Software Reusability for Component Based Control System

** Raghvendra Narain Tripathi*
*** S. K. Jha*

Abstract

In industry it is common to use control systems to control various hardware. By using control system it becomes easy to reduce costs. Traditional control systems are very costly and they take a lot of time to execute and are not portable. On the other hand a control system which is made with the help of components is less costly and it takes less time to execute and is easy to transfer from one system to another. With this paper we tried to develop a way through which we can get a control software for industry from software components which will increase the use of software reusability, which means applying the previous software components for making new software.

.

Keywords: Components, object-oriented

I. INTRODUCTION

The concept of making software out of previous software is very old. Many scientists and developers have given many techniques such as structured analysis and OOPs. The single finest way is component-oriented way as an improved version of OOP. With OOP a system is formed of small components which work together to do a task. These components can be made freely. Components are required to solve various large scale problems in future.

II. COMPONENTS

Components are the objects used for making adjustable software. Software may be made out of components which are joined by *Plugs*. The regulation defines the communication of components connected into a framework.

Earlier components have been counted with machinery parts. Hence, the terms software IC and software bus are used. As a matter of fact, software components are rather the blue prints and writing diagrams than hardware.

Currently, COM and Java are considered as the best commercial component systems.

III. CLASSIC CONTROL SYSTEMS

Commonly guided system vendor are controlled by titled results. These systems like robot controls (RC) etc. are unsuitable and they take more effort to build as compared to modern control systems and these cannot be easily transferred from one device to another. It costs more to make a control system.

IV. ALTERNATIVE METHOD FOR FLEXIBLE CONTROL SYSTEM

Diverse vendors have displayed how control systems can be made with object-oriented methods. These control systems are very cost effective and they can be easily transferred from one device to another and they take less time to execute. This is the reason these control systems are replacing the classic control system. These components use the concept of Open System Architecture for Control (OASCA) or use Open Modular Architecture Controller (OMAC).

Example of flexible control system is OAC for Windows.

Manuscript received August 3, 2018; revised August 10, 2018; accepted August 16, 2018. Date of publication September 6, 2018.
*R. N. Tripathi is a student of Amity Institute of Information Technology, Amity University, Noida, Uttar Pradesh, India-201301.
(email: raghvendran78@gmail.com)
**S. K. Jha is Assistant Professor with Amity Institute of Information Technology, Amity University, Noida, India-201301.
(email: skjha2@amity.edu)
DOI: 10.17010/ijcs/2018/v3/i5/138777

V. PRIMARY OVERVIEW

Before starting to describe how to make a component-based control system, we want to give requirements for the control software and offer an OOP guided system we have to better understand this area. Based on these needs we designed a model of component technology for the control systems.

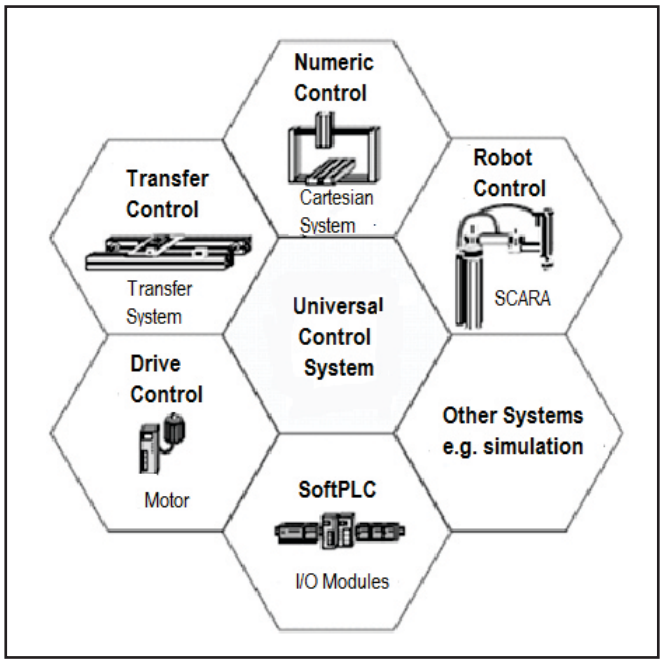

Fig. 1. Integrated Control System

A. REQUIREMENTS

We found the various requirements which users want control systems. These requirements are that a system must provide the service of controlling various other parts as well as it also gives the services of multitasking which means that at the same moment more than one work can be done by the control system.

1) Control Service: The new control system can be used in many areas and for this reason diverse classic controls are joined into a single control. Examples of such control are Robot Control. The advantage of this service is that a single person can manage the production cell very easily.

2) Multitasking Control: In order to guide many devices a major control system has to provide for multitasking. The devices or collection of devices are guided by respective utilization tasks which may execute in parallel. The utilization task can be communicated by interprocess communication tool given by OS.

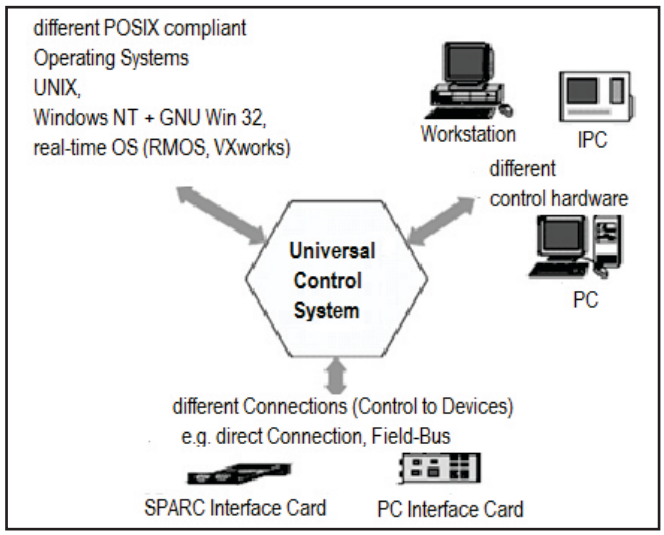

Fig. 2. Platforms of the Universal Control

3) Platform Independence: The system has to back diverse mediums (fig. 2). The model should be self-supporting. Commonly devices are joined straight or by a bus to the control part. Mostly joins are given by interface cards which are connected in the control part or by combined interfaces. The software model must be free from the form of join.

B. OOP Control System

At the start of my work we made an OOP robot control (fig. 3) which could not fulfill all the requirements. The model in Fig. 3 consists of different panels which are methods in OOP. The different layers are given below:

1)Application Panel: This panel is a collection of user drafted application programs. It is basically used for interacton between users and the control part for doing the work.

2) Control Panel: This panel is made of a small system which gives control service. These small systems are owned by the application panel. This panel controls the whole control system.

3) Communication and Coordination Panel: It is a main member of the system and it gives pulses to the main part. It manages other systems as well. It also talks with all the small systems which are present in our control system.

4) Connection Panel: The connection is a collection of solid functions to talk among devices. This panel is important because it keeps track of the connection and talking and arrangement free from old connection. The connection component is used to make connection control system.

Backbone component plays the key role as it processes the various requests and commands sent by the user to the control system with the application part. This backbone component also manages and coordinates with various other components of the system.

Third as well as the last part of our system is Application component which basically interacts with the user with the help of User Menu service present in the

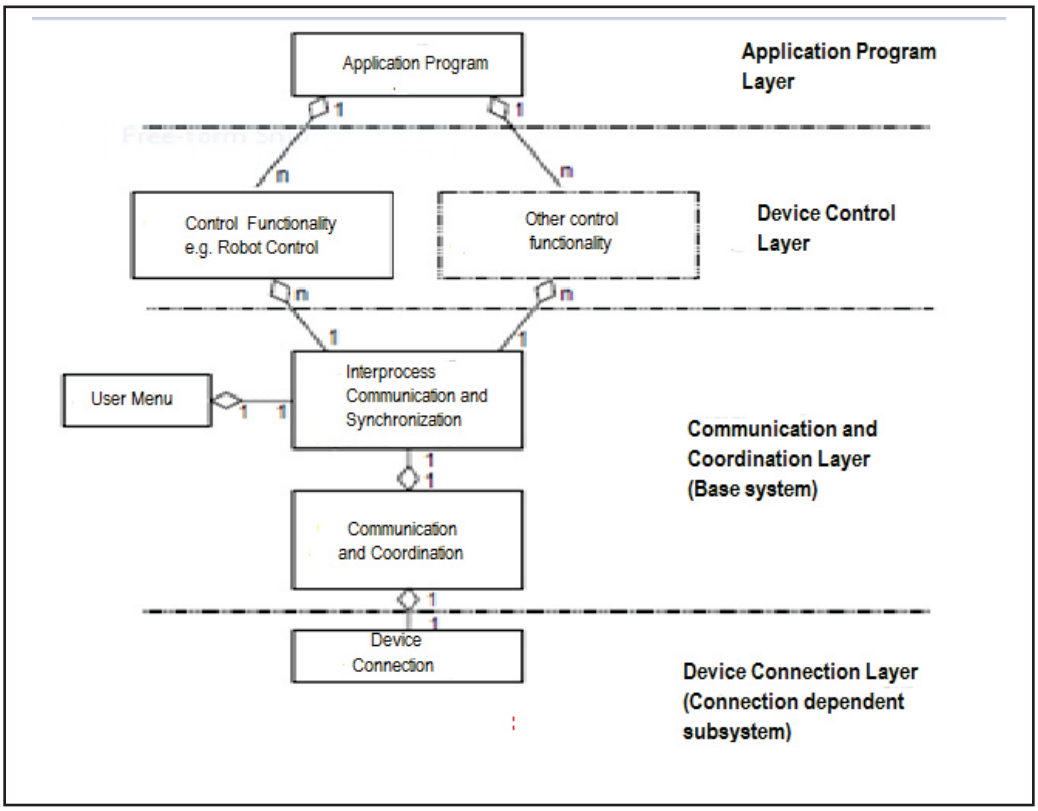

Fig. 3. High-level Class Diagram

VI. COMPONENT ORIENTED CONTROL SYSTEM

First we introduce our primary mindset of utilization of control system. After that we explain the procedures which are used for making components.

A. Basic Components

Basically the control system is a collection of three parts that are connection component, backbone component, and application component.

The connection component is used to make connection between various other components of our control system.

application component.

B. Detailed Model

Basically the control system is a collection of three parts that are the Connection component, Backbone Component, and Application Component.

The Connection component is used to make connection between various other components of our control system.

Backbone component plays the key role as it processes various requests and commands sent by the user to control system with the application part. This

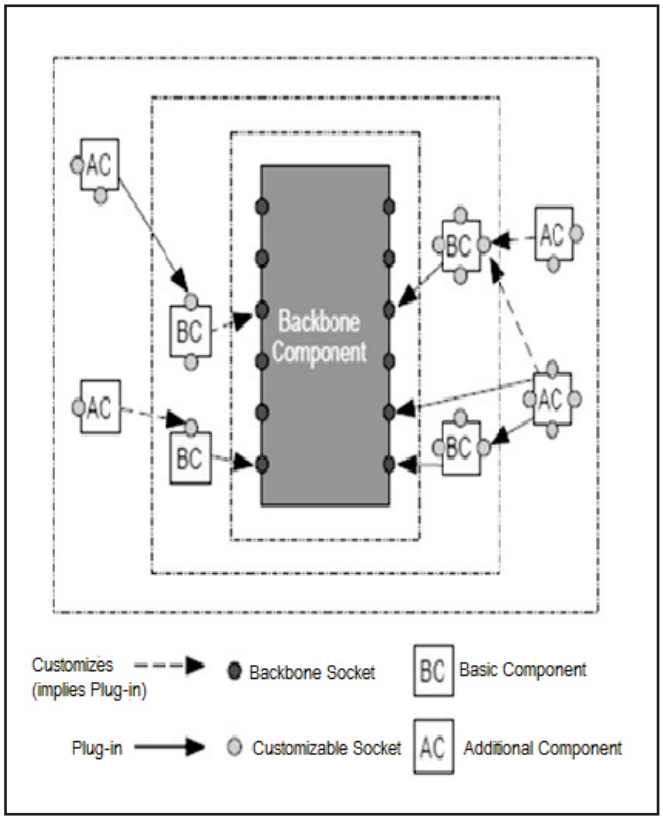

Fig. 4. Component Types

backbone component also manages and coordinates with various other components of the system.

The third part of our system is application component which basically interacts with the user with the help of User Menu service present in the application component.

C. Execution

The component-oriented control systems have been implemented on various mediums such as Windows etc. They get executed very easily and they take less time to execute or to be managed. We have used and checked components of robot arms, a transfer system, and various peripheral hardware joined by I/O devices. The things which we understand while working on component method are:

1) Easy Development Process

A component oriented control system is easy to develop and it consumes less resources than developing a framework. While developing a framework all adjustable components have to be kept in mind before opening and making it.

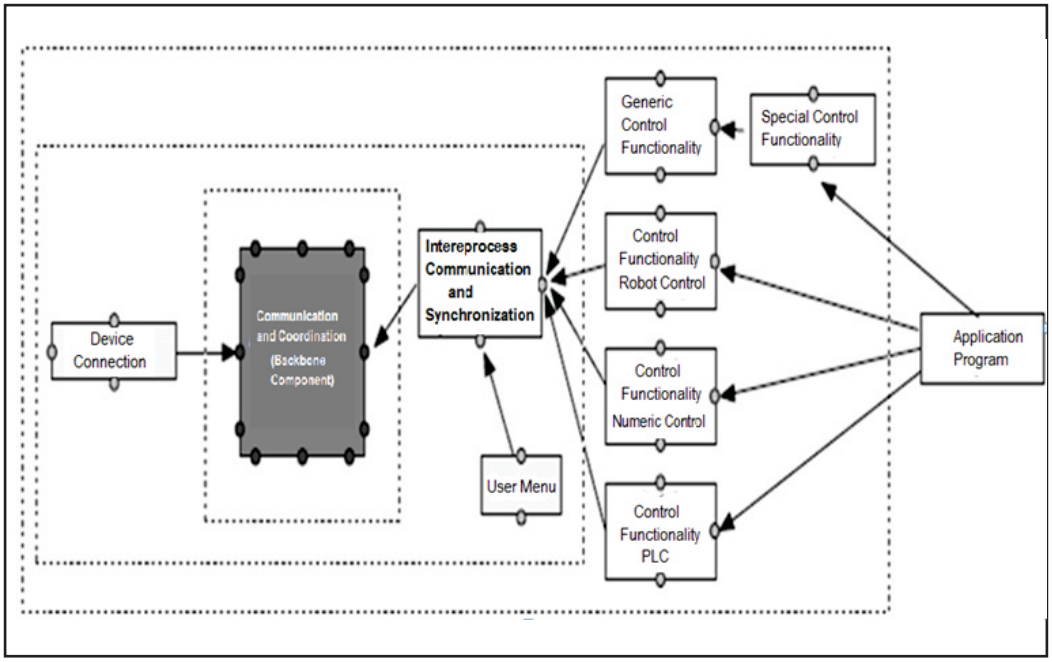

Fig. 5. First Approach of the Component Based Control System

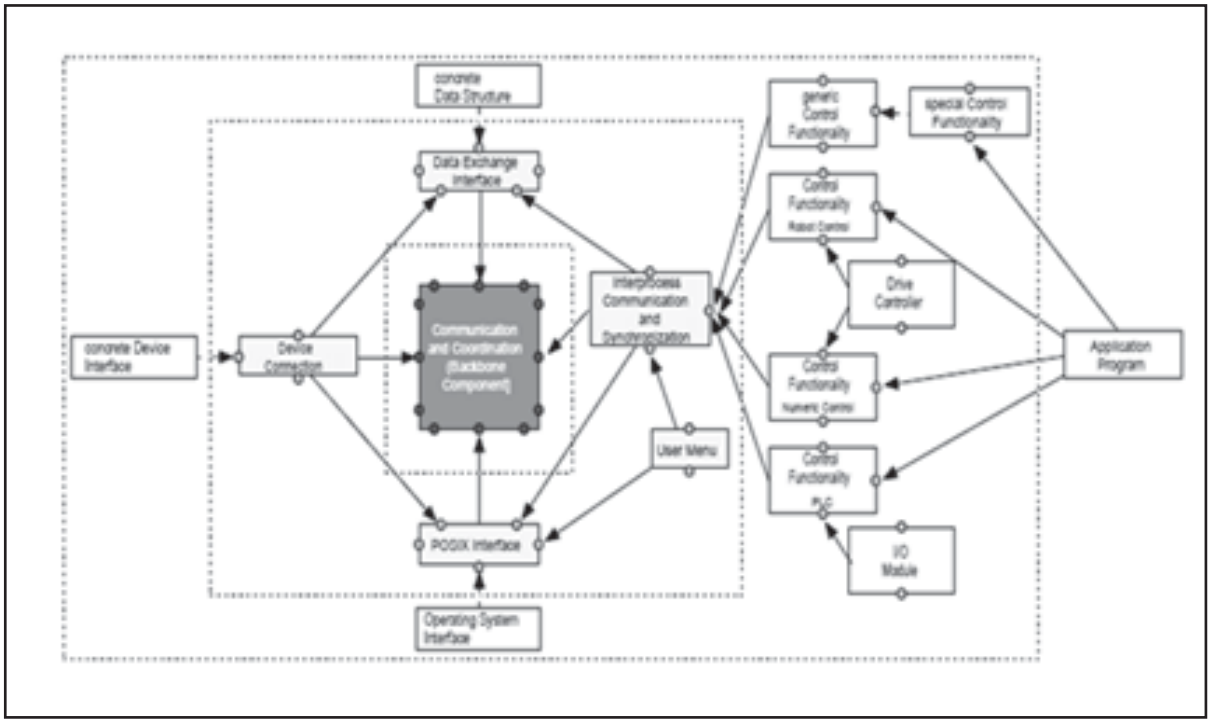

Fig. 6. Extended Component Diagram

2) Interface as Arrangement

The fair explanation of terminals between small systems is important. In robotics area these components are of high standards.

3) Transferable

When we transferred our system to a totally different medium like PC, and we were able to move almost all components from one system to another. The parts which had to be remade were device interface etc.

4) Reusability

Component oriented system has been implemented for various control systems. Therefore, the method of component model with convertible components is a much adjustable model or pattern etc.

VII. CONCLUSION

Component oriented software is the best way for making adjustable as well open systems. These systems are more reliable than old control systems and they take less time to build a control system. These control systems are more cost effective than old control systems. They take less time to execute the work given to them as compared to the classic systems. It is very easy for us to make new software for our control system if any changes are required or suggested by the user. Our control system can be used in any place which uses robots or machinery for doing their work.

ACKNOWLEDGMENT

Raghvendra Narain Tripathi would like to thank his guide Dr. S.K. Jha for his constant guidance and support.

REFERENCES

[1] OSACA Open Architecture for Controls Within AutomationSystems OSACA I & II Final Rep. Project No EP 6378 & EP9115. EU Brussels, April, 1996. [Online]. Available: https://cordis.europa.eu/project/rcn/8845_en.html

[2] S. Demayer, T. D. Meijler, O. Nierstarz, and P. Steyaert,"Design guidelines for tailorable frameworks," *Commun. of the ACM*, vol. *40*, no. *10*, pp. 60 - 64, October 1997.

[3] D. Doscher and R. Hodges,"SEMATEC'S Experiences with CIM Framewo *ACM,* vol. *40*, no. *10*, pp. 82-84, October 1997.rk," *Commun. of the ACM, vol. 40, no. 10, pp. 82-84, October 1997.*

About the Authors

Raghvendra Narain Tripathi has done Bachelors of Computer Applications from MCUJNC, Bhopal in 2017 and he is currently pursuing Masters of Computer Applications from AMITY UNIVERSITY, Noida. His key fields of interest are Networking and Database. During his summer assignment he had written the research paper Software Reusability for Component Based Control System under the guidance of Mr. S.K. Jha.

S. K. Jha is Assistant Professor at Amity Insitutute of Information Technology, Amity University, Noida, India- 201301.