SYSTEM CONTROL 12
3.2.1 Communication Between the Physical System and the Computer System 13
3.2.1.1 Radio Frequency Technology 14
3.2.1.2 GSM Technology 17
3.2.1.3 Internet Protocol (IP) Network 18
3.3 Summary of Communication Technologies 19
4.0 COMMUNICATION INSIDE THE COMPUTER 21
4.1 DISCRETE EVENT SYSTEM SPECIFICATION (DEVS) 21
4.1.1 Atomic DEVS Model 22
4.1.2 Coupled DEVS Model 23
4.1.3 DEVS SimStudio Simulation Package 24
4.2 Communication Between the Computer Ports and The DEVS Model 24
4.2.1 Serial/Parallel Port Communication API 25
4.2.1.1 Java Communication API (javax.comm) 25
4.2.1.2 RxTx Communication API (gnu.io.SerialPort) 25
4.2.1.3 Java Universal Serial Buss (USB) API (javax.usb) 26
4.2.1.4 Java Network API (java.net) 26
4.3 Summary of Java API for Communication 27

5.0 CONCLUSION 28
5.1 Summary of Work Done 28
5.2 Future Work 28

LIST OF FIGURES

Figure 1.1: Basic entities in M&S and their relationships 1
Figure 3.1: Model-Based Closed Loop System Control 11
Figure 3.2: Communication in a DEVS-based System Control 12
Figure 3.3: Types of Sensor Nodes (i) Wireless Sensor (ii) Wired Sensor 13
Figure 3.4: Electromagnetic Spectrum (GSMA, 2016) 15
Figure 3.5: RF-Based Communication System Architecture (i)Wired Sensors (ii) Wireless Sensors 16
Figure 3.6: SMS-Based Communication System Architecture 18
Figure 3.7: Summary of Technologies for Wireless Communication 19
Figure 4.1: Example of a Coupled Model 23
Figure 4.2: Communication Inside the Computer System 24

LIST OF TABLES

Table 2.1: Comparison of Control Methods 9

Table 3.1: Wireless Communication Technologies 17

TABLE OF ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
AUST	African University of Science and Technology
CGI	Common Gateway Interface
CMAES	Covariance matrix adaptation evolution strategy
DE	Differential evolution
DEVS	Discrete Event System Specification
DMC	Dynamics Matrix Control
EDGE	Enhanced Data rates for GSM Evolution or EGPRS
EF	Experimental Frame
EP	Evolutional programming
ETSI	European Telecommunication Standards Institute
GPRS	General Packet Radio Services
HIECON	Hierarchical Constraint Control
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
M&S	Modeling and simulation
MBPC	Model-Based Predictive Control
MDE	Model-Driven Engineering
MPC	Model Predictive Control
MPSO	Modified particle swarm optimization
NMI	Nelson Mandela Institute
OOP	Object Oriented Programming
PID	Proportion integral derivative
RCP	Rapid Control Prototyping
RF	Radio Frequency
RGA	Real coded genetic algorithm
SMCA	Set-point Multivariable Control Architecture
SMOC	Shell Multivariable Optimizing Controller
SMS	Short Message Service
SPP	Serial Port Profile
USB	Universal Serial Buss

CHAPTER 1

1.0 INTRODUCTION

This chapter presents an overview of the research context, the motivation behind this research and the corresponding aims and objectives. It begins with a brief introduction of modeling and simulation, i.e. Discrete Event System Specification (DEVS). This is a formalism for discrete event modeling and simulation, thereafter system control is introduced before presenting how it has evolved. Moreover, it provides information about the motivation for the research and the drive which steered the research goals. It is followed by a description of the research aims and objectives, then the approach adopted is presented. Finally, the outline of the rest of the thesis is revealed.

1.1 Context

Computational science (modeling and simulation) has become the third pillar of science along side theory and physical experiment (PITAC, 2005). Modeling and simulation (M&S) enable researchers to build and test models of complex real life systems. They do so without conducting physical experiments, or building and testing models of phenomena that cannot be replicated in the laboratory/physically. According to "The Theory of Modeling and Simulation" (Zeigler et al., 2000), there are four major important concepts of M&S. The concepts and relationship between them are shown in fig 1.

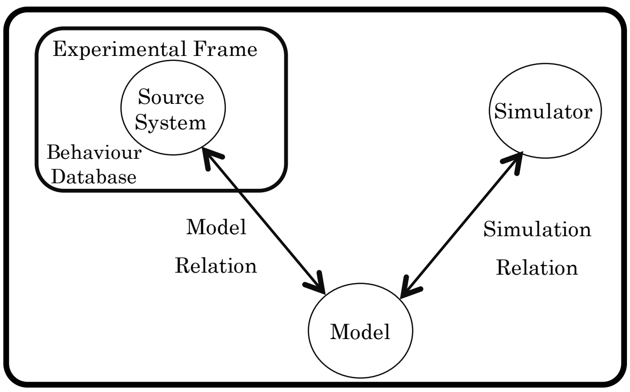


Figure 1.1: Basic entities in M&S and their relationships

a. **Source System:** Is a well-defined real or virtual environment that we are interested in. This system can be viewed as a source of observable data or a database of system behaviours.

b. **Experimental Frame (EF):** Is a specification of the condition under which the source system is observed or experimented with.
c. **Model:** Is any physical, mathematical or logical representation of a system. However, in M&S, a model is seen as a set of instructions, rules, equations or constraints used to generate input/output behaviours.
d. **Simulator:** Is any computational system (single processor, processor network, human mind or an algorithm) capable of executing a model to generate its behaviour.

For a complex dynamic system to be modeled, a robust formalism is needed. DEVS, a universal discrete event specification language (Zeigler, 1976) introduced by Zeigler in the early 70's is a theoretically well-defined formalism for modeling discrete event systems in a hierarchical and modular manner. In DEVS, each of the M&S components are well separated and a formal mechanism is used to describe each of the components. This modularity makes DEVS most suitable for the modeling of complex dynamic systems.

System Control is an application area of control theory, which is multidisciplinary in nature. Researchers began to clearly understand its principle in the year 1922; this was after a clear analysis of the control involved in position control was presented by Nicholas Minorsky (Benneth, 1996). System control has evolved from analogue to digital controls, from conventional to model-based to artificial intelligent based controls. In all forms of system control, a model of the physical system is either implicitly or explicitly used to determine the control input (Baskar et al., 2011). Most of the models are analytical models that require analytical solving, but solutions for some of these models (especially for non-linear systems) do not exist. To also note is that solving complex systems incurs a very high computational cost or is impossible. To overcome this, a simulation model, which always have a solution, is used.

Since DEVS can be used to model any physical system (discrete and continuous), and is a robust formalism with direct relation to a simulator, using DEVS model in system control is very much possible (Zeigler et al., 2000). We propose to use DEVS as the intelligence of physical devices. The DEVS model is used to generate the physical system's behaviour, and this information is remotely communicated to the physical system, using appropriate communication technology, while the model equally remotely gets feedback from the system. SimStudio, a Java-implemented DEVS simulator, will be used for the behaviour generation (simulation), while wireless communication will be used for the remote communication.

1.2 Research Objectives

The objective of this work is to propose an architecture for DEVS-Based System Control. We aim to define an architecture that establishes a remote connection between DEVS-model running on a computer and a physical scene. The physical scene receives commands from the computer and sends feedback to the computer. Survey of possible technologies to achieve DEVS-Based System Control was presented.

1.3 Approach Adopted

The approach taken in this thesis is to lay the foundation for the realization of DEVS-Based System Control. Communication between different components of the system, which is a fundamental problem to realizing the system control, is addressed. Survey of different technologies to solve this problem is presented.

1.4 Organization of Work

This work is organized as follows: in Chapter 2, we review different control methods: conventional control, Model-based control, intelligent control and DEVS-Based System Control. Chapter 3 presents different technologies that can be used to achieve communications in DEVS-Based System Control. Chapter 4 presents DEVS; DEVS Atomic Model, DEVS Coupled Model and DEVS Simulator (SimStudio) implemented in Java and Application Programming Interface (API) that can be used by DEVS model to communicate with the computer ports. Chapter 5 provides the summary of the work done and future works.

CHAPTER 2

2.0 LITERATURE REVIEW

Advancement in computing, software, and hardware technologies has led to several types of research into advancing system control methods in diverse areas of human endeavour. Much research was done in the era of "Early Control", before the pre-classical period. However, it was in the pre-classical era (1900 – 1940), that scientists and engineers began to understand the principles of system control. This was after Nicholas Minorsky, in 1922 presented a clear analysis of the control involved in position control systems and presented a control which is known today as the three-term or PID control (Benneth, 1996).

Accordingly, this literature review is grouped into three parts. In section 1, the state of the art in general system control is reviewed. In section 2, a review of model-based system control as related to traffic control is presented, to show an example of Model-based control. In section 3, the state of the art in DEVS-Based System Control are reviewed.

2.1 System Control

Despite the development of several control methods over the years, the challenges still encountered in the quest for efficient control are nonlinearities, time delays, multivariable interactions, disturbances, noise, unmeasured variables, time-varying parameters, and constraints (Edgar, 1987). System control methods can be classified into two, open-loop system control (without feedback) and closed-loop system control (feedback system).

A brief review of some of the control schemes using conventional PID control, Model-based control and artificial intelligent based control are presented here.

2.1.1 Conventional PID Control

Manual control has been the oldest form of control methods in control systems. It has proven to be too expensive, and ineffective in various control systems due to its reliance on human control experts (Samad & Annaswamy, 2011). However, the basic principle of manual control was used as a fundamental for building effective decision-making systems/supervisory control systems in industrial control systems (Kozak, 2014).

The conventional PID controller has been used since 1930's and is still the most widely used controller in industries because of its simplicity, robustness and ability to be used without adequate understanding of the process (Benneth, 1996).

PID control is a control loop feedback mechanism. The controller continuously measures the difference between the output of the controlled system and a set-point, whereby the input to the controlled system is tuned towards driving the output towards the set-point.

$$u(t) = K_p e(t) + K_i \int e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

Where

K_p, K_i and K_d are non-negative constants. p is present error value, i is the past error value while d is the future error value. u is the control signal and e is the control error.

$$e = measured\ variable - set\ point$$

However, the main problem of PID control is the tuning of the gain parameters to correctly satisfy the desired control process. In an attempt to solve the tuning problems of PID controllers which are commonly tuned using Ziegler-Nichols and Cohen-Coon methods, several recent methods have been proposed in literature to obtain optimal tuning of PID controller parameters. Such methods involved optimization using the neural network, fuzzy control, and evolutionary algorithms. Genetic search algorithm optimization technique was used to search for optimal PI/PID gains of a multivariable plant which has been decoupled into SISO subsystems (Yang, 2015). The results obtained showed better performance compared to Ziegler-Nichols approach. However, the method cannot guarantee that the gains obtained are optimal as compared to other optimization techniques. In (Nagaraj et al., 2008) and (Iruthayarajan & Baskar, 2009), several evolutionary and intelligent algorithms such as Differential Evolution (DE), real coded genetic algorithm (RGA), evolutional programming (EP), modified particle swarm optimization (MPSO), and Covariance Matrix Adaptation Evolution Strategy (CMAES) were used for optimal tuning of PID controller. The result obtained showed a better tuning method compared to other methods.

2.1.2 Model-Based Control

Despite the success of PID control, algorithm drive for higher efficiency and higher productivity while keeping costs low has opened new control methods to improve performance and simplicity (Kozak, 2014). To determine a sequence of control actions that are admissible which optimizes a performance function, future demands are considered while also making sure constraints are satisfied (Sussmann & Willems, 1997, Cassandras et al., 1998).

Optimal control or Model Predictive Control (MPC) are a class of computer control algorithm that explicitly utilize a mathematical model to predict future system/process behaviour.

In optimal control, the control problem is posed as an optimization problem which is based on the dynamical system model equations, subject to some constraints. Two basic approaches are used in solving optimal control problems: dynamic programming and calculus of variations (Baskar et al., 2011). The main drawback of optimal control is that the method is an open-loop control approach and thus suffers from disturbances and model mismatch errors. We will review the Model Predictive Control next. This is essentially a closed-looped control method, feedback from the system is used to overcome disturbances while a receding horizon is used to look ahead into the future.

Model Predictive Control is an advanced control technique which has made a huge impact in the control of industrial processes domain (Maciejowski, 2002). It is the second most popular control technique in industry. An early description of MPC approach known as Identification and Command [IDCOM] was presented in the paper, "Model predictive heuristic control: applications to industrial processes" (Rachika, 2013). In the paper, discrete time finite impulse response was used to define the relationship between the input and output process. Other acronyms used to describe the algorithm of MPC includes, Dynamics Matrix Control algorithm (DMC), Quadratic programming solution of dynamic matrix control (QDMC), Hierarchical Constraint Control (HIECON), Shell Multivariable Optimizing Controller (SMOC), Set-point Multivariable Control Architecture (SMCA), and Model-Based Predictive Control (MBPC). Although the details of each algorithm differ, the main idea is still based on the concept of a receding horizon. However, all MPC control strategies are based on the assumption of a model which has to be quite accurate, but real world problem modeling which are non-linear, and time-varying often present problems because MPC is a standard linear model-based algorithm (Rachika, 2013).

2.1.3 Artificial Intelligent Based Control

Due to complexities, disturbances and the highly non-linear process of real systems, and the need for high performance and robust controllers, intelligent controller over the years has been advocated to solve the challenging problems of conventional PID controller and Model-based control. The main advantage of using artificial intelligent system – is based on high machine intelligent quotient of the systems, tractability and capability to generate acceptable solutions under uncertainties, and data imprecision (Mahfouf et al., 2001). While the artificial neural network is used to find the optimum combination of weights and activation function to fit the measured data that describes the process, fuzzy system exploits its ability to use intuitive human thinking to obtain a reasonable model of an ill-defined process.

In summary, all the reviewed control approaches have the same architecture, but the major differences lie in the way through which the control signal is generated. For the conventional PID controller, the control signal is generated by summing the three terms together (proportional, integral and derivative). In the model-based system control, the control signal is generated by solving an optimization problem which is based on an explicit model of the system. Finally, in intelligent system control, the control signal is generated by Artificial Intelligence methods like case-based reasoning, fuzzy-logic or neural networks.

2.2 Traffic System Control

Almost all the system control methods discussed in section 2.1 are applied in the industrial system control. Some of them have been modified and used in traffic system control, and control of other physical systems. The main problem of the traffic system is congestion. Several researches have focused on congestion control, congestion avoidance and congestion management, while others have focused on the area of route guidance, ramp metering, speed control and traffic light control.

The control method used to determine optimal speed and optimal release time at on-ramp, for vehicles in intelligent vehicle highway systems was Model Predictive Control (Baskar et al., 2008). The result showed that the approach improved traffic performance by 26.16%, in comparison to the uncontrolled speed and uncontrolled on-ramp release time. Baskar et al (2011) carried out a survey on different control methodologies as it applies to ramp metering. The result (Table 1) presented, showed that conventional control method (static feedback control) is not predictive and is localised with low computational. Model-based control methods (optimal control and MPC) are predictive and scalable but have high computational complexity, while the advance control methods (fuzzy-logic, rule-based system and ANN) have medium computational complexity but not scalable, and are not predictive (Baskar et al., 2011).

Table 2.1: Comparison of Control Methods

Control Method	Computational Complexity	Constraints (hard)	Future Inputs	Model-based	Scalability
Static Feedback	Low	No	No	Not explicitly	Localised
Optimal Control and MPC	High	Yes	Yes	Model-based	System-wide
AI-based	Medium	No	No	Not explicitly	Localised

2.3 DEVS-Based System Control

All the control methods already discussed above are mostly applicable to continuous systems which are majorly sample-timed control against event-based control. DEVS formalism introduced in 1976 by Zeigler, provides a means for specifying systems as mathematical objects and simulation of such objects (Zeigler, 1976). DEVS formalism is to event-based control, as differential and difference equation, is to conventional control. In (Chi & Zeigler, 1990) DEVS was used to describe the development of event-based intelligent control systems. Schwatinski et al (2010) showed that PDEVS and RT-DEVS can be integrated together with the concept of Rapid Control Prototyping (RCP) to control a system in real-time.

DEVS-Based System Control is even-based therefore is reactive. In this thesis we will explore how to use a DEVS model to run fast simulations and then be able to provide proactive system control in real-time.

In summary, we have seen different methods of system control starting from, classical methods to the advanced methods and also DEVS-Based System Control methods. This thesis will be focusing on the use of DEVS simulation to proactively control a physical system in real-time.

CHAPTER 3

3.0 SYSTEM ARCHITECTURE AND COMMUNICATION TECHNOLOGIES

This chapter presents the conceptual system architecture of Model-Based system control and afterwards presents technologies that can be used to realize communication in Model-Based system control. Finally, a summary of all the technologies is presented.

3.1 Model-Based System Control Architecture

The proposed Model-Based system control architecture will be made up of a closed-loop system control. Where the physical system is the system being controlled and a model-based controller that controls the physical system. Since at the heart of the controller is a model of the physical system, the model can be checked by fast simulation for optimal condition, and then the result of the simulation will be used to determine the control input (commands) to the physical system.

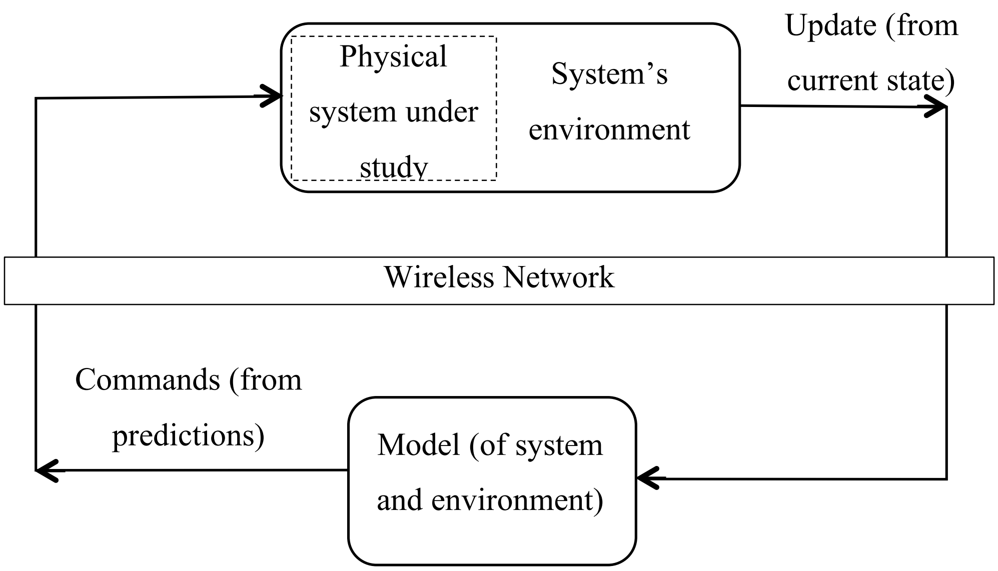


Figure 3.1: Model-Based Closed-Loop System Control

The output from the physical system (current system state) will be used by the model to run fast simulations, just like in the conventional system control, the result of the simulation will determine the control command that will be sent as input to the physical system. This control loop continues until the control goal is achieved. This control approach belongs to the Model-based control method.

3.2 SURVEY OF TECHNOLOGY FOR DEVS-BASED SYSTEM CONTROL

In order to achieve DEVS-Based System Control, we identified a two-phase communication (see Figure 3.2) problem which must be solved for the system to be realized. The first phase of the communication problem is to, define how the computer interacts with the physical system, while the second phase is to, define how the DEVS model running on the computer receives data and transmits control commands. In this section, we will be focusing on the technologies used to solve the first phase of the communication problem.

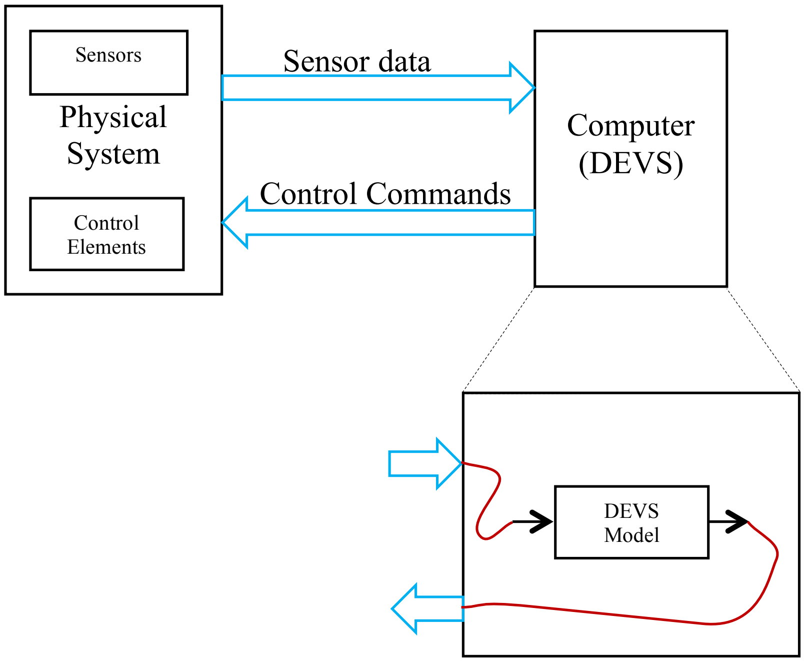


Figure 3.2: Communication in a DEVS-based System Control

The DEVS model requires estimation of the current state of the physical system to run a fast simulation based on the current state of the system, not based on some randomly generated data. To realize this, data needs to be collected in real-time, i.e. the data collecting point is connected to the DEVS model. Here, the data collecting point is a sensing device (sensor), while a sensor is a transducer that transforms the physical parameters to an electronic signal (Nellore & Hancke, 2016). Sensors are used for:

a. Detecting,
b. Identifying,
c. Counting,
d. Measuring and
e. Monitoring.

The area of deployment for the DEVS-based System Control and also the type of physical parameter being sensed, will determine the type of sensor to be used. There are wide varieties of sensors, but we classified our sensors into, (i) Wireless Sensors and (ii) Wired Sensors (see Figure 3.3). The wireless sensor is made up of a sensing module, processing & storage module, power module and the radio module. A wireless sensor node has all it takes to sense, process and communicate with a computer wirelessly through its radio module. While the wired sensor has only a sensing module, without a well-defined digital interface for communication.

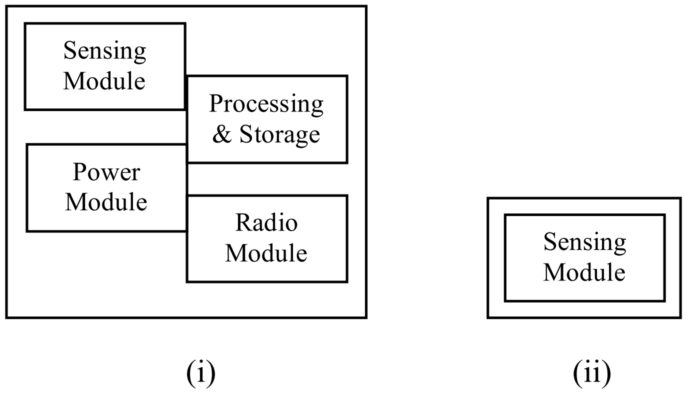


Figure 3.3: Types of Sensor Nodes (i) Wireless Sensor (ii) Wired Sensor

Most control elements already have a defined/established means of receiving control commands. We are now faced with the challenge of defining an appropriate interface for communication between the computer and the control element. According to Baskar et al. (2011), control elements can be classified into (i) Manual System (ii) Automated System. The type of the control element will determine how control commands will be received and implemented.

A manual system receives control commands through voice commands or visual display; here the control commands are interpreted by the human controller and implemented by the human. In an automated system, the system receives control commands through a digital interface, interprets the commands and take necessary action independent of human interaction.

3.2.1 Communication Between the Physical System and the Computer System

Communication between the physical system and the computer system is classified into two categories based on the means of transmission. One is wired communication, and the other is wireless communication. The simplest way of connecting a physical device to the computer system is through wired means (Burgess, 2015). The wired connection can be directly from the device to the computer, or through the wired Ethernet. The Ethernet connection might connect to wired Internet service. These were the methods used before the development of the radio links in the 1980s.

The problem with wired type of communication is that it will not be suitable for DEVS-Based System Control because:

i. Some of the physical elements are mobile.
ii. Physical elements are spread across a wide geographical region.

Based on the above reasons, running cables to connect mobile devices is not appropriate and for a wide geographical region, it is very expensive.

Due to the disadvantages of wired communication between devices, in the rest of this section, we concentrated on wireless means of communication between the physical system and the computer system. According to literature, the following technologies have been used as communication links between the computer system and physical system: Radio Frequency, GSM, and the Internet. We now discuss each of these technologies in details.

3.2.1.1 Radio Frequency Technology

Radio Frequency (RF) is any electromagnetic wave frequency that lies between 3 KHz to 30 GHz used for wireless communication and radar signal. Traditionally, RF defined frequencies as from a few KHz to approximately 1 GHz. Considering higher frequency microwaves extends the range to 30 GHz (see Figure 3.4). RF is the de facto means through which wireless communication is realized. RF is used for both short-range and long-range communications.

Since RF carries data from one device to another device over the air, it has been used as a medium of communication in industrial remote device controls, smart sensor applications, and smart home automation (Jivani, 2014).

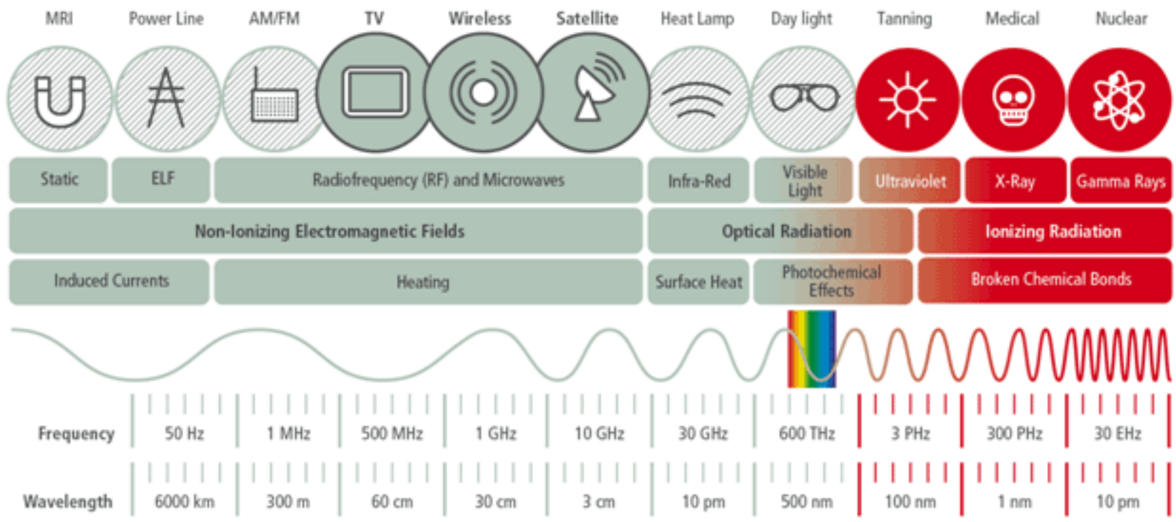
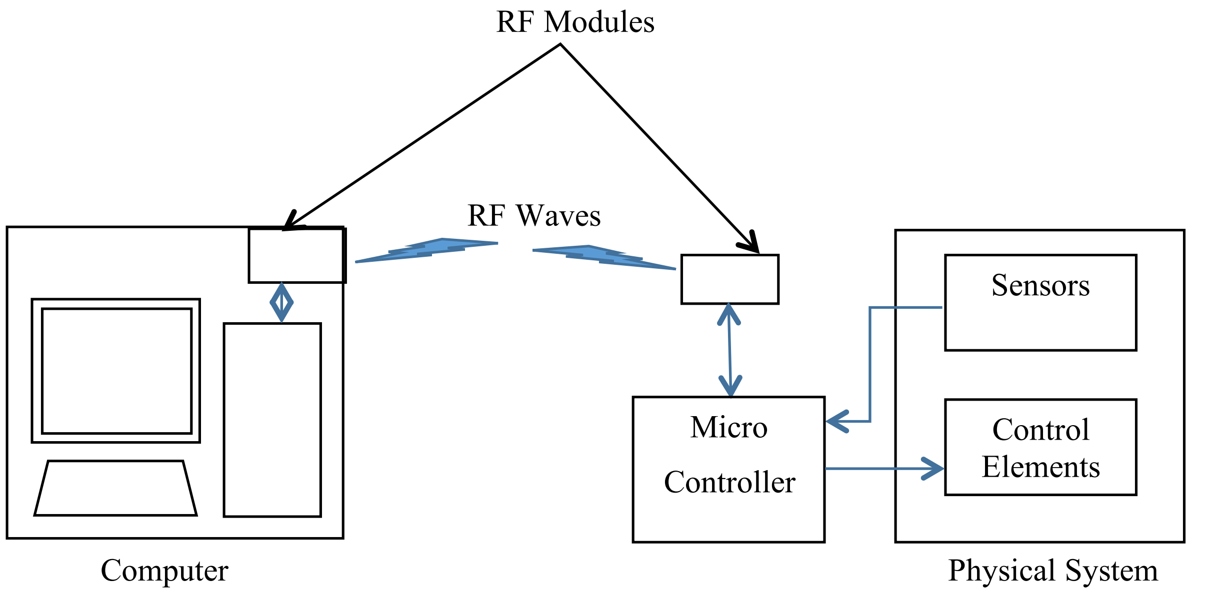


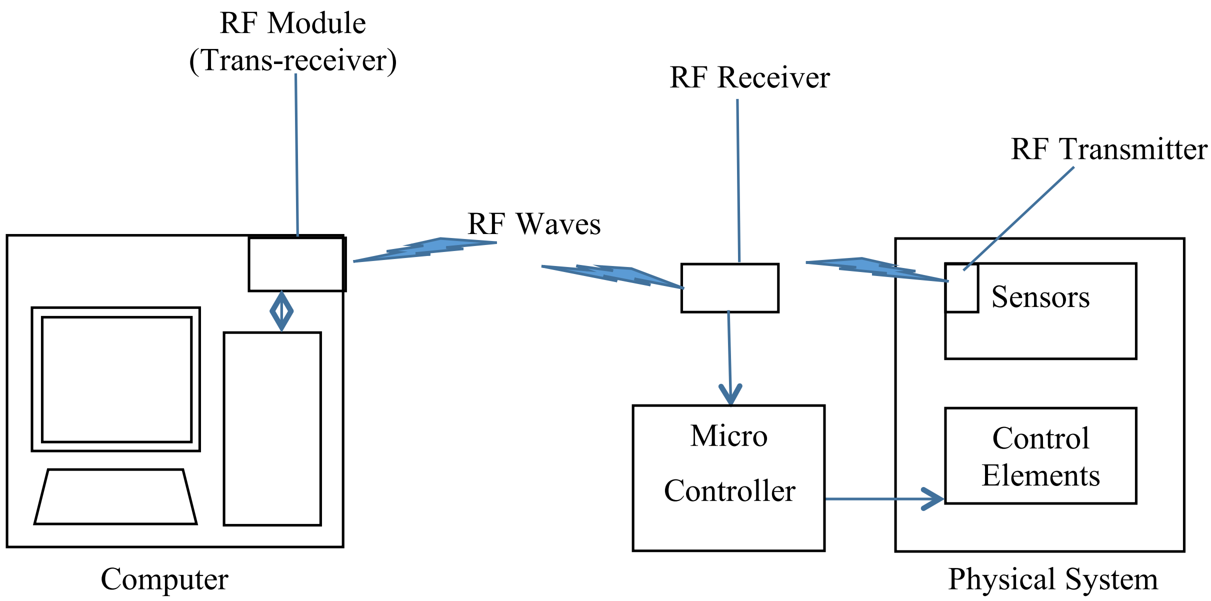
Figure 3.4: Electromagnetic Spectrum (GSMA, 2016)

RF module is a hardware device that is used to transmit and/or receive radio signals between two devices. There are regulations both nationally and internationally governing the use of RF for communication; this has placed restriction on the use of some frequencies. Therefore, RF modules are commonly designed to operate in commercially available Industrial, Scientific and Medical (ISM) radio bands which are 433MHz, 900MHz, and 2.4GHz frequencies.

Based on the type of sensors used, two types of system architectures can be used (see Figure 2.5). In the first architecture (i) Wired sensors, the sensors do not have the capability to communicate directly with the computer, therefore, a microcontroller will be used as the interface for communication where the microcontroller is connected to an RF module. While on the other hand, the sensors communicate directly with the computer. In this case, the sensor must also possess some processing power for encoding the data in a format that can be transmitted over the RF channel.



(i)



(ii)

Figure 2.5: RF-Based Communication System Architecture (i)Wired Sensors (ii) Wireless Sensors

For RF modules to be compatible across different manufacturers, the modules are designed to comply with standard protocols for communication. Non-proprietary protocols are WiFi, Bluetooth, ZigBee, UWB, WiMax. Table 3.1 shows a summary of non-proprietary protocols, standard, frequency in which they operate and the range of communication.

Table 3.1: Wireless Communication Technologies

Technologies	**Standard**	**Frequency**	**Range**
WiFi	IEEE 802.11	5.8 GHz 2.4 GHz	<100 m
Bluetooth	IEEE 802.15.1	2.4 GHz	10 m - 100 m
UWB	IEEE 802.15.3	3.1 – 10.6 GHz	< 10 m
ZigBee	IEEE 802.15.4	2.4 GHz	<75 m
WiMAX	IEEE 802.16	2 – 11 GHz	<10 km

3.2.1.2 GSM Technology

GSM is a standard protocol developed for second-generation (2G) digital cellular networks by the European Telecommunication Standards Institute (ETSI). This standard was later on expanded over time to include data communications, first by circuit-switched transport, then packet-data transport through General Packet Radio Services (GPRS) and Enhanced Data rates for GSM Evolution (EDGE or EGPRS). Subsequently, the 3GPP developed third generation (3G) UMTS standards followed by fourth generation (4G) LTE Advanced standards, which are not part of the ETSI GSM standard (Vineeth, 2014).

GSM has grown from a standard used for just voice calling, to a standard that can be used to send short messages and even for data communication. Short Message Service (SMS) has taken a front stage as a means of wireless communication. SMS is a method of communication where texts are exchanged between mobile phones, or from a PC to another device. SMS also serves as a communication channel for monitory, supervisory and control in SMART home automation (Bai et al., 2012). In Jivani (2014) SMS was used to control devices remotely by sending AT commands to switch different devices on/off. Also in the same paper devices were also monitored by sending device status to a mobile device via SMS.

SMS by default is provided by GSM network providers currently. For SMS to be exchanged between a PC and another device (non-mobile phones) a GSM module is required.

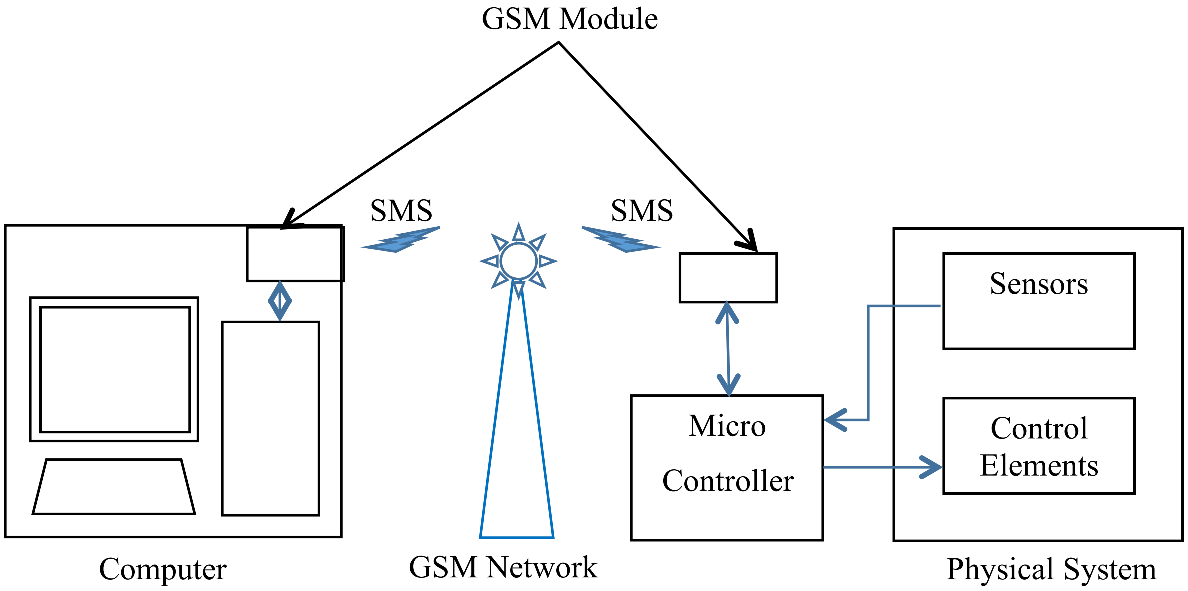


Figure 3.5: SMS-Based Communication System Architecture

A GSM module is a device with the capability of emulating a mobile phone (i.e., it can make/receive calls and also send/receive SMS). A GSM module uses a SIM card and requires the services of a mobile network provider just like a mobile phone. This module is connected to a PC and a device between which, an SMS will be exchanged. Communication between the PC-GSM module and device-GSM module is usually through a serial port (RS232). Since most devices (sensors and control elements) cannot communicate directly with a GSM module, a microcontroller is used to interface between the devices and the GSM module as shown in system architecture Figure 3..

3.2.1.3 Internet Protocol (IP) Network

Another means for wireless communication is through the IP network. Here, every device communicating over the network possesses a unique IP address for communication. The model of communication is based on a server/client mode of communication. The use of the Internet has exploded over the last decade. Today the Internet is no more primarily used for emailing, Web surfing, stock trading, online shopping, etc. Recently several researchers have used the Internet to control physical systems in an open-loop arrangement, meaning that the reference and not the control signal propagates through the Internet. Recently, researchers, have tried to establish feedback control through the Internet, which requires transmitting the control signal through the network. This exposes the system control loop to delays of varying time inherent to a packet switched network. Time delay, however, occurs in every electro-mechanical system. In most cases the time delay is so minimal that it is not noticeable, but in other cases, it can render the system unstable as in the case of control over the Internet. Different techniques have been tried to compensate for this effect, such as a time forward observer developed for a supervisory control over the Internet and a position-based force-feedback scheme (Saghir & Wayne, 2002).

Internet interface between a computer and a remote device can be realized following two different approaches. In the first approach, a Web server and a Web browser are used for Internet interfacing. Control commands are sent to the remote device via Common Gateway Interface (CGI). The other one consists of a server program and a client program implemented by C or C++ language. Users utilize the client program to control the remote device. The server program receives control commands from the client program and controls the remote device directly as per the command.

3.3 Summary of Communication Technologies

Three technologies were presented in this chapter for realizing communication in a DEVS-Based System Control; Radio Frequency technologies, GSM technologies, and Internet technologies.

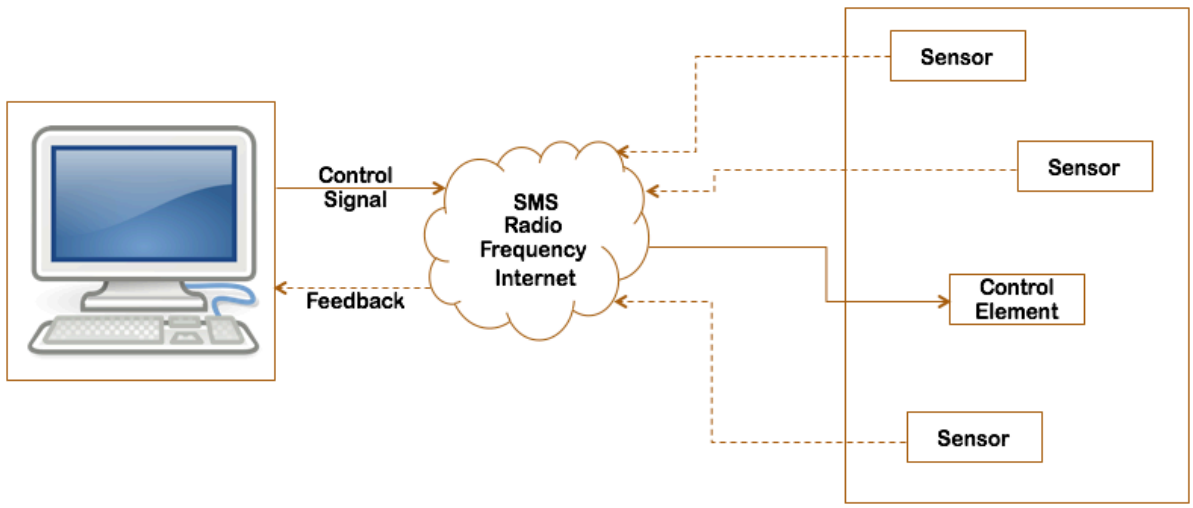


Figure 3.6: Summary of Technologies for Wireless Communication

Radio Frequency communication is best used in remote areas without reach of GSM cellular network and the Internet. RF is also suitable for data transmission that requires high bandwidth and low latency. Some RF protocols enable the creation of a wireless network (e.g.. ZigBee), this eliminates the need for wireless routers because every node also acts as a router. The problem or disadvantage of RF is majorly interference, especially if non-licensed and commercial frequency bands are being used. Also, the distance of coverage is limited to a few hundred meters except for higher frequencies where microwave signals can cover few kilometres but are unidirectional and operate in line of sight.

The coverage of cellular networks is increasing tremendously while the cost of using the network is reducing. These factors made the use of SMS as a communication channel for devices to communicate wirelessly more economical. SMS is used where low bandwidth is required and covers a very long distance based on the coverage of the network provider. Using SMS as a communication channel requires the purchase of additional hardware (GSM module/modem) which makes it somewhat expensive to achieve.

Using RF and SMS requires hardware drivers to be installed on the computer system for the computer to be able to communicate with the hardware. Using the Internet eliminates the need for device drivers and simply transmits data using already existing TCP/IP protocol. Using the Internet also removes the barrier of hardware compatibility issues because, once the devices are connected to the Internet they can communicate. Internet connection can be established through WiFi or GSM network. One major problem with using the Internet as a means of communication is latency, i.e. the time it takes data to travel through the network. High latency can render a system control unit useless.

CHAPTER 4

4.0 COMMUNICATION INSIDE THE COMPUTER

In this chapter, we briefly explored DEVS models (atomic and coupled models), and DEVS simulation, SimStudio. This is an architecture built on DEVS formalism for integrating Modeling and Simulation tools. Finally we discussed how DEVS model interacts with the computer communication ports.

4.1 DISCRETE EVENT SYSTEM SPECIFICATION (DEVS)

DEVS is a hierarchical and modular formalism for describing discrete event systems introduced by Zeigler in 1976. DEVS allows description of system behaviour at two levels. At the lower level, an atomic DEVS model describes the individual behaviour of a system as, a sequence of deterministic transitions between sequential states as well as how it reacts to the external input (events) and how it generates output (events). At the higher level, a coupled DEVS describes the structure of the system as a network of coupled components. The components can be atomic DEVS models or coupled DEVS models. This concept of coupling atomic DEVS models or coupled DEVS models establishes the fact of hierarchical modeling in DEVS formalism (Concepcion & Zeigler, 1988).

DEVS, is a theoretically well-defined system formalism that specifies a system as a mathematical expression based on set theory. DEVS offers a platform for modeling and simulation of complex and dynamic systems in different domains. DEVS models are seen as black boxes which have input and output ports through which system structure and behaviour are described. The formalism clearly differentiates the modeling and simulation approaches, modeling and simulation are linked together by a DEVS simulator. A DEVS simulator is used to reproduce behaviour that is identical to that of the physical system or environment being observed. DEVS also provides a generic mechanism as well as mechanism to mix formalism of different types (Adegoke et al., 2013).

Zeigler originally proposed DEVS which is now known as Classic DEVS (C-DEVS), C-DEVS was based on the sequential operation of early computers, and this hindered the exploitation of parallelism in today's computer. To overcome the challenges of C-DEVS, Parallel DEVS (P-

DEVS) was introduced (Zeigler et al., 2000). They had the concept of bags to accommodate multiple input messages and a confluent function to handle simultaneous internal, and external events.

4.1.1 Atomic DEVS Model

The atomic DEVS model is a structure describing the different aspects of the discrete event behaviour of a system:

$$\boldsymbol{AtomicModel} = < \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{S}, \boldsymbol{\delta_{int}}, \boldsymbol{\delta_{ext}}, \boldsymbol{\lambda}, \boldsymbol{ta} >$$

Where

$X = \{(p, v) \mid p \in InPorts, v \in X_p\}$ is the set of inputs events where $InPorts$ is the set of input ports and X_p is the set of input values.

$Y = \{(p, v) \mid p \in OutPorts, v \in Y_p\}$ is the set of output events where $OutPorts$ is the set of output ports and Y_p is the set of output values.

$S =$ Set of states

$\delta_{int}: S \rightarrow S$ is the internal transition function

$\delta_{ext}: Q \times X \rightarrow S$ is the external transition function

$where\ Q = \{(s, e) \mid s \in S, e \in [0, ta(S)]\}$ is the total state set

$\lambda: S \rightarrow Y$ is the output function

$ta: S \rightarrow \mathbb{R}^{+}_{0,+\infty}$ is the time-advance function

A DEVS model always remains in a state $s \in S$ until a transition function δ_{int} or δ_{ext} causes the model to change to another state. The model stays in this state until the elapsing of its lifetime of $ta(s)$ in the absence of external events. Outputs value $y \in Y$ of a model is sent to the output port when the lifetime of the model elapses using the output function $\lambda(s)$ (this is the only way output is generated from the model). The time-advance function for a state $ta(s)$ can take a real value between 0 and $+\infty$. If $ta(s) = 0$ immediate transition is triggered, such state is transient. On the other hand if $ta(s) = +\infty$ the model remains in this state until an external event occurs, such a state is said to be a passive state.

4.1.2 Coupled DEVS Model

The couple DEVS model is a structure describing the network of several atomic or coupled sub-models:

$$\boldsymbol{CoupledModel} = < \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{D}, \{\boldsymbol{M_d} \mid \boldsymbol{d} \in \boldsymbol{D}\}, \boldsymbol{EIC}, \boldsymbol{EOC}, \boldsymbol{IC}, \boldsymbol{select} >$$

Where

$X = \{(p, v) \mid p \in InPorts, v \in X_p\}$ is the set of inputs ports and values.

$Y = \{(p, v) \mid p \in OutPorts, v \in Y_p\}$ is the set of output ports and values.

$D =$ Set of the component or model names

$M_d =$ DEVS model $\forall d \in D$

$EIC =$ set of external input couplings which connects external inputs to components inputs

$EOC =$ set of external output couplings which connects components output to the external output

$IC =$ set of internal couplings which connects component output to component inputs

$Select: 2^D \rightarrow D$ is the tie breaking function used to select a model to execute in case of simultaneous internal events.

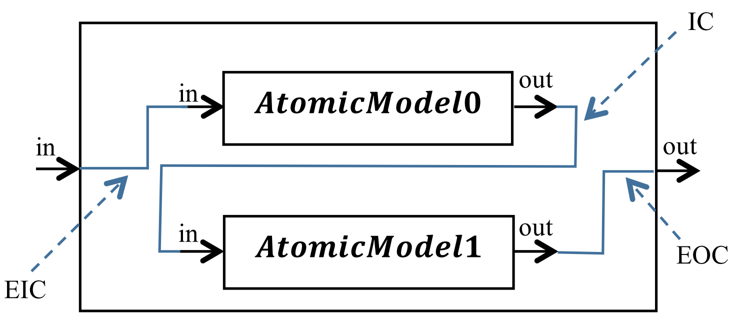


Figure 4.1: Example of a Coupled Model

Coupled DEVs model groups two or more models into a composite model that can be seen as a new DEVS model due to the coupling property. The closure property guarantees that the coupling of the several class instances results in a model of the same class which allows the hierarchical model.

4.1.3 DEVS SimStudio Simulation Package

SimStudio (Traore, 2008) is an architecture built upon the DEVS formalism that aims at integrating tools for modeling and simulation, analysis and collaboration through Model-Driven Engineering (MDE). The SimStudio simulation package is a component of the SimStudio, this is an Object Oriented Implement of simulation algorithm for DEVS implemented in Java.

4.2 Communication Between the Computer Ports and The DEVS Model

This section deals with technologies that can be used to establish communication between the computer ports and the DEVS model. Communication between the computer ports and the DEVS model will be realized through an API. The type of communication channel used between the physical system and the computer system will determine the API that will be used to read/write to the computer port.

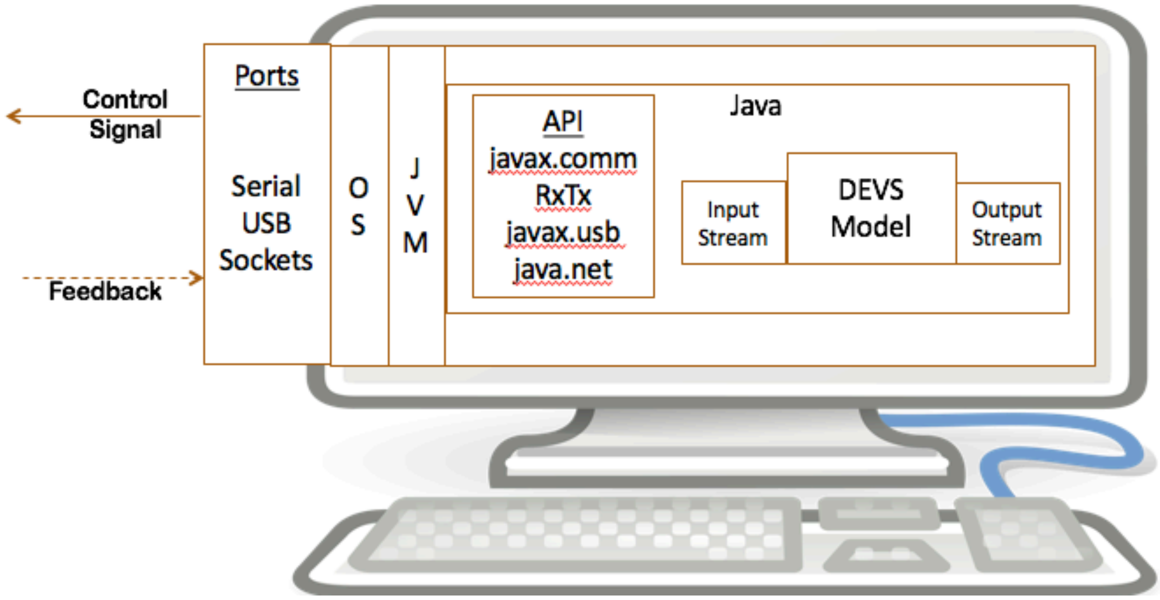


Figure 4.2: Communication Inside the Computer System

Since SimStudio is the simulation package that will be used to implement and simulate the DEVS model is written in Java, we will be concentrating on the APIs that are available in Java programming language for communication. Java is an Object Oriented Programming (OOP) language that is used to build platform-independent applications. It has an API for communicating with devices through the serial or parallel port (JavaComm API also known as javax.comm) and also an API for network communication (java.net) (Oracle, 2016). We will be discussing these different APIs in detail in the sections to follow.

4.2.1 Serial/Parallel Port Communication API

4.2.1.1 Java Communication API (javax.comm)

The Java Communications 3.0 API is a Java extension that enables developing communication applications that are platform-independent for technologies such as smart home devices, embedded systems, point-of-sale devices, financial services devices, fax, modems, display terminals, and robotic equipment. The Java Communications API (also known as javax.comm) provides applications access to Recommended Standard (RS)-232 hardware (serial ports) and limited access to IEEE-1284 (parallel ports), Serial Port Profile (SPP) mode. SPP emulates a serial cable to provide a wireless substitute for existing RS-232 standard.

According to (Oracle, 2016) Java Communication API has the following features:

- Enumeration of ports (administrator and user configurable port mapping)
- Port configuration (baud rate, speed, stop bits, parity)
- Access to EIA232 standard DTR, CD, CTS, RTS and DSR signals
- Transfer of data over RS-232 ports
- Hardware and software flow-control options
- Receive-buffer threshold control
- Asynchronous event option for notification of:
    - Data available on an RS-232 port
    - Port hardware line level changes
    - Port ownership changes within a single JVM

Lack of technical support and also a lack of support for some platforms has made javax.comm not to always be the API of choice for serial communication when developing applications that communicate with serial devices. RxTx Serial Communication API was developed to overcome this challenges. In the next section, we will discuss RxTx Serial Communication API.

4.2.1.2 RxTx Communication API (gnu.io.SerialPort)

RxTx Serial Communication API is a free-software library which is available for some platforms, not only Linux. It can be used in conjunction with JavaComm (RxTx providing the hardware-specific drivers), or it can be used stand-alone. When used as a JavaComm driver the bridging between the JavaComm API and RxTx is done by JCL (JavaComm for Linux). JCL is part of the RxTx distribution.

When RxTx is not used as a JavaComm driver, it provides a richer interface, but one which is not standardized. RxTx supports more platforms than the existing javax.comm implementations. Recently, RxTx has been adopted to provide the same interface as JavaComm, only that the package names are different from the JavaComm packages.

4.2.1.3 Java Universal Serial Buss (USB) API (javax.usb)

Java USB API (javax.usb) is a Java API that gives full access to Universal Serial Buss (USB) from Java on any platform that supports USB 1.1 or later specification. USB is a tree based, dynamic, expandable, plug-and-play, high bandwidth, and powered bus, unlike the simple serial ports. For each USB platform, the Java Virtual Machine will need to implement the javax.usb specification by adding some native code that plugs in javax.usb and binds it to the host OS USB driver architecture.

USB is a more complex and feature rich bus than standard serial/parallel ports. Java USB API has the following features:

- Supports the dynamic nature of USB
- Supports multiplex operations
- Attached devices form a tree instead of a list
- Provides access to transport data and communication signals

4.2.1.4 Java Network API (java.net)

The Java network API (java.net) provides two major classes for implementing networking application; Low-Level API and the High-Level API.

The Low-Level API deals with the following abstractions:

- Addresses, which are networking identifiers, like IP addresses
- Sockets, which are basic bidirectional data communication mechanisms
- Interfaces, which describe network interfaces

The High-Level API, deals with the following abstractions:

- URIs which represent Universal Resource Identifiers

- URLs which represent Universal Resource Locators
- Connections, which represent a connection to the resource pointed to by URLs.

4.3 Summary of Java API for Communication

JavaComm API is the default API provided by Sun for Java to communicate with the serial/parallel port. Lack of support for the API by Sun led to the development of RxTx Serial Communication API, which is a free-software library for serial communication in Java. JavaComm lacks adequate support, while RxTx lacks documentation. If the serial port is to be used for communication, either JavaComm (javax.comm) or RxTx Serial Communication API can be used by Java or a proprietary SerialIO, which has much support.

If the USB is used for communication, javax.usb API must be used by Java to establish communication between the device and the computer. USB which also is an advance type of serial port can also be accessed through the javax.comm API but its full functionality and capability will not be available.

For network communication, the javax.net API is used to read/write to the network. Devices communicating over the network use the client/server mode of communication. This implies that one of the devices needs to be the server, while the other is the client for communication to be established.

Once communication is established between the Java Virtual Machine and the hardware via the underlying OS using any of the Ports/API above, reading from or writing to the port is done through the InputStream or OutputStream respectively.

CHAPTER 5

5.0 CONCLUSION

This chapter gives a summary of the research work presented in this thesis. It highlights the conclusions and gives an evaluation of the achievements of the thesis, and provides recommendations for future work.

5.1 Summary of Work Done

DEVS is a formalism used to describe discrete event systems in a hierarchical and modular manner. DEVS-Based System Control is a proposed concept that uses DEVS model of a physical system to, run fast simulation and determine the best control action at any given point in time. The architecture of the system is made up of sensors, the physical system, control elements (actuators), DEVS model and wireless communication network. Communication between the components of the system is identified as a very crucial issue that must be solved for DEVS-Based System Control to be realized.

The communication issue is two-fold: first communication between the physical elements and the computer system, then secondly the communication inside the computer system between the computer ports and the DEVS model. Technologies that can be used to solve the first phase of the communication problem were presented, while different API that can be used by Java to communicate with computer ports was also presented to address the second phase of the problem.

5.2 Future Work

Future research on DEVS-Based System Control needs to move on to the next stage of implementation. Work needs to be done in setting up a physical system and trying to establish communication between, the physical system and the computer system. Also, trying out the use of the different stated technologies to evaluate their performance based monitoring and control operations should be done. Finally, a DEVS model that reads/writes to the computer port needs to be implemented for DEVS-Based System Control to be realized.

REFERENCES

Adegoke, A., Togo, H., & Traore, M. K. (2013). A Unifying Framework for Specifying DEVS Paralle and Distributed Simulaiton Architectures. *Simulation: Trans. of the Soc. of Modeling and Simulaiton Int'l, 89* (11), 1293-1309.

Bai, F., Beg, S., & Khan, M. F. (2012). Controlling Home Appliances Remotely through Voice Command. *International Journal of Computer Applications, 48* (17), 1-4.

Baskar, L. D., Schutter, B. D., & Hellendoorn, H. (2008). Dynamic Speed Limits and On-Ramp Metering for IVHS using Model Predictive Control. *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems*, (pp. 821 - 826). Beijing.

Baskar, L. D., Schutter, B., Hellendoorn, J., & Papp, Z. (2011). Traffic Control and Intelligent Vehicle Highway Systems: A Survey. *IET Intelligent Transport Systems, 5* (1), 38-52.

Baskar, L. D., Schutter, B., Hellendoorn, J., & Papp, Z. (2011). Traffic Control and Intelligent Vehicle Highway Systems: A Survey. *IET Intelligent Transport Systems, 5* (1), 38-52.

Benneth, S. (1996, June). A Brief History of Automatic Control. *IEEE Control Systems*, 17 - 25.

Burgess, L. (2015, October 13). *How Does Sensor Data Go From Device to Cloud.* Retrieved May 5, 2016, from ReadWrite: http://readwrite.com/2015/10/13/sensor-data-device-to-cloud/

Cassandras, C. G., Pepyne, D. L., & Wardi, Y. (1998). Optimal control of systems with time-driven and event- driven dynamics. *In Proceedings of the 37th IEEE Conference on Decision and Control*, (pp. 7 - 12). Florida.

Chi, S.-D., & Zeigler, B. P. (1990). DEVS-Based Intelligent Control of Space Adapted Fluid Mixing. *Conference on Artificial Intelligence for Space Applications*, (pp. 25 - 32).

Concepcion, A. I., & Zeigler, B. F. (1988). DEVS Formalism: A Framework for Hierarchical Model Development. *IEEE Transaction on Software Engineering, 14* (2), 222-241.

Edgar, T. F. (1987). *Current Problems in Process Control.* Retrieved March 19, 2016, from http://ieeecss.org/CSM/library/1987/april1987/w13-15.pdf

GSMA. (2016, May 17). *Electromagnetic spectrum.* Retrieved May 17, 2016, from GSMA: http://www.gsma.com/publicpolicy/faq/electromagnetic-spectrum

Iruthayarajan, M. W., & Baskar, S. (2009). Evolutionary algorithms based design of multivariable PID controller. *Expert Systems with Applications, 36*, 9159 - 9167.

Jivani, M. N. (2014). GSM Based Home Automation System Using App-Inventor for Android Mobile Phone. *Int'l J. of Adv. Research in Electrical, Electronics and Instrumentation Engineering, 3* (9), 12121-12128.

Kozak, S. (2014). State-of-the-art in Control Engineering. *Journal of Electrical Systems and Information Technology, 1* (2014), 1-9.

Maciejowski, J. M. (2002). *Predictive Control with Constraints.* London: Prentice-Hall.

Mahfouf, M., Kandiah, S., & Linkens, D. A. (2001). Adaptive estimation for fuzzy TSK model-based predictive control. *Transactions of the Institute of Measurement and Control, 23* (1), 31 - 50.

Nagaraj, B., Rampriya, B., & Subha, S. (2008). Tuning Algorithms for PID Controller Using Soft Computing Techniques. *International Journal of Computer Science and Network Security, 8* (4), 278 - 281.

Nellore, K., & Hancke, G. P. (2016). A Survey on Urban Traffic Management System Using Wireless Sensor Networks. *Sensors, 16* (157), 1-25.

Oracle. (2016). *Java Communications API*. Retrieved May 23, 2016, from oracle.com: http://www.oracle.com/technetwork/java/index-jsp-141752.html

President's Information Advisory Committee. (2005). *Computational Science: Ensuring American's Competiveness.* Virginia: National Coordination Office for Information Technology Research and Development.

Rachika, N. R. (2013). Model Predictive Control: History and Development. *International Journal of Engineering Trends and Technology, 4* (6), 2600 - 2602.

Saghir, M., & Wayne, J. B. (2002). Internet-Based Teleoperation Using Wave Variables With Prediction. *IEEE/ASME Transactions on Mechatronics, 7* (2), 124-133.

Samad, T., & Annaswamy, A. (2011). *The Impact of Control Technology.* (T. Samad, & A. Annaswamy, Eds.) Munich: IEEE Control Systems Society.

Schwatinski, T., Pawletta, T., Pawletta, S., & Kaiser, C. (2010). Simulation-Based Development and Operation of Controls on the Basis of DEVS Formalism. *EuroSim.*

Sussmann, H. J., & Willems, J. C. (1997). 300 years of optimal control: From the brachystochrone to the maximum principle. *IEEE Control Systems Magazine, 17* (3), 32 - 44.

Traore, M. K. (2008). SimStudio: A Next Generation Modeling and Simulation Framework. *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications Networks and Systems Workshops* (pp. 67:1-6). Brussels, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Vineeth, V. (2014). Design of Duplexer Using Waveguide Filters For GSM Applications. *International Journal of Engineering and Technical Research, 2* (4), 266-271.

Yang, J.-S. (2015, June). Optimization-based PI/PID control for a binary distillation column. *American Control Conference. Proceedings of 2005, 5*, pp. 3650 - 3655.

Zeigler, B. P. (1976). *Theory of Modeling and Simulation.* New York: Wiley-Interscience.

Zeigler, B. P., Praehofer, H., & Kim, T. G. (2000). *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems* (2nd Edition ed.). San Diego, California: Academic Press.