ANALYSIS OF REMOTE DIAGNOSIS ARCHITECTURE FOR A PLC BASED AUTOMATED ASSEMBLY SYSTEM

A Thesis

by

RAMNATH SEKAR

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2010

Major Subject: Mechanical Engineering

ANALYSIS OF REMOTE DIAGNOSIS ARCHITECTURE FOR A PLC BASED AUTOMATED ASSEMBLY SYSTEM

A Thesis

by

RAMNATH SEKAR

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee, Sheng-Jen Hsieh
Committee Members, Chii-Der Suh
Jim Ji
Head of Department, Dennis O Neal

August 2010

Major Subject: Mechanical Engineering

ABSTRACT

Analysis of Remote Diagnosis Architecture for a
PLC Based Automated Assembly System.
(August 2010)
Ramnath Sekar, B.Tech, Jawaharlal Nehru Technological Institute, India
Chair of Advisory Committee: Dr. Sheng-Jen Hsieh

To troubleshoot equipment installed in geographically distant locations, equipment manufacturers and system integrators are increasingly resorting to remote diagnosis in order to reduce the down time of the equipment, thereby achieving savings in cost and time on both the customer and manufacturer side. Remote diagnosis involves the use of communication technologies to perform fault diagnosis of a system located at a site distant to a troubleshooter. In order to achieve remote diagnosis, several frameworks have been proposed incorporating advancements such as automated fault diagnosis, collaborative diagnosis and mobile communication techniques. Standards exist for the capabilities representative of different levels of remote equipment diagnosis. Several studies have been performed to analyze the ability of human machine interface to assist troubleshooters in local fault diagnosis. However, the ability of a remote diagnosis system architecture to assist the troubleshooter in performing diagnosis and the effects of the failure types and other factors in a remote diagnosis environment on remote troubleshooting performance are not frequently addressed.

In this thesis, an attempt is made to understand the factors that affect remote troubleshooting performance: remote diagnosis architecture, nature of failure, skill level of the local operator and level of expertise of the remote troubleshooter. For this purpose, three hierarchical levels of remote diagnosis architectures to diagnose failures in a PLC based automated assembly system were built based on existing standards. Common failures in automated assembly systems were identified and duplicated.

Experiments were performed in which expert and novice troubleshooters used these remote diagnosis architectures to diagnose different types of failures while working with novice and engineer operators.

The results suggest that in the diagnosis of failures related to measured or monitored system variables by remote expert troubleshooters, remote troubleshooting performance improved with the increase in the levels of the remote diagnosis architectures. In contrast, in the diagnosis of these failures by novice troubleshooters, no significant difference was observed among the three architectures in terms of remote troubleshooting performance and the novice troubleshooters experienced problems with managing the increased information available. Failures unrelated to monitored system parameters resulted in significantly reduced remote troubleshooting performance with all the three architectures in comparison to the failures related to monitored system parameters for both expert and novice troubleshooters. The experts exhibited better information gathering capabilities by spending more time per information source and making fewer transitions between information sources while diagnosing failures. The increase in capabilities of the architectures resulted in reduced operator interaction to a greater extent with experts. The difference in terms of overall remote troubleshooting performance between engineer and novice operators was not found to be significant.

DEDICATION

I would like to dedicate my thesis to Amma, Bhagavan, my parents and my brother who have always given me encouragement and the courage and who have always stood by me and have inspired me in all aspects of my life. I would also like to dedicate my thesis to my teachers and friends at D.A.V. Public School and the Jawaharlal Nehru Technological University who helped me during the course of my schooling and undergraduate degree.

ACKNOWLEDGMENTS

I would like to express my gratitude and respect to my advisor, Dr. Sheng-Jen (“Tony”) Hsieh, for guiding me through my research and helping me at every step throughout the course of my research. I would like to thank the ETID department and the Mechanical Engineering Department of Texas A&M University for supporting me through my Master’s study.

I would also like to thank my committee members, Dr. Steve Suh and Dr. Jim Ji, for serving on my committee. I would like to thank my friend Zhenhua Wu for his ideas and support during different stages of the project, Robert Fleming for helping me build the website and assistance with Visual Basic and Riddhi Doshi for helping with trial experiments. I would like to thank my colleagues and the department faculty and staff for making my time at Texas A&M University a great experience. Finally, thanks to my family for their encouragement and love.

TABLE OF CONTENTS

Page

CHAPTER

CHAPTER Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

Page

Page

CHAPTER I

INTRODUCTION

1.1 Motive

Factory automation has seen tremendous advancement due to development in fields in the fields such as computing, artificial intelligence and communication technology. Despite these improvements, there is always a possibility of occurrence of failures in automated systems. While the causes of failure may vary, the fact remains that any system irrespective of the computing strength backing it is susceptible to failures. Fault diagnosis is the process of identifying whether a system is working under normal condition or deviating from the desired behavior and determining fault type, location and potential root causes for those abnormal behaviors if any. With the globalization of trade and international markets, equipment manufacturers sell or install equipment to customers in several countries. As a result, very often the experience and process knowledge of the experts may not be readily available at the manufacturing/assembly site. Downtime of the machine, cost and time constraints on the part of the supplier are crucial factors that necessitate the use of computer and communication technologies to aid the fault diagnosis of the machine. This presents the need for a remote monitoring and diagnosis setup.

Remote fault diagnosis combines the strength of traditional fault diagnosis and computer communication technology [1]. It enables an expert to access any key production equipment from his location via a network or a modem connection [2], in order to remotely monitor, diagnose faults and control the equipment to bring it into full productive state.

This thesis follows the style of *IEEE Transactions on Electronics Packaging Manufacturing*.

Remote diagnosis has evolved from a classical setup [3] involving information exchange using a service hotline where an expert diagnosed the fault by giving advice to the operator for adjustment via phone. In more recent times technical consulting done via internet that enables emails, updates, drawings, diagrams, manuals, video, images etc.

Active information exchange involving remote control of the system via the computer network [4], [5], [6], [7], [8], automated failure detection and diagnosis[1], [4], [8], automated alerting of the operators [1], [9], cell-phone and PDA based diagnosis [10]are some of the advancements in the field.

Despite these advancements, automated systems are still popularly diagnosed by human troubleshooters [11] because their knowledge, experience, and skills for working with unexpected situations that are very difficult and expensive to reproduce by intelligent systems. When working in a remote diagnosis environment, the ability of a troubleshooter to diagnose failures is influenced to a great extent by the quality of tools provided to him and the nature of failure that is to be diagnosed. Many remote diagnosis systems have been proposed and implemented with features like diagnostics algorithms incorporating neural networks [12], fuzzy logic [13] and support vector machine (SVM) [14] for different applications. Researchers have also started exploring sensor deployment strategies for fault diagnosis from qualitative [15] and quantitative perspectives [16]. However, researchers seldom address the area of design methods for remote diagnosis system architecture, because it is a complicated problem [9] involving the knowledge from fields such as computer science, network infrastructure and ergonomics. Remote fault diagnosis architecture refers to the configuration of the components in a remote diagnosis system including network infrastructure, hardware and software. The manner in which they are configured will facilitate diagnosis of abnormal status of automated systems from remote locations.

In a remote diagnosis environment, there exist factors that could affect remote troubleshooting performance. These factors include remote diagnosis architecture, nature of the failure diagnosed, skill level of the operator and the level of expertise of the remote troubleshooter. The difference in troubleshooting performance with the levels of remote diagnosis architecture could be studied. By examining the effect of the nature of failure on the overall performance, it could be possible to identify cases where a type of remote diagnosis architecture could be befitting or may not be the most viable option. By examining the effect of the skill level of the troubleshooter, the ability of remote troubleshooters to utilize the tools provided on the architectures to improve remote troubleshooting performance could be studied. It could be possible to determine if the additional capabilities provided by the architectures would allow remote diagnosis to be performed with operators with limited technical skills. Alternately, a skilled operator could allow remote diagnosis to be performed with lesser capabilities. It would be economical to employ an operator with limited technical skills but this would increase the cost of the architecture.

1.2 Nature of the problem- factors affecting remote diagnosis

A remote diagnosis system involves a variety of cross-platform information integration issues, such as data transformation mechanisms, the design of communication messages, the selection of data transmission protocols, and the construction of a safe network connection [17]. Condition monitoring and fault diagnosis is relatively complex because the solutions consist of a large number of interacting components such as sensors, control units, information processing modules and intelligent computing algorithms [9]. The ability to connect field systems with expertise centers located at distant geographical sites is one of the key aspects of remote diagnosis [18]. The amount of time it takes to communicate a production problem to a remote expert and the quality of information provided affects the resolution time (time taken to diagnose the failure).

Some of the key properties of remote diagnosis [19] are diagnosis ability, collectivity, reliability, expandability, economical construction and maintenance cost. The remote diagnosis environment may be a collaborative multi-user environment or a single expert system. A collaborative environment makes the manufacturing set-up all the more transparent and efficient [18]. The properties identified [9] as crucial to the success of such architecture in this domain are re- configurability due to changes in the number of machine installations, scalability, reliability and effectiveness of information systems. The security and reliability of transactions over the internet is a very important factor in the design of a secure remote diagnosis system [18].

The importance of remote fault diagnosis [20] is that certain failures exist under certain operational conditions which are difficult to be duplicated offline. This is one aspect that is resolved in a remote diagnosis environment wherein, near-real time data is fed via the web and aids fault diagnosis by the expert. System availability which ensues as a result of reduction in the down time of the machine is one of the major contributions of remote diagnosis.

1.3 Scope of work

1.3.1 Objective

The objective of this work is to understand the factors that impact remote troubleshooting performance. These factors are identified as:

- Remote diagnosis architectures.
- Level of expertise of the troubleshooter.
- Types of failure in the automated system.
- Skill level of the local operator.

1.3.2 Methodology

A programmable logic controller (PLC) based discrete automated assembly system is required to be diagnosed remotely. The diagnosis is performed by expert and novice troubleshooters for failures introduced in the system. Different diagnosis tools in the architectures are put to use during the troubleshooting process. The following tasks are outlined to achieve the research objectives.

1. Develop the three levels of architectures to perform remote diagnosis for the PLC based automated assembly system.
2. Identify the failures in PLC controlled discrete automated assembly systems to be diagnosed using the architectures and develop a methodology to duplicate these failures for the automated assembly system.
3. Design experiments based on the factors involved: failures, architectures and operators.
4. Perform experiments in which the troubleshooters remotely diagnose failures while interacting with engineer and novice operators. Collect the data in the form of video recording of the screen viewed by the remote troubleshooter and the audio recording of the conversation between the troubleshooter and the operator.
5. Develop a model to evaluate troubleshooting performance under different combinations of architecture, failure and operator using the principle of multiple attribute decision making.
6. Analyze performance parameters to evaluate how the remote diagnosis architecture facilitates the troubleshooter in performing diagnosis, and study the effect of the factors involved-type of failure, expertise of the troubleshooter and the skill level of the operator-on overall troubleshooting performance.

1.3.3 Assumptions

1. In this study, the application of remote diagnosis to PLC controlled discrete automation systems is considered. Discrete parts manufacturing processes are principally sequences of discrete events and the purpose of process control is to coordinate these events [21].

2. There would be a single point of failure. Each time a failure is introduced, it is not accompanied by any other failure at that time. Cases involving faulty replacements are not considered either.

3. The diagnosis is performed by human troubleshooters and the purpose of the remote fault diagnosis architecture is to facilitate the troubleshooter in diagnosis. No self diagnostics are considered.

4. Information such as long-term and short-term frequency of failure occurrences, history of failures encountered and fixed and record of components changed is not available and not used.

5. The experiments performed in this study are fractional factorial experiments. As a result, some of the higher order interactions between factors are assumed to be negligible.

CHAPTER II

LITERATURE REVIEW

In this section, literature review of the studies on failures in automated systems, remote diagnosis architectures with standards and evaluation of human troubleshooting performance is presented.

2.1 Overview of e-manufacturing, tele-maintenance and tele-diagnosis

E-manufacturing is a system methodology that represents a complete integration between manufacturing and upper level enterprise applications involving planning, maintenance, quality control, optimization etc with the use of computer communication technologies [22]. Performance maintenance and diagnosis is an integral part of e-manufacturing.

E-maintenance [18] is defined as a distributed artificial intelligence environment, which includes information processing capability, decision support and communication tools, as well as the collaboration between maintenance processes and expert systems. Monitoring, diagnosis, predictive maintenance of operations and processes are functions of e-maintenance that are carried out on a web based platform. By leveraging information and internet technologies, the e-diagnostics/maintenance system provides an equipment supplier's experts with the capability to remotely link to a factory's equipment through internet, thereby allowing them to take remote actions, such as setup, control, configuration, diagnosis, de-bugging/fixing, performance monitoring, and data collection and analysis [17].

Tele- service [23] is technically defined as the automatic reading and availability of system and process data and the transmission of these data to analyses and diagnostic programs at the customers' or manufacturers' premises. The basic components of a tele-service unit are identified in [6] as the service provider, service receiver and communication medium. The service provider refers to the internal or external technical personnel with the assistance of necessary technical tools as well as their personal know-how. The service receiver indicates the industrial equipment including industrial devices and their controllers. The communication medium includes all available information technology based infrastructure that can support the tele- service over a distance. Similarly, a remote diagnosis environment involves basic entities like remote troubleshooter, automated system and operator, communication medium and diagnostic tools.

2.2 Failures and faults in automated systems

In reviewing the literature, it appears that the words *fault* and *failure* are often used interchangeably. At times there is no clear distinction between the two, which could lead to confusion. In the context of automated systems, [24] service is defined as the result of performing any task in a process plan. A failure occurs when a resource-which is a collection of entities such as controllers, machine, tools and software program-ceases to deliver the expected service. An error occurs when some part of the resource reaches an undesired state. A fault is the cause of an error, a sequence of errors or a failure. This distinction between fault and failure is adhered to in this thesis. The failures, errors and faults in manufacturing systems can be classified as follows:

1. **Hardware failures** [25]: The causes of hardware failures are categorized as sensor fault and actuator fault. Sensor fault refers to a sensor that is damaged in some way that prevents it from carrying out its required task. Actuator fault refers to an actuator that is damaged in some way preventing it from acting at all, or preventing it from acting within a prescribed period of time. They are also known as equipment faults [26].
2. **Product failures** [25] manifest in the form of products manufactured not conforming to a specific standard. These are also known as quality faults [26] and refer to deterioration in product quality that is not normally detectable by the system sensors, which are conventionally used for control purposes. They could be caused by low

quality in materials or components, or by a hardware fault in the manufacturing system, such as arrival of a faulty component or a component being dropped.

3. **Task faults** [26]**:** A task fault is defined as a deviation from the expected operation of the process due to unpredictability and lack of constraint. E.g. failed insertion operation in a peg -in- hole assembly. They can be detected if they are expressed at the sensor outputs as deviation from the normal operation. These can also be referred to as operational errors [24].
4. **Software failures** [25]: These may be caused due to faults that arise from improper software design or implementation. They manifest in the form of a system or component failure. E.g. an actuator may not actuate at the right time, system may not initialize.
5. **Tolerance errors** [24]: These errors are caused due to defective parts or parts that do not meet the specifications. These are errors attributed to the properties of parts.

The errors and their frequency of occurrence in a video tape recorder assembly system is presented in [27]. The assembly line considered in the study was made up of components such as conveyors, robots and part feeders. The failure data of 89 such assembly cells showed that the parts feeder system, robot grasp and insertion system and fixture location system were most susceptible to faults, followed by unqualified parts.

Data regarding the distribution of faults in robotic assembly [28] were acquired from three robotic assembly cells grouped under set A, set B and set C with 98, 392, 368 samples of assembly actions respectively. It was observed that the failure cases registered were 31.6% for set A, 30.6% for set B and 13.3% for set C. All the failures except one in set A were attributed to failure of insertion or seating (insertion where gravity is intended to assist). Presentation failures were caused by the deviation in the part configuration as expected by the work cell. For sets B and C, failures were distributed as shown in **Table 1**.

Table 1—Failure distribution for robotic assembly cells

Failure (B)	**Percentage**	**Failure (C)**	**Percentage**
Insertion	51.3%	Insertion	71.4%
Grasping	17.5%	Dropping	16.3%
Sensing	16.7%	Grasping	6.1%
Presentation	8.3%	Others	6.1%
Flawed parts	5.8%		

The screw insertion process [29] was considered in detail wherein the causes for insertion failure of the screw were identified as mismatch in the diameter of the hole in the base plate with the screw, which is inserted. Another type of insertion failure is jamming, which could occur due to a number of reasons, including manufacturing errors where the main body of the screw widens close to the head, a hole diameter reduction at the end of the insertion is encountered or presence of burrs in the hole.

The possible failures in robotic assembly [30] were identified as eccentric gripping of the peg due to loss of tolerance of the position of the gripper or fixture; impacts damaging the peg or fixture during extraction of the peg and presence of burrs on the edge of the base part or dirt on the chamfer of the bore, resulting in a fault. The causes for insertion failure were identified as dimensional errors of the peg or the hole part (height and diameter), including the angular misalignment of the peg; presence of extraneous matter at the contact point resulting in high friction; and improper peg parts having ruts and burrs.

From these observations, it can be concluded that insertion and grasping a part and keeping hold of it are among the difficult robotic operations and are most susceptible to faults. This is followed by sensing failure and bad parts. These failures would ideally represent typical situations that need to be addressed in the implementation of remote diagnosis for automated assembly systems.

2.3 Classification of existing work in remote diagnosis

The three most fundamental aspects of remote diagnosis of automated systems are monitoring, control and diagnosis. We classify the capabilities discussed in existing work in the field of remote diagnosis in terms of remote monitoring, remote control and remote diagnosis as shown in **Table 2**.

Table 2—Summary of capabilities discussed in existing work

Remote Diagnosis architecture capabilities	**Existing work supporting capability**
Remote Monitoring	
Graphical interfaces displaying operational status data	[31], [1], [12], [6], [4], [7], [32], [33]
Sensors (external to the system) to measure parameters related to system condition	[9], [12], [34], [35], [10], [7], [8]
Alarm messages, automated messaging	[9], [1], [12], [5], [6], [35], [10], [7]
Web camera (video feedback)	[31], [1], [12], [4], [10], [7], [8], [36], [37], [38]
Duplicating information on the server computer on the client side	[31], [5], [32], [33]
Remote Control	
Remote control of the system (access to the controller)	[4], [5], [6], [7], [8]
Access to the server computer	[4], [1] [31], [5], [10], [32], [33]
Remote diagnosis	
Video conferencing (local operator and remote expert)	[1], [12], [8], [32], [33]
Voice chat, telephone	[12], [4], [6], [10] , [32], [33]
File transfer	[12], [4], [6] , [32], [33]
Collaborative diagnosis	[4], [12], [6], [8], [32], [33]
Historical data	[4], [5], [10], [8]
Textual communication (chatting, e mail messages)	[4], [12], [6], [10]
Automated failure diagnosis (neural networks, petrinet, VI, rule base, intelligent system)	[9], [4], [1], [12], [34], [17], [35], [10], [7], [8]

2.4 Remote diagnosis architectures

In the remote diagnosis architecture proposed in [1], failure of components or subsystems on the client end instigated the transfer of machine status and other information to the remote expert center via the computer network such as internet, ISDN etc. The experts then carried out the diagnosis operation. The proposed remote diagnosis architecture comprised of a local monitoring and diagnosis system and a remote expert system interconnected by means of internet. The remote system was a fall back when the local diagnosis system was incapable of ascertaining the cause of failure.

In the remote diagnosis architecture proposed in [12] , sensor data was transferred via a web server and used by virtual instruments accessible to local operators and remote experts to conduct fault diagnosis. Multimedia information in the form of images, audio was also transferred via the web to aid diagnosis. The internet platform allowed for collaboration between various experts and operators by means of net meeting, video conferencing, web broadcasting etc. An automated intelligent neural network based fault diagnosis system was used to supplement the diagnosis process.

A vibration based remote monitoring and diagnosis system was proposed in [10]. Equipment data processed by the data acquisition system was analyzed by the diagnostic software in order to determine the health condition of a machine. The diagnosis system analyzed vibration signals to indicate the occurrence of a failure and sent automated messages and upon abnormality, made calls to the engineer through the mobile phone connected to the server. The diagnostic results were stored to a data base and published to a web server for the troubleshooter to view on the internet browsers or authorized mobiles and PDAs. Web cameras provided images, real time video and audio of the machine operation. Different kinds of virtual instruments that perform analysis on the vibration based data served as the primary diagnosis tools for the remote troubleshooter.

A remote diagnosis architecture to diagnose placement machines [39] was presented [4], involving remote access to the system. A JAVA based platform was built which provided for dynamic acquisition of system parameters i.e. system states, failure messages and knowledge regarding important control variables. A centralized failure database and diagnosis tools was made available on the web so that any of the remote clients or the local operator could avail of it to perform diagnosis. The sub contractors could also collaborate with the expert and the operator to aid in trouble shooting. The description of failure were meted out to the expert via the internet in the form of fixed images, moving images, audio data, text data etc. and the guidance support could come in the form of text messages, emails and conversation over phone.

In this client server based remote diagnosis architecture [31], a web based remote monitoring, control and diagnosis platform using digital cameras, PLC controller and human machine interface (HMI) for a flexible manufacturing system was proposed. A VB based HMI running on the server computer, duplicated on the client side enabled both monitoring and control of the system including the change of parameter settings of the process. The HMI was created so as to represent the status of each task in the operational sequence for each of the cells in the plant. It also supported automated fault diagnosis based on a petri- net model of the system.

Remote monitoring and maintenance system for CNC machine tools was proposed in [5]. Communication device with mobile technology was provided for each machine tool which sent an automated email notification of machine tool status to the manufacturer's remote diagnosis server. Additional details such as operational status of the CNC machine with the time stamp were recorded. The HMI of each CNC machine transferred data to a centralized data base. The troubleshooter could make use of the remote screen of customer machine tool and was also allowed access to the CNC and obtain data necessary to perform machine tool diagnosis.

The remote diagnosis of a transportation model involving conveyor transporting metal blocks, sensors, indicator lights etc. controlled by a PLC was considered in [6]. Tele- monitoring of real time process status was achieved via HTML interfaces. Remote access to the automated system enabled the adjustment of certain control parameters. Tele alarming by means of email messages was developed wherein the controlling PLC sent email notifications in error situations. File transfer was supported by means of secure FTP between the service provider and the service receiver.

2.5 Remote diagnosis standards

SEMATECH [2] laid down standards for e- diagnostics of semiconductor manufacturing industry. E-diagnostics [2] is defined as the capability to enable an authorized equipment supplier's field service person to access any key production or facilities equipment from outside the facility/factory via network or modem connection.

Access includes ability to remotely monitor, diagnose problems or faults, and configure/control the equipment to bring it into full productive state rapidly and within security guidelines.

According to Sematech, e-diagnostics capabilities [2] are described within four levels (0–3). Each level is intended to build on the previous level, each bringing increased capability. An architecture that is level-3 capable necessarily has levels 0 through 2 capabilities as well. The level numbers increase according to a blend of many factors: the sequence of support tasks that might be performed, the ease of implementing the necessary factory infrastructure and tools required to execute the diagnostic and repair tasks and decreasing human assistance and increasing automation expected with each level. MIMOSA, OSA-CBM are among the other standards established for e-maintenance but focus more towards predictive maintenance [18]. The following **Table 3** summarizes the capabilities of three remote diagnosis architectures representative of the levels built in this work. Architecture-1 (A1) incorporates the capabilities of level-0. Architecture-2 (A2) incorporates the capabilities of level -1. Architecture-3 (A3) builds on level-1 and incorporates certain capabilities from levels-2 and 3. Thus the three architectures can be summarized as follows:

Architecture-1: This has the basic capabilities such as remote connectivity and collaboration between the troubleshooter and the operator. It is similar to the capabilities discussed on level-0 type architectures presented in [2]. Capabilities include video, voice transmission, still images, text-chat capability, and secure file transfer.

Architecture-2: It is similar to the capabilities discussed on level-1 type architectures presented in [2]. It includes the capabilities of architecture-1 and additionally involves direct access to the PLC, near real time monitoring of operational status by means of a graphical interface. Existing architectures supporting capabilities similar to this architecture are [4], [1], [12], [34], [5], [35], [10].

Architecture-3: Includes the capabilities of architectures 1 and 2 along with some additional features characteristic of levels-2 and 3 presented in [2] like stage diagram based human machine interface (HMI), video playback, time based record of sequence of operations. Existing architectures supporting capabilities similar to this architecture are [31], [6], [7].

Table 3—Summary of capabilities- remote diagnosis standard

S. No	**Capability (level)**	**Description**	**A1**	**A2**	**A3**
1.	Voice transfer (0)	Ability to provide Voice over IP between participants.	✓	✓	✓
2.	Video transmission (0)	Ability to provide streaming video between participants.	✓	✓	✓
3.	Still images (0)	Ability to exchange still images between participants.	✓	✓	✓
4.	Chat capability (0)	Ability to support text chat sessions among participants.	✓	✓	✓
5.	File transfer (0)	The ability to provide users with the capability to transfer pre-authorized files.	✓	✓	✓
6.	Data collection-operational (1)	Provision of agents to collect equipment data. E.g., sensor data, engineering measurements, equipment settings, etc.		✓	✓
7.	Data collection-exceptions (1)	Errors, warnings, alarms and other unexpected anomalies automatically detected by the equipment.		✓	✓
8.	Data storage (1)	Data stored at the system site for future analysis and reporting.		✓	✓
9.	Monitor remote equipment operation in near real time (1)	Means by which an authorized remote user can watch the evolution of equipment parameters and/or state changes in near real time.		✓	✓
10.	Remote configuration (1)	Allow a remote user access to a piece of equipment or equipment environment to analyze and modify software aspects of the equipment.		✓	✓
11.	Remote equipment operation (1)	Ability to remotely view and actuate user interface functions as if standing at the equipment.		✓	✓
12.	Historical data (2)	Provision of reporting capability to view historical data including; operational, exception, and parametric data.			✓
13.	Reporting operational and exceptional data (3)	Provide sufficient detail to understand the operational and exception states of the equipment.			✓
14.	Diagnosis decision support (3)	The ability to apply logic or rules to the data to make simple decisions and initiate secondary actions.			✓

2.6 Evaluation of human troubleshooting performance in fault diagnosis

The ability of different levels of operator machine interface to assist operators in the local fault diagnosis of discrete automated systems was discussed in [21]. Experiments were performed wherein operators used 3 hierarchical levels of interfaces with increasing capabilities to diagnose 3 different failures in an automated manufacturing system. The purpose of the tests was to empirically evaluate the three types of operator interfaces and expose the drawbacks in some of the commonly used user interfaces in terms of their inefficiency to facilitate human troubleshooting performance in fault diagnosis tasks. The time taken to diagnose the fault, the number of information screens viewed and the number of diagnostic tests performed were identified as the measures of performance. The impact of confounding variables: type of interface, nature of failure and the order of experiments were also considered.

Experimental evaluation of the effectiveness of functionally abstracted information in fault diagnosis was done in [40] in order to design for visual information display for process control. Improving human problem solving performance is the objective of the process interface considered for fault diagnosis in nuclear power plants. The ability of the hierarchical abstraction to improve troubleshooting performance was tested by implementing three levels of interface with increasing capabilities. The impact of the complexity of the diagnosis problem on the performance was also considered. It was seen that certain combinations of level of information and type of display exist that generate optimum performance. Recommendations regarding the integration of information level with display type were made to improve the effectiveness of any given display.

An empirical study to test the ability of ecological interfaces to help diagnose faults in petrochemical processes was performed in [41] with professional operators in realistic plant settings. Three types of display interfaces: one contemporarily used and two hierarchical levels of ecological interfaces- one traditional and another augmented with additional task based information were used. It was seen that the ecological interface with additional task based information facilitates the operator to a greater extent to troubleshoot failures and the contemporary interface was least helpful. The time taken to complete the task, the number of control actions and the diagnosis accuracy were used to determine the performance score. Lower task completion time, lower number of control actions and better diagnosis accuracy were seen as the desired characteristics of an effective interface.

In [42] was proposed the experimental investigation of the compatibility of information types with diagnostic strategy. The application was related to building decision aiding systems for fault diagnosis in nuclear power plants. Experiments were performed using four different types of information aids that are representative of common operator support systems for diagnosis tasks in nuclear power plants in order to determine what information type would be effective for a particular strategy and facilitate the operator during diagnosis. Conclusions were made regarding the suitability of information aids for operator strategy and that the effectiveness of information aids was dependent on the strategy employed.

The effects of hierarchical display on human problem solving performance was studied in [43]. Faults were introduced in computer simulations of logic circuits which were diagnosed by subjects with different levels of technical competence. It was seen that with subjects less competent in diagnosis, the hierarchical display interface was more helpful where as competent troubleshooters found both types of interfaces similarly compatible. Thus, they established that the ability of an interface to facilitate diagnosis was also dependent on the skill of the user.

2.7 Summary of needs required to be addressed

The following **Table 4** is a brief overview of the similarities and contrasts between previous research and this study with regards to evaluation of troubleshooting performance in fault diagnosis tasks.

Table 4—Similarities and contrasts between existing research and present study

Previous research	**Present study**
Contrasts	
Evaluation of troubleshooting performance for fault diagnosis by operators at the system site (local). [43], [42], [41], [40], [21].	Evaluation of troubleshooting performance for remote diagnosis of failures by troubleshooters.
The troubleshooter and any other operators (if involved) were physically present at the system site. Treatments were implemented by the troubleshooter himself.	Troubleshooter interacts with the operator in order to achieve failure diagnosis. The troubleshooter can view the system through the web camera and the treatments are implemented by the operator. The troubleshooter is given feedback by the operator about the failure symptoms, effect of a treatment etc.
Similarities	
The effect of failures on troubleshooting performance was considered [40].	The effect of failures on the performance is considered keeping in mind the interaction effects.
The effect of the skill of the troubleshooter on the troubleshooting performance was considered [43].	The difference in strategies of expert and novice troubleshooters in a remote diagnosis environment is considered.
Effect of the nature of the process interface on the troubleshooting performance was considered.	The ability of the remote diagnosis architecture to assist the remote troubleshooter is studied.

From the literature review it is seen that a lot of previous research focused on various methods of achieving remote fault diagnosis. Different levels of remote diagnosis architectures exist that support different types of capabilities summarized in the standards for remote diagnosis [2]. Several proposed architectures incorporate these capabilities for different automated systems. But they seldom address the level of detail involving the mechanism with which the capabilities on the proposed architectures enable the troubleshooter to remotely diagnose failures occurring in automated systems. Although the generic mechanism of achieving failure diagnosis using the tools on the remote diagnosis architecture is addressed to some extent in [31], a failure based empirical evaluation is still missing.

A lot of work has been done to identify the failures occurring in discrete manufacturing systems and the frequency of occurrence of the failures [28]. Experimental evaluation of factors facilitating human troubleshooting performance in local fault diagnosis tasks has been addressed in work done previously. The effectiveness of an interface to aid local fault diagnosis by human troubleshooters on the basis of different types of failure, type of interface, skill level of the troubleshooter, etc. is considered. However, there is limited discussion on how the existing architectures facilitate troubleshooters in diagnosing different types of failures in automated systems in a remote diagnosis environment. Whether the capabilities on the advanced levels of architectures are required needs to be tested empirically based on the nature of the failure, expertise of the troubleshooter and the skill of the operator.

In a remote diagnosis environment, the remote troubleshooter works in conjunction with a local operator in order to achieve failure diagnosis. In such an interactive setting, it is possible for the troubleshooting performance to be affected by the skill of the local operator. The ability of the troubleshooters to use the capabilities provided and their diagnostic strategies could vary depending on their level of competence. A remote diagnosis architecture forms the interface of an automated system to a remote troubleshooter. The primary objective of any remote diagnosis architecture is to facilitate its users to determine the root cause of the failure in the system diagnosed. The usability of the interface by human troubleshooters, the skill level of the local system operator, nature of failures, the competence of the troubleshooter considered in traditional fault diagnosis are perfectly applicable and contribute towards performance in a remote diagnosis environment.

So, in this thesis the diagnosis of different types of failures in an automated system using three increasing levels of remote diagnosis architectures is addressed with emphasis on the use of these architectures by a human troubleshooter. An attempt is made to empirically evaluate the ability of the remote diagnosis architectures to assist troubleshooters of different expertise levels by comparing the remote troubleshooting performance with three different levels of architectures in alternative situations of failures and operators. As a result, recommendations can be made regarding the use of the architectures depending on the nature of failure, expertise of the troubleshooter and skill level of the operator.

CHAPTER III

EXPERIMENTAL SET UP

This chapter explains the experimental set up used in this research and the hardware and software tools used to implement the remote diagnosis architectures.

3.1 System: automated assembly line

The re-configurable dual-robot assembly system [44] in the Rockwell Automation Laboratory shown in **Figure 1** consists of four stations. The first is the inspection station which is used for verifying whether the base part is within specifications. The second station works as a buffer station for station 3. Stations 3 and 4 are identical assembly stations, where pneumatically operated, gantry type pick and place robots assemble pegs into the holes on the base part. The assembly line mimics the assembly of pegs into the holes in the base part carrier and its actions are controlled and synchronized by a programmable logic controller (PLC). The major components of the automated assembly system are:

1. Two pneumatic robots.
2. Part stoppers and sensors.
3. Part feeders
4. Conveyor
5. Vision system.

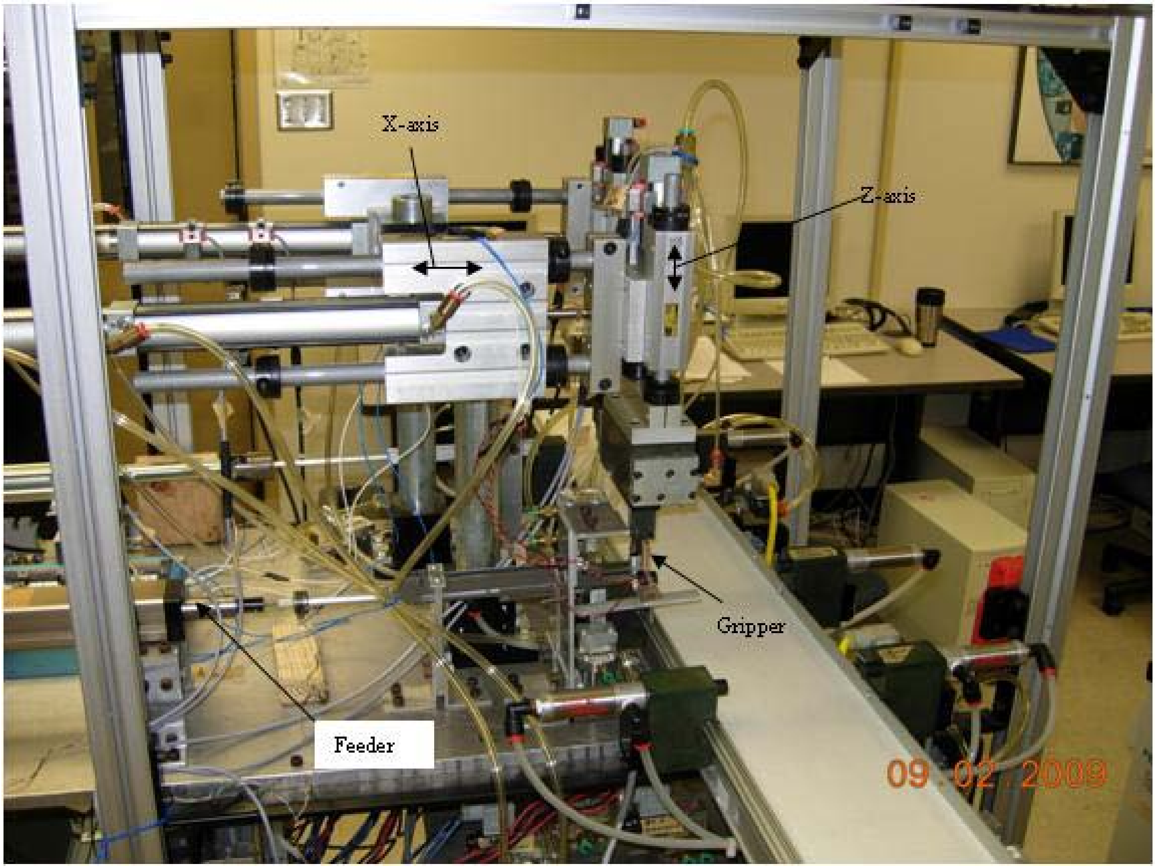


Figure 1—Overview of the automated assembly line

3.1.1 System hardware

3.1.1.1 Robot arm

The robot arms consist of shoulder, elbow and gripper that allow for three degrees of freedom: X-direction, Z-direction and grasp. The movement along each axis is actuated by an air cylinder, controlled by a solenoid operated direction control valve and synchronized by the PLC. The shoulder and the elbow of the robot arm provide for the translational X and Z degrees of freedom respectively. They comprise of gantry type slides, mechanical stops and hall-effect sensors. The stroke length can be adjusted by varying the position of the mechanical stops. A standard pneumatic angular gripper picks up the peg parts and inserts them into the hole on the base part. The movement along each axis is actuated by an air cylinder, controlled by a solenoid operated direction control valve.

3.1.1.2 Conveyor, part stopper and base part

A belt conveyor transports the base part through the assembly line. It consists of a steel reinforced canvas mat and is driven by a 120V AC motor. The conveyor is approximately 60 inches long and 5 inches wide. Four sets of part stoppers (pneumatic cylinders) are mounted at regular intervals along the track of the conveyor, one for each station. The rectangular base part block fits the width of the conveyor and has slots cut on all four corners which allow the part stopper cylinders to locate into to fix the base part at each station. It has holes in the center-one square and another round-for the placement of pegs during assembly. The details of assembly station-1 are shown in **Figure 2**.

3.1.1.3 Part feeders

The part feeder system consists of a sliding rail in which the parts to be picked up by the robot, are moved. The parts slide on the rail, as they are pushed by a linear screw type actuator. The parts are placed manually onto the rail. The linear actuator at assembly station 1 is powered by a stepper motor driven using a stepper motor driver while the one at assembly station 2 is powered by a DC motor driven by means of relays by the PLC.

3.1.1.4 Peg parts

The peg parts are made of 60A durometer hardness silicone rubber. The ones assembled at station 1 have a square cross section of 3/8" x 3/8" and are approximately 1" long. The ones assembled at station 2 have a circular cross section of 3/8" diameter and 1" in length. The parts are chosen to be of rubber because in case of a misalignment during assembly, the rubber parts being flexible do not cause any damage to the robot gripper or the base part.

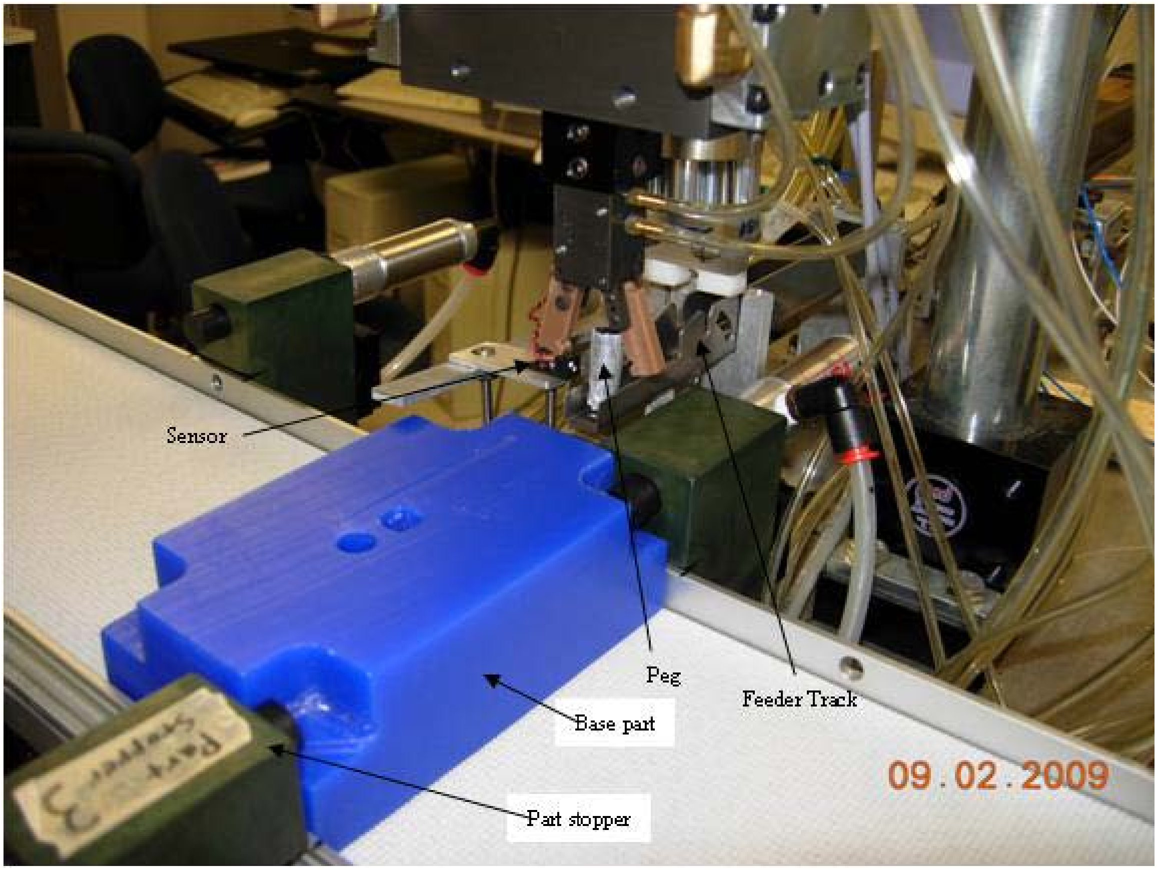


Figure 2—Details of assembly station-1

3.1.1.5 Vision system

The vision system used is an Allen Bradley® Configurable Vision Input Module (CVIM) 5370. It is composed of a camera, pyramid integrator module (resource manager), logic processor, remote scanner and the vision processor. The local I/O gateway is used to communicate with the optical sensors aligned with the stoppers, external I/O board and interface box, and SLC 5/05. An interface board consisting of multiple solid state optical isolator modules, relay and power supply were utilized to achieve communication between the vision system and the PLC. The camera arrangement for the vision system is shown in **Figure 3**. The purpose of the vision system at station 1 is to inspect the base part for proper placement of the holes for the assembly operations. The image of the base part is captured by the camera in response to a trigger input from the PLC. If the result of the analysis by the vision system is bad, then the part stopper cylinder is not allowed to actuate.

3.1.1.6 Pneumatic system

The pneumatic system is comprised of the compressor, pneumatic line, FRL (filter, regulator and lubricator) unit and the solenoid operated direction control valves. The pneumatic actuators used for each robot are gantry slides for x and z-direction, a rotary actuator and a gripper. The x-direction slide has larger stroke than that on the z-direction slide. Also at each station, there exist pneumatic part stopper cylinders that extend to stop parts at each station and retract to release when the operation at each station is completed

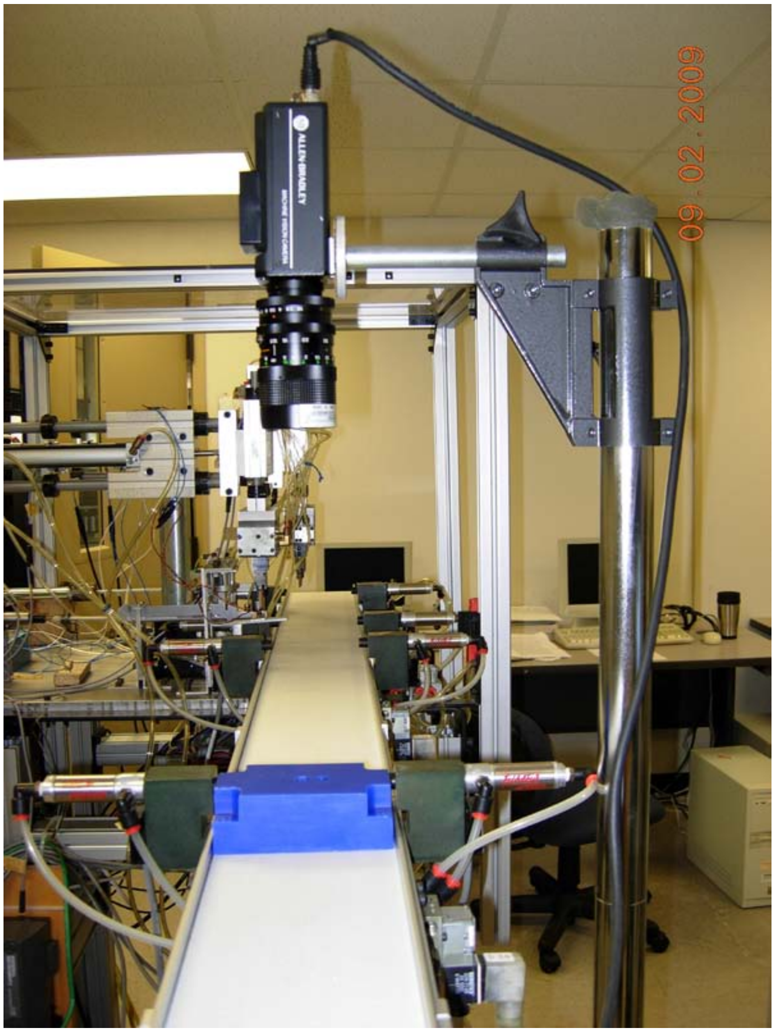


Figure 3—Vision system

3.1.1.7 Sensors in the system

The robot’s axes end limits are monitored by the use of hall effect sensors that indicate the end limits of the pneumatic actuator when it is completely extended or retracted. There are two such sensors on the horizontal axis and two on the vertical axis of each robot. Photo sensors at each part stopper are used to detect the presence of the base part at each station. Reflective photo sensor is used to indicate the limits of the return stroke of the second part feeder. On the part feeder rail at station 1, a reflective photo sensor is used to detect the presence of the part at the rail ready to be picked up the robotic gripper. On the part feeder at station 1, there are magnetic reed sensors that are used to indicate the end limits of the linear actuator.

3.1.2 Control system

The controller used to control the assembly line is an Allen Bradley® SLC 5/05 programmable logic controller (PLC). The processor communicates with the computer (programming interface) using RSLinx®. It is capable of communication via RS-232 and ethernet (local area connection). The PLC has two digital input modules and two digital output modules for interfacing the inputs and the outputs of the system.

3.1.3 Sequence of operation

When the base part coming along the conveyor is sensed by the optical sensor at the first part stopper, an input is sent to the vision system. The vision system analyzes the base part and the part stopper is released upon a good inspection result. When the base part arrives at the second part stopper, if there is no part blocking the optical sensor at station 3, part stopper 2 is released. When the part reaches the third station and is detected by the optical sensor at part stopper 3, part feeder 1 actuates in a small increment to push the peg along the feeder rail by a small distance. Then, the robot picks up the peg and places it on the base part. Once this pick and place operation is complete and the optical reflector at part stopper 4 is not blocked, part stopper 3 is released. When the part reaches the fourth station detected by the optical sensor at part stopper 4, the part feeder 2 actuates in a small increment to push the peg along the feeder rail by a small distance. Then, the robot arm picks up the peg from the part feeder 2 and places it on the base part. Finally, part stopper 4 is released. The sequence for robot arms 1 and 2 are:

1. Open gripper, lower elbow, pick up part from part feeder track.
2. Raise elbow, extend shoulder, lower elbow, open gripper and

3. Close gripper, raise elbow, retract shoulder.

3.2 Remote diagnosis architecture-1

This architecture has the basic capabilities to enable remote connectivity and collaboration between the troubleshooter and the operator. It's capabilities are similar to those discussed for level-0 type architectures presented in [2] which includes video, voice transmission, still images, textual communication and secure file transfer. The tools used to implement these capabilities are tabulated in **Table 5** below.

Table 5—Summary of capabilities on architecture-1

S No.	**Capability**	**Tool**
1.	Web camera (video feedback)	Network camera, TAMU VPN
2.	Textual Communication (chatting, email messages)	Skype
3.	Video conferencing (local operator and remote expert)	Skype
4.	Voice chat	Skype, Headset with microphone
5.	File transfer	Skype
6.	Documentation (Control Program, I/O listing)	RSLogix® 500 and the website for experiments

The schematic layout of the capabilities on architecture- 1 is shown in **Figure 4**.

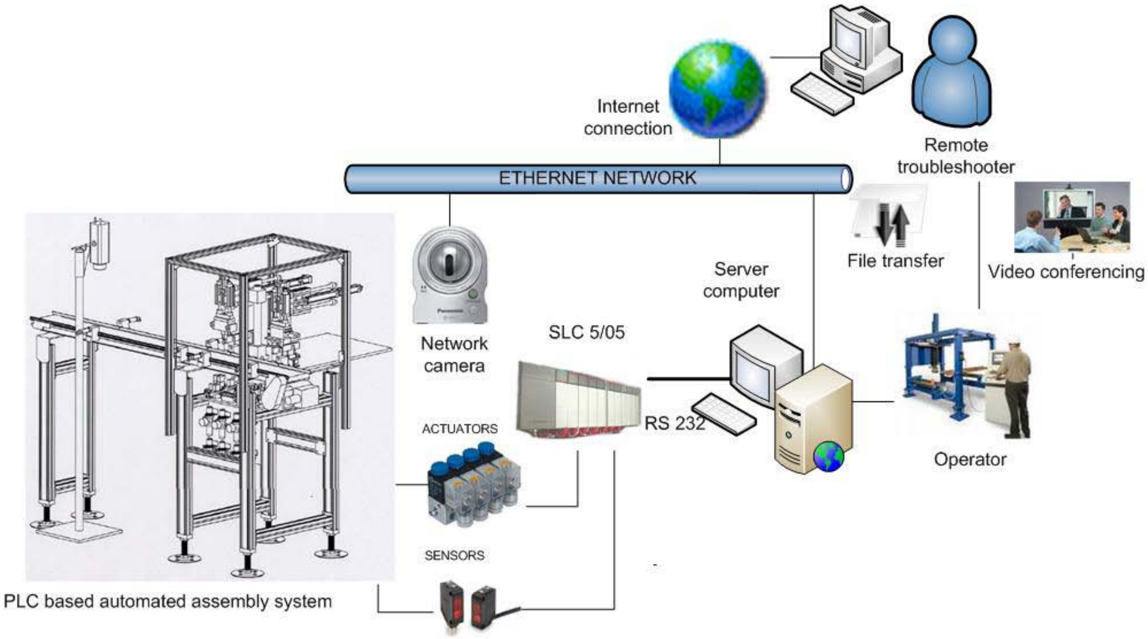
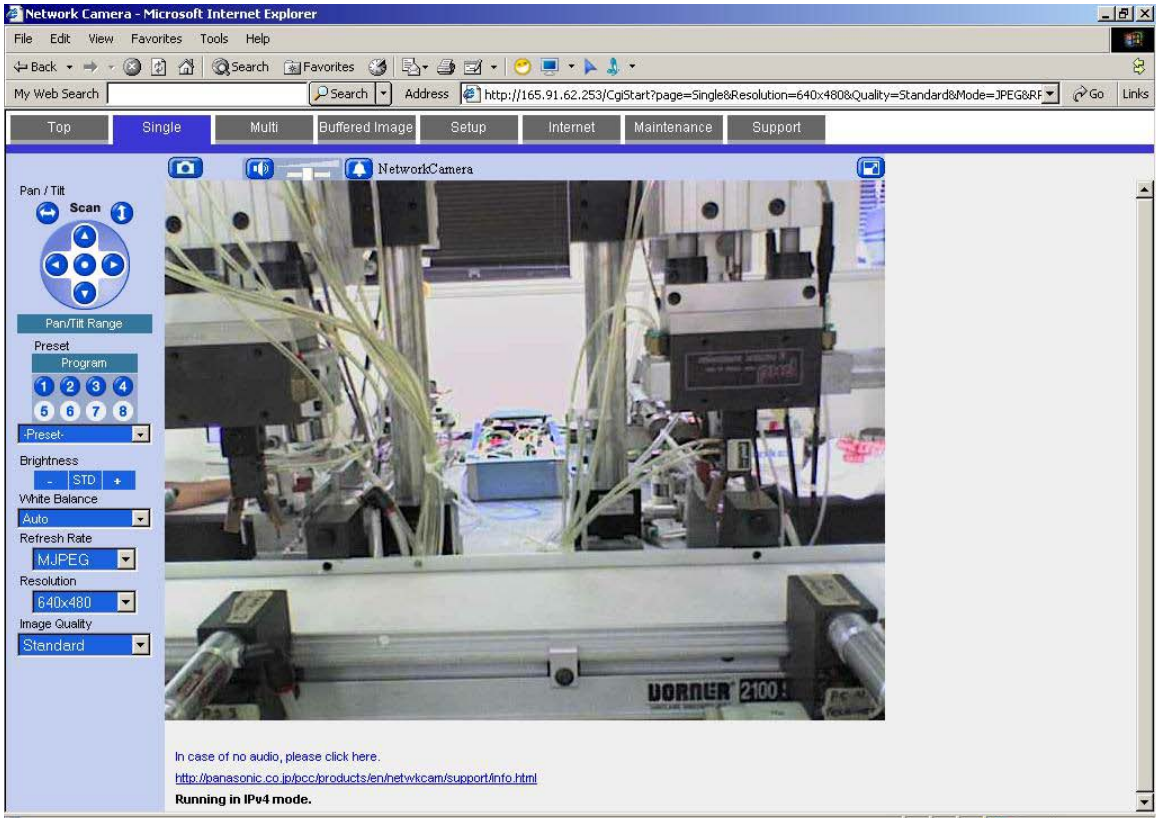


Figure 4—Schematic layout of capabilities on architecture-1

TAMU VPN [45] was used for off-campus troubleshooter to gain authorized access to university's computer network resources. In architecture-1, VPN is required for the remote subject to access the network camera which is an entity on the local area network. The network camera captures live images and sound that can be recorded at the server computer. The web camera communicates via the ethernet and can be controlled remotely. **Figure 5** is a screen shot of the interface of the network camera.

Skype [46] enables users to communicate by both voice and instant messaging via the internet. It also allows file transfer, using which the operator and the remote expert exchange programs, pictures, etc. The ladder logic control program is provided to the remote troubleshooter along with documentation of the system (description of the I/Os). However, on this architecture, the troubleshooter is not allowed direct access (go online) to the PLC using the ladder diagram.



Address http://165.91.62.253/CgiStart?page=Single&Resolution=640x480&Quality=Standard&Mode=JPEG&RF

White Balance Auto

Refresh Rate MJPEG

Resolution 640x480

Image Quality Standard

In case of no audio, please click here.

http://panasonic.co.jp/pcc/products/en/netwkcam/support/info.html

Running in IPv4 mode.

Figure 5—Interface of the network camera

3.3 Remote diagnosis architecture-2

It has capabilities similar the ones discussed on level-1 type architectures presented in [2]. It includes all the capabilities of architecture-1 and additionally involves direct access to the PLC, near real time monitoring of operational status by means of a graphical interface. The tools used to implement these capabilities are tabulated in **Table 6**.

Table 6—Summary of capabilities of architecture-2

S No.	Capability	Tool
1.	Web camera (video feedback)	Network camera, TAMU VPN
2.	Textual Communication (chatting, email messages)	Skype
3.	Video conferencing (local operator and remote expert)	Skype
4.	Voice chat	Skype, Headset with microphone
5.	File transfer	Skype
6.	Documentation (Control Program, I/O listing)	RSLogix® 500 and the website for experiments
7.	Graphical interfaces displaying operational status data	Labview®, Field point modules (NI)
8.	Remote control of the system (access to the controller)	RS Logix® 500, RS Linx®, TAMU VPN
9.	Sensors external to the system	Pressure senor, Load cell

The schematic layout of the capabilities on architecture-2 is shown in **Figure 6**.

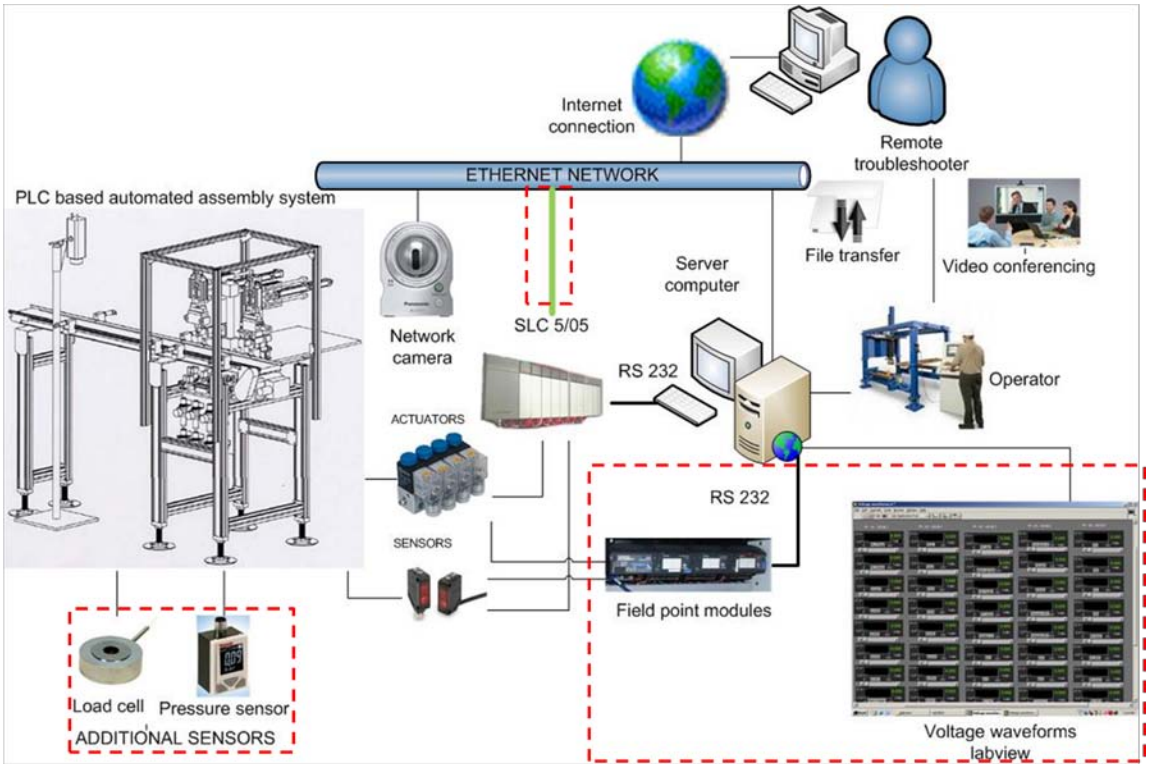


Figure 6—Graphical layout of the capabilities on architecture-2

3.3.1 Sensors external to the system

Sensors external to the system are used to facilitate detection of possible failures. These are not essential for controlling the system. Firstly, a pressure sensor is used to monitor the pressure of air going into both the robot arms and the gripper. It is configured to give a digital signal to the PLC when the pressure drops below the set point. The sensor used is a Bosch Rexroth PE5 digital pressure switch that allows two set points for pressure to be set. It is configured to give a digital signal to the PLC when the pressure drops below a set point by setting the value of the set point prior to measurement. The pressure sensor is used to monitor the pressure of air in the pneumatic line feeding both robot arms.

1. **Setpoint 1 (SP1):** The setpoint 1 is set to **1.5 bar**. This is the point when the sensor’s digital output ‘Out 1’ is energized.
2. **Reset point 1 (RP1):** The reset point is set to **1.3 bar.** When the pressure drops below 1.3 bar, ‘Out 1’ is de-energized.

A through hole load cell is fitted to the shaft on feeder #1 as shown in **Figure 7** in order to detect any jam in the feeder track. In case of a jam, the peg parts press against the feeder shaft applying force on the load cell. The load cell sends out a voltage to energize a relay that gives a signal to the PLC.

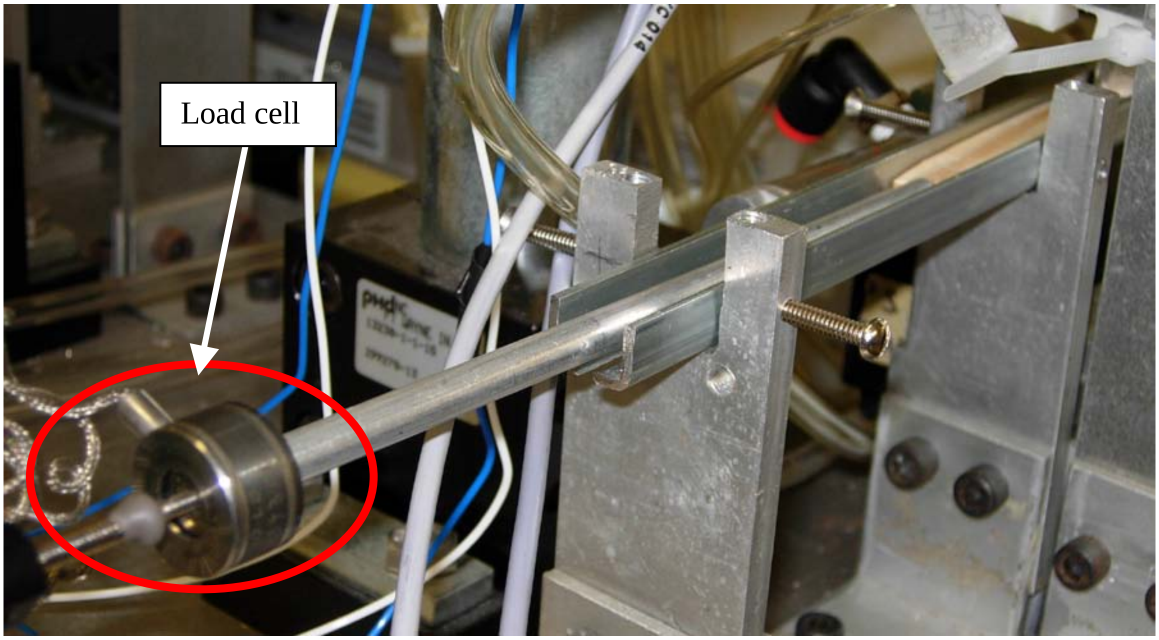


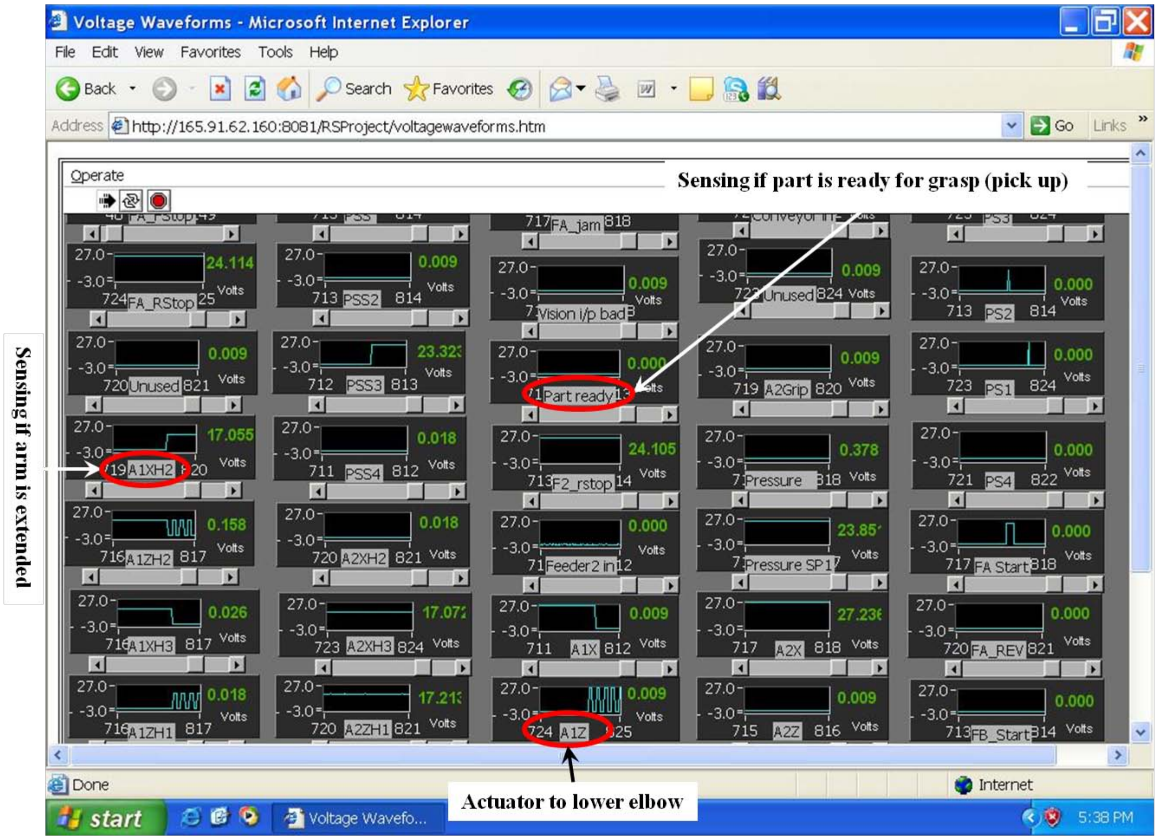
Figure 7—Load cell mounted on the shaft of feeder-1

3.3.2 Labview® based monitoring interface

Field-point modules® from National instruments® are used to monitor the voltage of the signals from all the I/Os in the system using the Labview® [47] software. The modules used are: 1) FP-1000 which manages communications between the host computer and the I/O modules via the Field-point terminal bases and 2) FP-100 analog input modules for measuring voltage coming in from all the sensors and going out to the actuators connected to the PLC. As the system is pneumatically driven, a majority of the outputs energized by the PLC are solenoid operated direction control valves. Additionally, there are relays used to turn ON/OFF the motors driving the part feeders and the conveyor.

The interface in Labview® developed in the work [48] is used to show the real time voltages of the all the inputs and outputs in the system (I/Os) as the assembly line operates. The interface depicts the time based waveforms of all the 40 I/O voltages in real time as shown in **Figure 8**. When the application is run, the waveforms show the variation in the voltages measured at each input or output as it changes to approximately 24V when energized and 0V when not energized. Beneath each wave form is a horizontal scroll bar to view the activity associated with a particular I/O over the past few cycles of the system. This interface is capable of being viewed by the remote subject via a web browser owing to the web publishing capability of Labview®. The interface can thus be accessed and controlled by remote clients in real time provided they have the Labview® run time engine installed on their computer.

An illustration of how the voltage waveforms facilitate failure diagnosis is made by considering the failure to pick part caused due to the lack of detection of the peg part by the photo sensor titled “Part ready” on the track of feeder 1. As seen in **Figure 8**, the waveform for the sensor “Part ready” does not show any activity during the operation of the system. The scroll bar beneath the waveform can be dragged to see if there was activity on one of the previous cycles. It was seen in this case that the previous cycle did register some activity as measured voltage for the sensor. The square shaped waveform for “A1Z” indicates that the elbow is oscillating in the z- direction when extended (“A1XH2”), reflective of the failure symptom.



Sensing if part is ready for grasp (pick up)

Sensing if arm is extended

Actuator to lower elbow

Figure 8—Labview® voltage waveforms

3.3.3 Online verification with the PLC

The client computer is equipped with the RSLogix® 500 software to program and to monitor by going online with the PLC and the RSLinx® to set up Ethernet communication with the PLC. The virtual private network created by VPN software creates a platform for the client to access the PLC via the Ethernet. The RSLogix® 500 software provides a powerful interface for diagnosis with features such as real time monitoring of the status of the inputs and the outputs while looking at the logical relationship between outputs and the inputs real time. The only disadvantage being that at any given time, it is possible to view only a few rungs of the ladder on the display screen. The software also provides the ability to force outputs ON individually, but the status of the inputs affected by this forcing can be observed only if they are on the rungs adjacent to the output owing to the viewing limitations on the screen.

3.4 Remote diagnosis architecture-3

This includes the capabilities of architectures 1 and 2 along with some additional features of levels 2 and 3 presented in [2] like hierarchical monitoring interface of the process, video playback, historical record events, remote desktop. The tools used to implement the capabilities are tabulated in **Table 7**.

Table 7—Summary of capabilities on architecture- 3

S No.	**Capability**	**Tool**
1.	Web camera (video feedback)	Network camera, TAMU VPN
2.	Textual Communication (chatting, email messages)	Skype
3.	Video conferencing (local operator and remote expert)	Skype
4.	Voice chat	Skype, Headset with microphone
5.	File transfer	Skype
6.	Documentation (Control Program, I/O listing)	RSLogix® 500 and the website for experiments
7.	Graphical interfaces displaying operational status data	Labview®, Field point modules (NI)
8.	Remote control of the system (access to the controller)	RS Logix® 500, RS Linx, TAMU VPN
9.	Sensors external to the system	Pressure senor, Load cell
10.	Video playback	Captured videos broadcast on youtube
11.	Stage diagram: Automated failure detection	Visual Basic 6.0, serial communication
12.	Access to the server computer (remote desktop)	VNC remote desktop
13.	Record historical data	Visual Basic, Labview®, Microsoft excel

The schematic layout showing the capabilities on architecture-3 is shown in **Figure 9**.

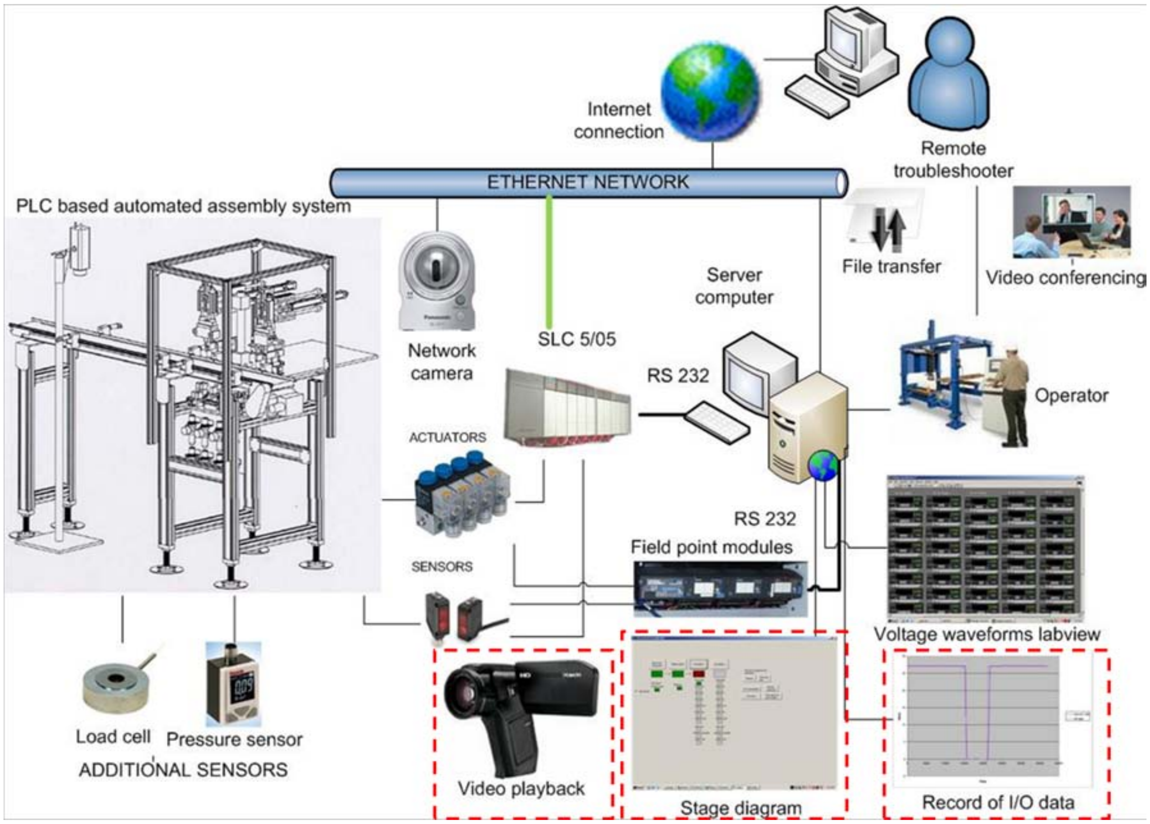


Figure 9—Graphical layout of the capabilities on architecture-3

3.4.1 Video playback

The authors in [21] recommend that video cameras can help in the detection of intermittent problems and timing problems. Video record can be used in conjunction with a record of process control events for detecting and localizing intermittent faults. The video of the failed operation was recorded using a camera and hosted on youtube [49] to be viewed by the remote expert. The link for viewing the video was given by the operator when asked for by the troubleshooter. The video recordings allow a troubleshooter to replay the video at any time thereby reducing the necessity to run the system to view the failed operation.

3.4.2 Stage diagram

In order to understand complex systems, a low resolution level of detail is required so that a person can see the overall system as well as the details of individual components [21]. The major problem with ladder logic is that, the only view of the system the troubleshooters can see is specific detail of a very small portion of the process control logic. Many ladder logic programs can be hundreds of rungs long but the screen can typically show eight ladder rungs at a time. This makes it difficult for a troubleshooter to obtain an overall understanding of the process and keep track of specific events. A stage by stage representation of an automated system was proposed in [31] as part of the remote diagnosis architecture. Apart from representing the current state of the process, the interface helps localize the part of the sequence that has failed. Keeping this in mind, the stage diagram of the system was developed that delivers the following capabilities that are summarized in **Figure 10**:

1. Overview of the process with real time representation of the process sequence.
2. Represents which stage of the process sequence the system is in and if a particular event in a sequence of events is success or a failure. This helps a troubleshooter identify the segment of the process that is faulty.
3. Ability to force the actuators in the system individually.

The interface was created in Visual Basic (VB) in which, the MSComm object was used to achieve communication between VB and the PLC [50]. The interface capitalizes on the discrete nature of the automated system in which all the control variables are discrete quantities. All the inputs (sensors) and the outputs (actuators) of the system are physically connected to the I/O modules of the PLC and the PLC's communication with the VB interface is through the serial RS 232 communication port. In order to view the stage diagram in real time, the remote subject was given viewing access to the desktop of the server computer using the VNC remote desktop software[47].

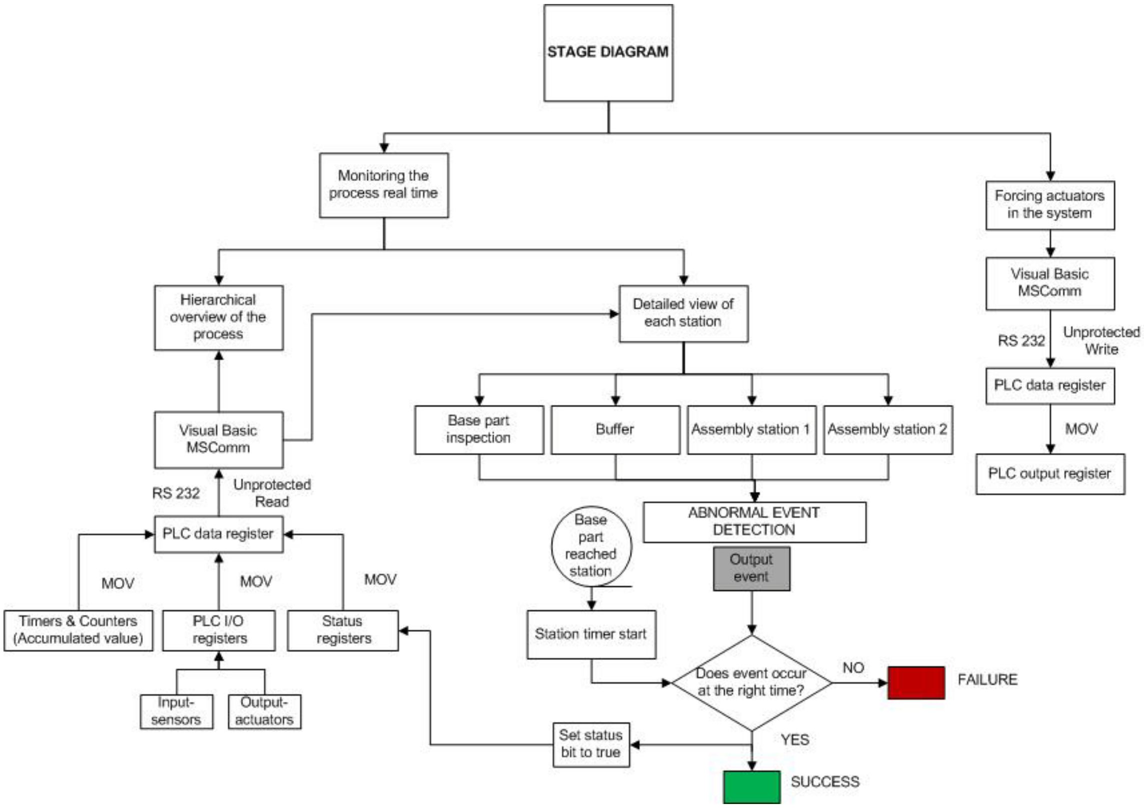


Figure 10—Methodology of stage diagram

3.4.2.1 Serial communication between VB program and PLC

The serial communications protocol is used to pass commands, information and data between the PLC and a PC or any slave devices connected to the PLC. For the purpose of data transfer, the unprotected read and the unprotected write command are used.

The basis for communication between the VB program and the PLC given in [50], capitalizes on the two commands namely:

1. Unprotected Read: This command is used to read data from locations in the PLC’s memory.
2. Unprotected Write: This command is used to write data t0 certain locations in the PLC’s memory.

The two commands listed above permit access only to the PLC's data memory. The maximum amount of contiguous memory locations that can be read with a Read command is 122 words. Similarly, a write command permits 121 contiguous words to be written with a single write command. In case of an SLC 5/05 PLC, the data memory is the N9 register. N9:0, N9:1, etc. The data register refers to the memory location in the PLC that is used to store the data regarding the status of all the inputs and the outputs in for use by the VB interface.

3.4.2.2 Communication procedure

The MSComm object in the VB software provides the serial interface for our set up. It is configured for a baud rate of 19200 baud and 8 bit data stream with one stop bit. The channel for communication will be channel 0 of the serial port which is COM1. The request is sent by the MSComm and it waits for an acknowledgement from the PLC. Once an acknowledgement is received the next step initiates else, the request is resent.

3.4.2.3 Principle of operation

The PLC program is supplemented with MOV commands to transfer data to and from the data registers N9, which correspond to the desired memory location in the PLC. The status of the I/Os are fed back to the interface as the I/O registers' contents are continually transferred to the PLC's data registers by the program running in the PLC.

During the unprotected read operation, MOV commands are used to transfer the data contained in the input registers I:5.0 and I:4.0 to the registers N9:2 and N9:3 respectively and also the data contained in the output registers O:2.0 and O:2.1 to the registers N9:4 and N9:5. This data is used to represent the status of the I/Os on the interface. Each register mentioned above is a group of 16 bits or one word. Along with these, several other data registers are used during the Read operation to transfer the accumulated values of all the timers in the process.

When an input or output is high, the corresponding bit in the data registers is set, since the contents of the I/O registers are transferred to the data registers by the program. This activates appropriate shape elements (rectangular indicators) under appropriate labels on the interface.

During the unprotected write operation, the command transfers the data from N9:0 and N9:1 to the output registers O:1.0 and O:2.0 respectively of the PLC. N9:0 and N9:1 are written to by the interface while the outputs are forced. All the I/Os on the interface are a combination of labels and shape elements. When forcing outputs, clicking the appropriate label does two things:

1. Changes the color of the shape element to green signifying it is activated.
2. Sets the appropriate bit in one of the two data registers N9:0 or N9:1 depending on the output which is activated.

The program running in the PLC transfers this change in status of the bit in the data register to the output registers causing the corresponding output to be set thereby energizing an actuator.

3.4.2.4 Working

In **Figure 11** once the ‘AutoRead’ is checked, the VB application starts to read data from the PLCs data registers. The initial screen is a low resolution mapping of the process. Clicking on one of the buttons (‘Base Part Inspection’ or ‘Buffer station’ or ‘Assembly1’ or ‘Assembly2’) opens a more detailed view containing the status of all the I/Os at that station. From the detailed view of station 3 (first assembly station) shown in **Figure 12**, it is possible to see the real time status of all I/Os at that station and the accumulation of timing and or counting elements through the logical sequence.

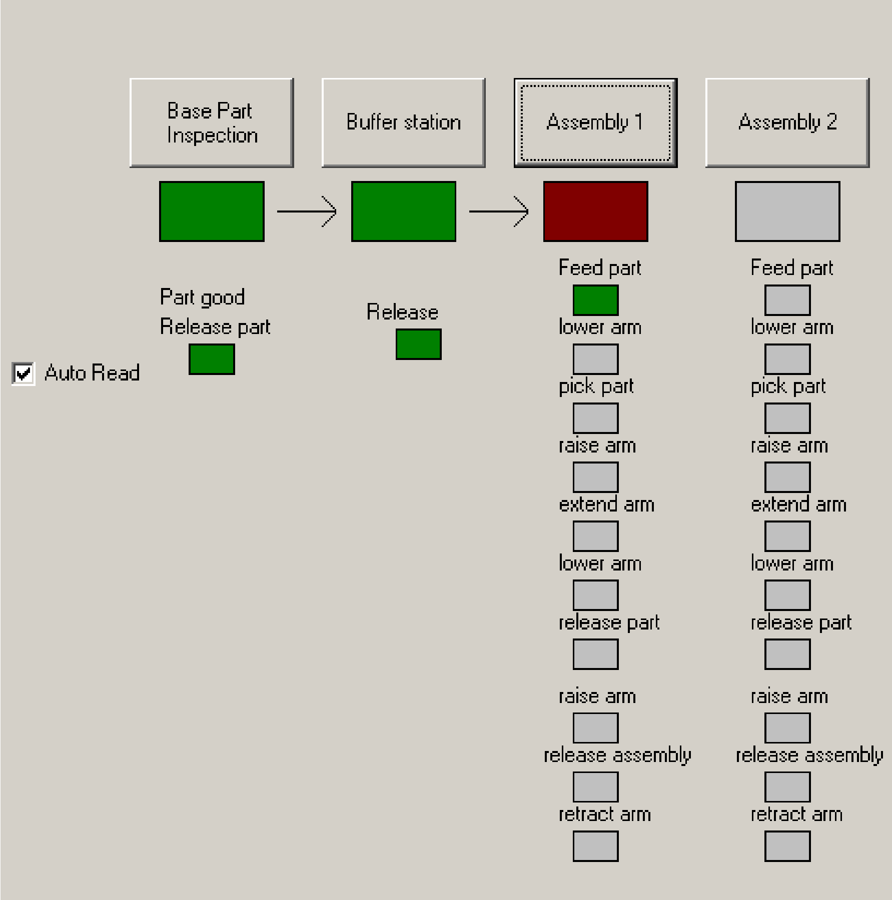
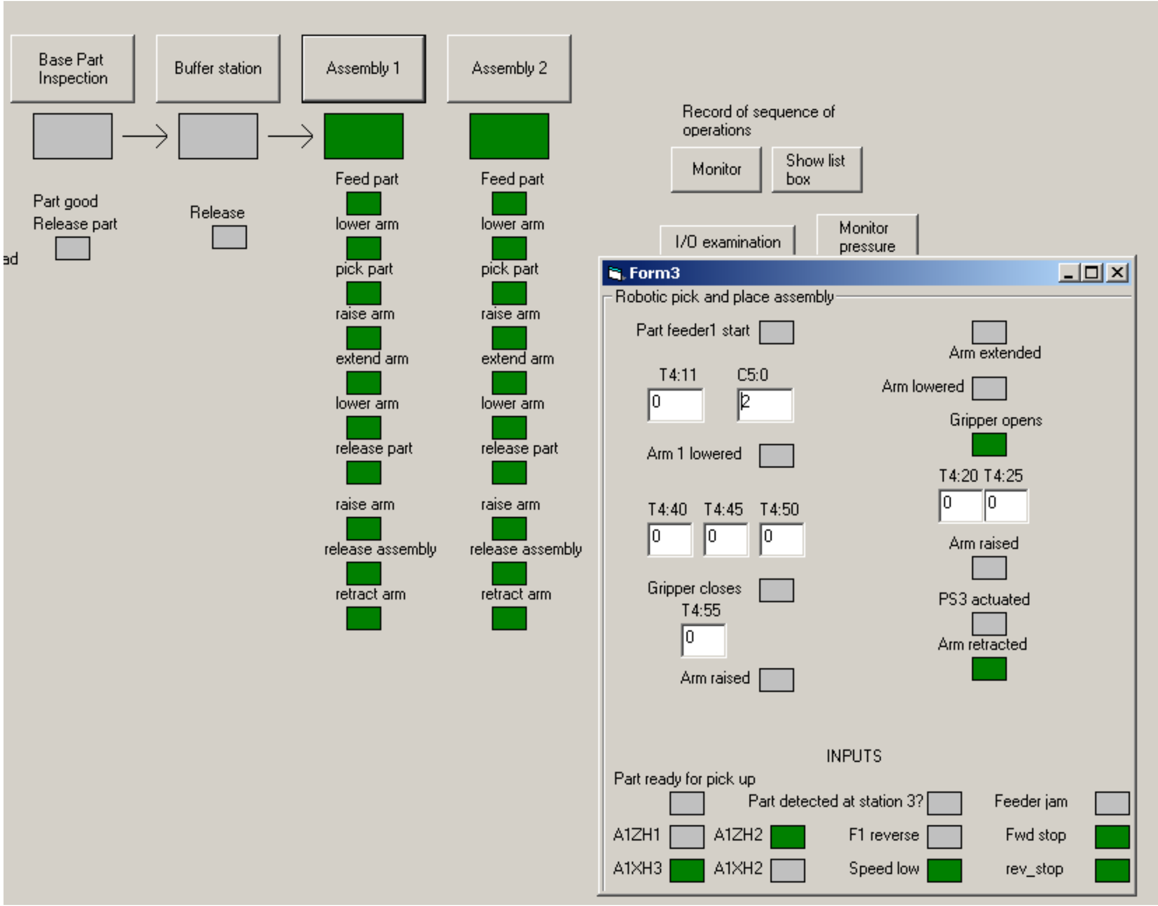


Figure 11—Stage diagram in operation when part is at station-3

Each station in the assembly line is represented by a large rectangular indicator beneath the button for the station. Additionally, there are smaller indicators corresponding to every event in the sequence of operations at each station. When the base part moving along the conveyor reaches a station, the large indicator at that particular station turns red indicating that the process at that station is active. Once the base part leaves the station and all the outputs corresponding to that station have been energized in accordance with the control logic, the large indicator turns green. **Figure 11** above is a screenshot of the interface during operation depicting that the events at station1 and station2 were successfully implemented and station3 is currently active. In **Figure 12**, it is illustrated that all the smaller indicators under the events for Assembly1 and Assembly2 turned green indicating a successful execution of all the events in the assembly operations at both the stations.



T4:11 0
C5:0 2
T4:40 0
T4:45 0
T4:50 0
T4:55 0
T4:20 0
T4:25 0

Figure 12—Stage diagram with detailed view of assembly station-1

3.4.2.5 Abnormality detection

If any of these smaller indicators at the assembly stations turns red, it is an indication of an abnormality. In conjunction with this, the large indicator for the station on the assembly line containing the failed event turns grey. The identification of an abnormal event is timing based. An event is considered to be abnormal if it does not occur at the time it was supposed to occur. The time at which any particular event at each station occurs after the base part reaches the station is predetermined. During the operation, the ladder logic in the PLC verifies if each of the events occurs as timed. In case of an abnormal event (non-occurrence of the event/ mis-timed occurrence), the PLC transmits a message to the interface and that particular indicator is turned red indicating a failed operation. In order to illustrate this, consider the event "pick part" which corresponds to the closing of the gripper. **Figure 13** shows the part of the ladder logic for detecting the abnormality of the event: closing the gripper on robot arm 1. T4:9 is the timer that is triggered when the base part reaches station 3, the first assembly station detected by the part stopper sensor at that station. Within 5 seconds of this timer, the output to close the gripper, "A1_GRIPPER" must be energized for a period of 4 seconds. If this is true, then the bit "N9:26/4" is latched indicating that the event was executed successfully. Else, the indicator under "pick part" on **Figure 12** would turn red denoting an abnormality. The bits in the PLC are read by the VB program and used to notify the operational state.

In order to understand how the stage diagram aids diagnosis, let us consider the scenario in which the robot arm fails to lower when extended to insert the peg into the base part. The cause of this failure is the loss of input from the sensor "A1XH2" which senses that the arm is extended along the x axis. The interface corresponding to this failure is depicted on the stage diagram as shown in **Figure 14**, where in the indicator below the tag "lower arm" has turned red indicating an abnormal event. Also, the large indicator for the station 'Assembly1' has turned grey after the base part has passed station 3 indicating an incomplete process. From the detailed view of station 3, it is possible for the remote expert to view the real time status of the input A1XH2, which is never activated (indicator remains grey during the complete cycle). This helps him to isolate the problem to a sensor malfunction which is later confirmed from the time based record of events.

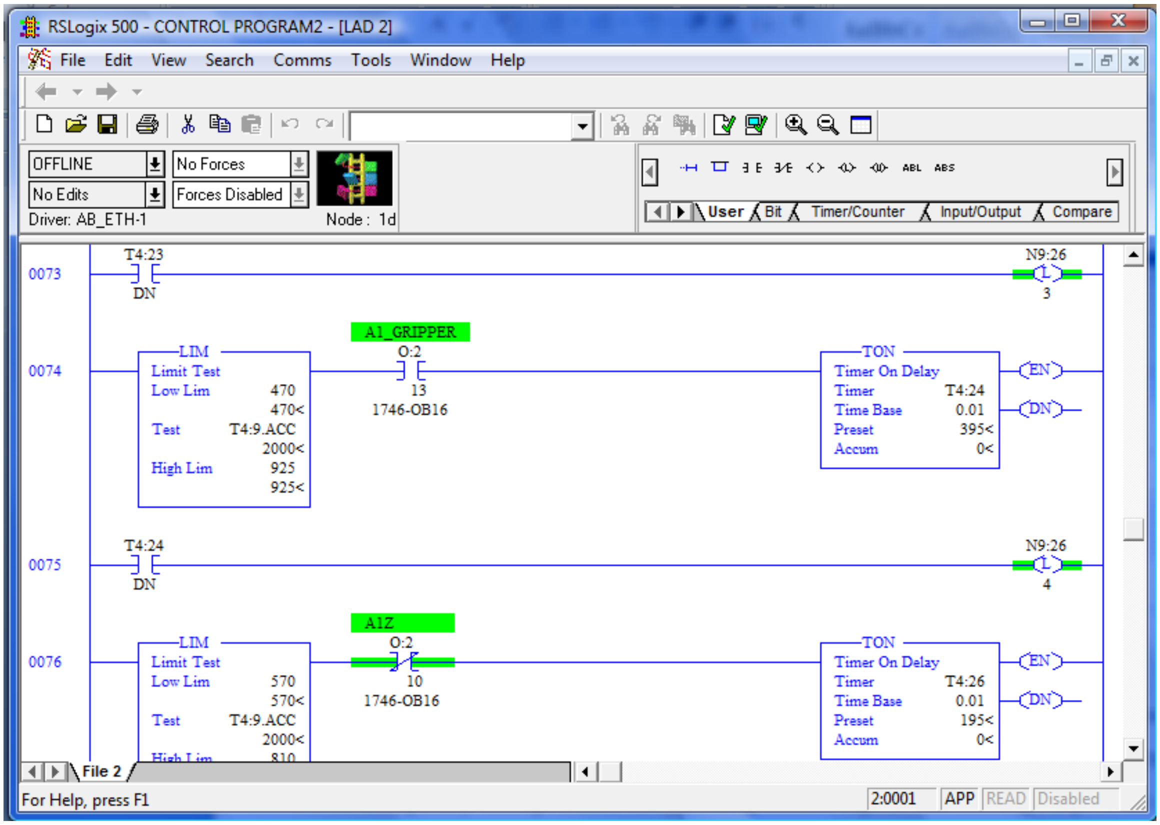


Figure 13—Abnormality detection logic

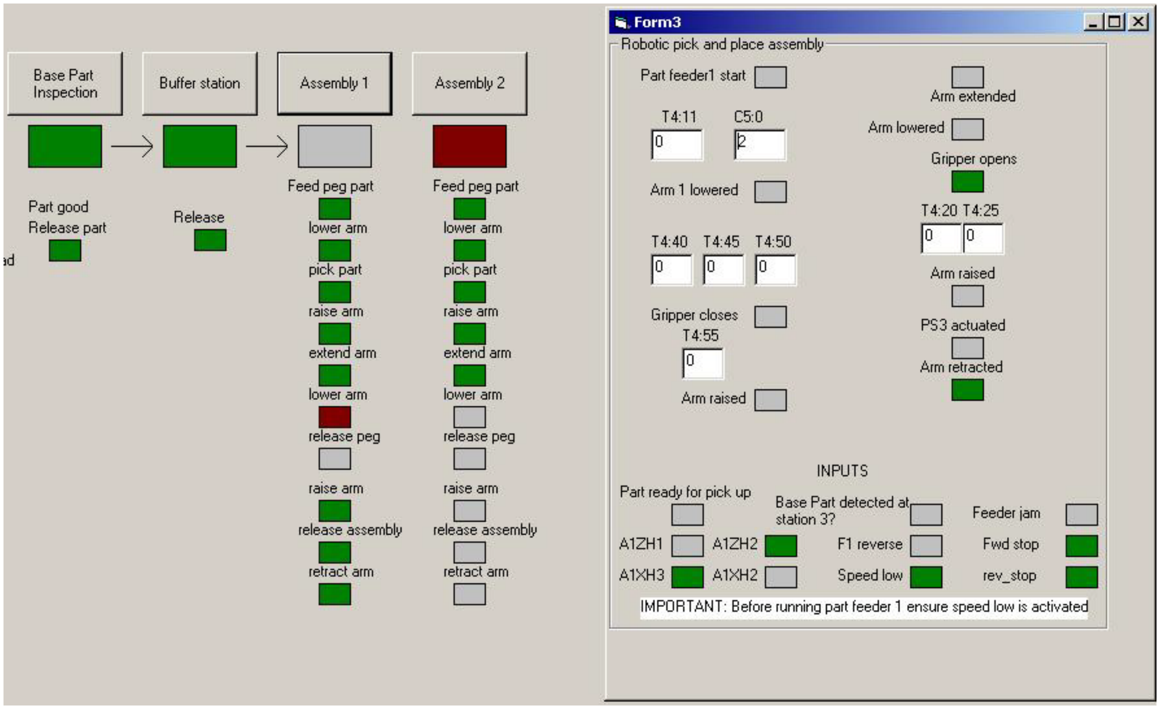
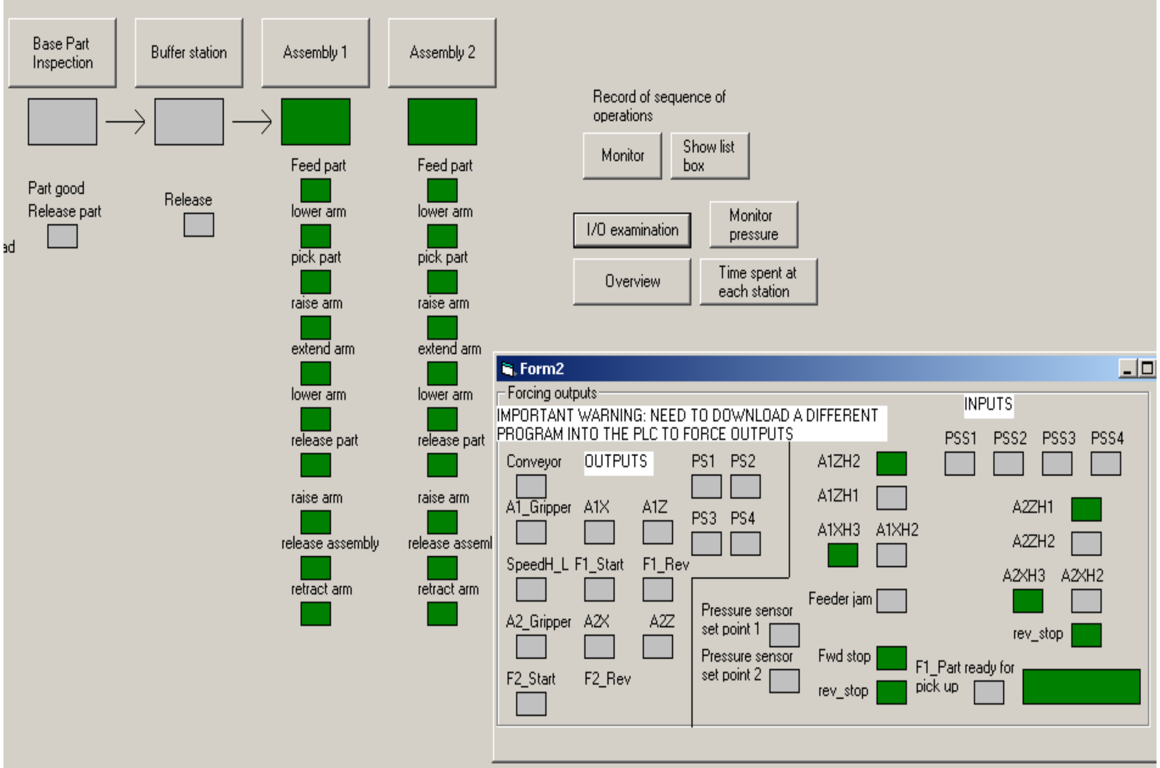


Figure 14—Stage diagram for failure to lower arm

3.4.2.6 Forcing outputs

Making temporary changes to the process like bypassing certain inputs and forcing outputs on or off is an important part of troubleshooting, especially for difficult troubleshooting experiences [21]. The interface that can be used for forcing the actuators manually is shown in **Figure 15**. While the right half of the form displays the current I/O status, the left half of the form is used for forcing the outputs by clicking the label above the output that is desired to be energized. The inputs and the outputs have been placed beside one another in order to be able to see how forcing outputs affects the inputs if any. E.g. forcing the robot arm to extend along the x axis will de-activate the hall effect sensor “A1XH3” (to sense that the arm is retracted) and activate “A1XH2” (to sense that the arm is extended).



IMPORTANT WARNING: NEED TO DOWNLOAD A DIFFERENT PROGRAM INTO THE PLC TO FORCE OUTPUTS

Figure 15—Forcing outputs in VB and examining I/Os

3.4.3 Time based record of events

Many process controllers lack the ability to record and archive a log of process control events [21]. Sometimes, the events surrounding a failure happen so quickly that it is difficult for a person to process all the knowledge about process control events. Without a way to record these events, the troubleshooter must try to replicate the failure (if possible) or work with only the subset of information that he is able to remember.

The sequence of events is recorded in two ways:

Record of I/O status: The digital status of the inputs and the outputs is recorded by the Visual Basic program interfacing with the PLC. The data from the PLC's I/O registers is in the form of bits. As the assembly line runs, several inputs and outputs energize and de-energize at different points of time. So, when these bits are written to a file by the VB program, the corresponding timestamp is also saved. As a result, it is possible to view the history of the status of any I/O in the system, in other words, to see at what point of time, a particular output or input was energized or de-energized. The data samples in time are taken at a rate determined by the MSComm object in VB.

Record of I/O voltages: As discussed in architecture-2, the Labview® interface obtains analog field voltages by communicating with the field point modules. This data for each input and output of the system is saved along with the timestamp. The I/O status refers to the binary data as seen in the PLC's I/O image table but the voltage data refers to the analog voltage received by an output device or coming from an input device. The data samples in time are taken at a rate determined by the Labview® software and the field point modules.

The data along with the timestamp for each input and output is plotted separately. These plots of the status and the field voltage are super imposed on one another. Figure 17 below, is an example of one such plot for the output "**A2_Gripper**" which refers to closing the gripper. The VB data is originally either 1 or 0 but has been scaled by a factor of 27 to facilitate comparison with the actual voltage when the output is energized (approximately 27 V). The sampling rate of the VB program is higher than that of the Labview® program, as a result of which the field voltage appears to follow the VB data.

The merit behind such data recording is that it helps identify failures in communication between and output device and the PLC due to broken wires, bad solenoids etc. As an illustration, consider the failure-3 in which the gripper on the second robot arm fails to close. The cause of this failure is a broken wire connection between the output module on the PLC and the solenoid valve that controls the flow of air to close the gripper. As seen in **Figure 16**, the status of the output “A2_GRIPPER” has been energized between 25 seconds and 29 seconds. But this is not matched with a corresponding rise in the voltage as measured by the field point modules. The plot for the ideal scenario in which the gripper closed is shown in **Figure 17**. This enables the troubleshooter to isolate the cause down to a connection issue.

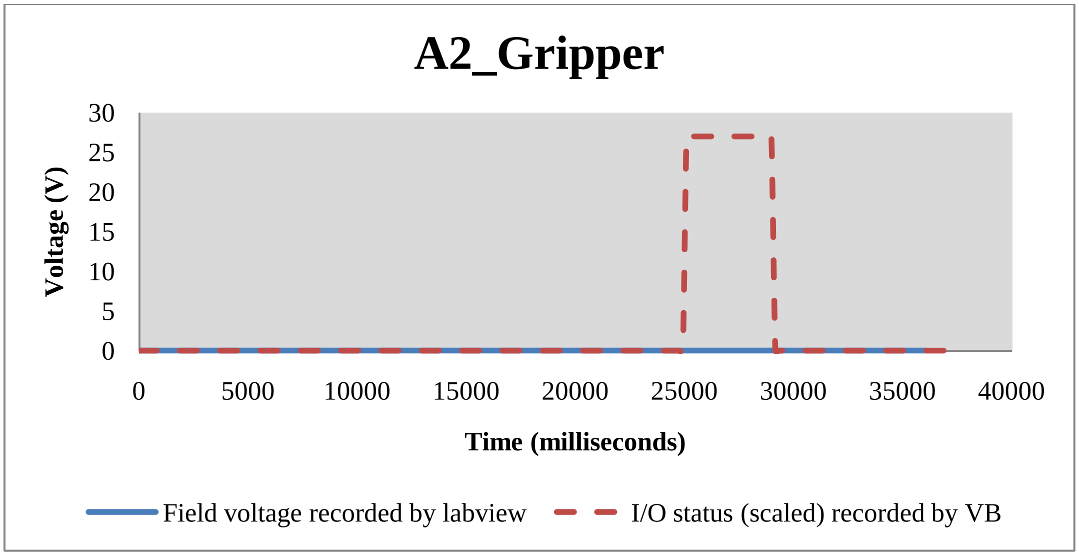


Figure 16—Record of event - failure to close gripper

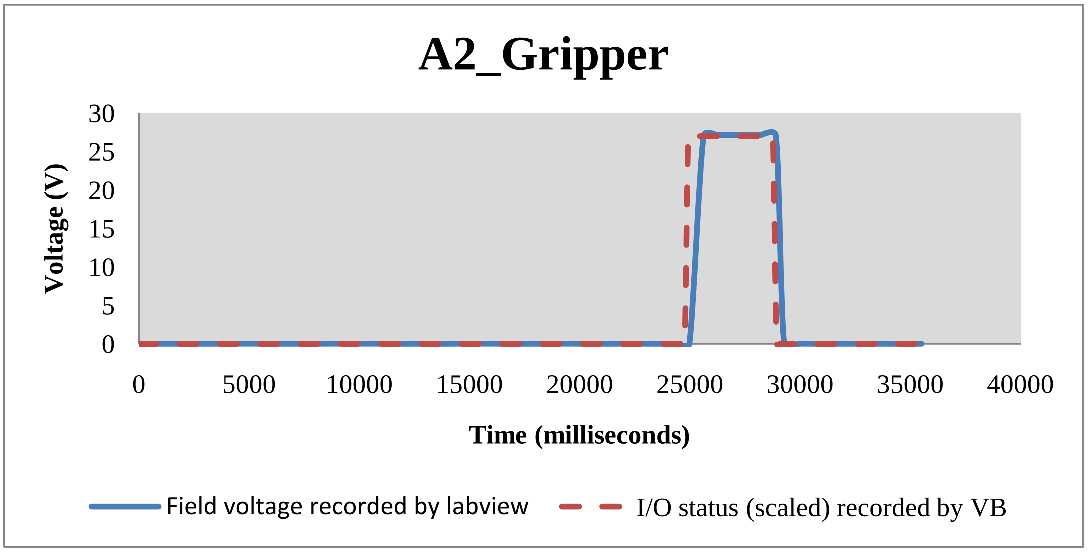


Figure 17—Record of event-close gripper

CHAPTER IV

DESIGN OF EXPERIMENTS AND ANALYSIS OF RESULTS

This chapter talks about the design of experiments and analysis of the experimental results.

4.1 Experimental objectives

In order to determine how remote diagnosis architecture facilitates troubleshooters to perform remote diagnosis and to understand the impact of the factors on remote troubleshooting performance, the following objectives were established.

- To develop a model for evaluating remote troubleshooting performance under alternative combinations of failure, operator and architecture.
- To study the effect of the nature of failure diagnosed on the troubleshooting performance with a remote diagnosis architecture.
- To study the effect of the technical skill of the local operator on performance with a remote diagnosis architecture.
- To compare the troubleshooting strategies of expert and novice troubleshooters.

4.2 Experimental variables

The three input (independent) variables are identified as: remote diagnosis architectures (X1), system operators (X2) and automated system failures (X3). Dependent variables include time taken to diagnose the failure (A1), amount of searching (A2), number of diagnosis tests performed (A3) and quality of architecture (A4). A detailed description on these variables will be presented in this section.

4.2.1 Independent variables

4.2.1.1 Operator

The role played by the system operator [21] is to describe what he sees happening and how that differs from the way the system is supposed to work. He also operates the equipment so that the troubleshooter can observe the occurrence of the problem. Two scenarios can be configured:

- **Operator with low technical knowledge (novice):** Operator is merely a user of the system and has no technical background to understand the operation of the system, electrical, electronic or mechanical subsystems. Students who had not taken a course on PLCs and automation were selected.
- **Operator with sufficient technical knowledge (engineer):** Operator has the required technical background to understand the system operation and to go online with the PLC. Students who had taken a course on PLCs and automation were selected.

4.2.1.2 Architectures

Based on the categorization of e-diagnostics capabilities [2] by Sematech, three architectures representing the capabilities of the different levels of remote diagnosis were implemented. Architecture-1 incorporates the capabilities of level-0. Architecture-2 incorporates the capabilities of level-1. Architecture-3 builds on level -1 and incorporates certain capabilities from levels-2 and 3. These three architectures can be summarized as follows:

Architecture-1: This architecture has the basic capabilities enabling remote connectivity and collaboration between the troubleshooter and the operator. It’s capabilities are similar to those discussed on level-0 type architectures presented in [2] which includes video, voice transmission, still images, textual communication and secure file transfer.

Architecture-2: It has capabilities similar to those discussed on level-1 type architectures presented in [2]. It includes all the capabilities of architecture-1 and also additionally involves direct access to the PLC and near real time monitoring of operational status by means of a graphical interface.

Architecture-3: This includes the capabilities of architectures 1 and 2 along with some additional features of levels-2 and 3 presented in [2] like hierarchical monitoring interface of the process, video playback, historical record events and remote desktop.

4.2.1.3 Failures

Based on the categorization of failures discussed in literature review Section 2.3 and the commonly occurring failures in automated systems, four different types of failures were chosen to be duplicated in this work. The failures used in this study are: failure to lower robot arm caused due to a failed sensor (hardware failure), failure to pick part caused due to bad parts incapable of being sensed (combination of software failure and tolerance error), failure to close gripper caused due to wire disconnection (product failure), insertion failure or scratch purely related to the system hardware (task failure). **Table 8** and **Table 9** detail the methodology of the duplication of each of these failures along with their symptoms for the automated assembly system.

Table 8—List of failures for experimentation

Failure	**Root cause**	**Symptom**
Failure to lower arm (failure-1)	Loss of signal from the sensor (A2XH2) that detects that the arm is extended in the X direction caused due to misplacement of the sensor.	Robot grasps peg, retracts (rises) along the Z- axis and extends (along X). But does not lower the elbow when extended and releases the peg in the raised position causing the peg to fall out.
Failure to pick part (failure-2)	Bad part that cannot be detected by the sensor prior to assembly combined with ladder logic written without accounting for failed inputs.	Robot extends along the X-axis without initially lowering the elbow to grasp the peg. Once extended, the elbow raises and lowers repeatedly. The base part is never released.
Failure to close gripper (failure-3)	Communication failure between the PLC output port and the solenoid valve controlling the gripper caused due to a disconnected wire.	Gripper fingers are jammed open during the entire assembly sequence while the rest of the operations occur normally. Consequently, the peg is never grasped. The final product is output without a peg.
Insertion failure/ scratch (failure-4)	Loss of tolerance between the grasping location of the gripper and the location of the parts in the part feeder. The parts are round and are made of rubber. They push each other on the feeder track. Given minor clearance between the parts and the walls of the track, they move sideways resulting in inaccurate placement of the parts prior to grasping by the robotic gripper.	Insertion failure occurs in the form of a non- insertion of the peg into the hole similar to jamming or scratching of the edge of the hole in the base part by the peg before insertion. In the first couple of runs, the pegs under-travel the hole and in the last couple of runs the pegs over-travel the hole.

Table 9—Methodology of duplication of the failures

Failure	Duplication methodology
Failure to lower arm (failure-1)	In the industrial production line, it is possible for the sensor to fail to sense parameters [25]. The hall-effect sensor A2XH2 senses that the second robot arm is extended. This condition must be satisfied for the elbow to be lowered after it has extended with the peg in the gripper. The sensor is displaced from its position mimicking the displacement of sensors due to vibrations in an industrial assembly line. This displacement of the sensor results in the lack of an input to the PLC from the sensor when the arm is extended. Consequently, the arm is not lowered prior to releasing the peg as seen in the symptom.
Failure to pick part (failure-2)	In an industrial production line, it is possible to have parts that are not within tolerance specifications [24]. In order to mimic this failure, one of the pegs is fed without a reflective tape causing the peg to not be sensed by the photo sensor prior to pick up by the gripper. When this is combined with a ladder logic diagram written without accounting for such unexpected situations, it causes the arm to extend and repeatedly raise and lower the elbow along the Z-axis as seen in the symptom.
Failure to close gripper (failure-3)	Manufactured products of unacceptable quality occur in industrial production lines [26]. In order to mimic a failure in communication between the PLC’s output module and the actuator, the wire between the PLC’s output port and the solenoid valve controlling the gripper was disconnected. This caused the gripper to not pick the peg in the feeder while the rest of the events occurred normally as seen from the symptom.
Insertion failure/ scratch (failure-4)	Insertion failure is common in robotic assembly operations [28]. In order to mimic the scratching of the base part or non-insertion of the peg into the hole, round pegs were used that were lined one behind the other in the feeder track. They were pushed in the track for a set amount of time by the linear actuator. A certain amount of clearance was created between the feeder track and the parts. The roundness of the parts caused them to move sideways in the feeder track as they pushed one another. Since the feeding of the peg was timing based, the sideways movement of the pegs reduced the effective linear distance travelled resulting in their inaccurate placement below the gripper. So, some of the pegs scratched the edge of the hole in the base part while few could not be inserted as seen in the symptom.

4.2.2 Dependent variables

The dependent variables refer to the measures of troubleshooting performance. In these experiments, both product measures [51] that look at the end result of the problem solving task and process measures that examine the process of remote diagnosis are considered. The process measures consider the individual steps towards achieving the end result of determining the root cause of failure. The quantitative performance measures such as time taken to diagnose the failure, amount of searching and the number of diagnostic tests performed were adopted from the measures for troubleshooting performance used in [21]. A more detailed use of the process measures has been proposed for human problem solving performance in fault diagnosis tasks in [51]. Some of these measures have been used in this study and categorized as sub attributes under the quantitative performance measures. The qualitative performance measures are adopted from the tele-maintenance system design criteria proposed in [52]. The following are the set of performance measures identified.

4.2.2.1 Time taken to diagnose the failure

Time taken to diagnose the failure is measured as the time interval between the failure occurence to when it is diagnosed.

4.2.2.2 Amount of searching

- Number of information sources consulted: It is proposed in [21] that the fewer the number of information sources consulted on the path to diagnose the failure, the more localized is the subject's search. Each return to a previously viewed page constitutes as a new page. For example if a person consults information source A, then B, then A again, he has consulted a total of three information sources.
- Number of questions asked by the expert: These refer to the questions asked by the expert to the operator about the symptoms of failure, details of the system etc.

4.2.2.3 Number of diagnostic tests performed

A diagnostic test or a treatment refers to the explicit verification of a hypothesis of a cause of failure held by the troubleshooter. This could be as simple as checking the pressure in a line, requesting the operator to test the connections to a sensor, checking the status of a sensor or an actuator, adjusting the timing of an action etc. Sometimes, treatments can be performed by the troubleshooters themselves but on most occasions, they are performed by the operator upon instructions from the troubleshooter. Each time the troubleshooter requests to cycle the system, it is counted as one diagnostic test because cycling the system is done in order to support or refute some hypothesis of the failure cause held by the troubleshooter. The system is normally cycled either to recreate the failure or to verify the effect of a treatment performed.

4.2.2.4 Quality of architecture

- **Reliability**: Ideally, the reliability level of remote diagnosis architecture has to be as close as possible to the level which is reached when the diagnosis operations are performed on site. This facilitates sound decision making by the remote expert. The scoring is established as 1- unreliable, 10 - very reliable.
- **Quality of treatments performed by the operator**: This refers to the nature of treatments that the operator is required to perform. Treatments are the diagnostic tests that the troubleshooter requires the operator to perform. An architecture is good if it can reduce the quality of treatments performed by the operator. The scoring is established as 1 - demands high technical skill, 10 - very little technical skill.
- **Accessibility**: This refers to the ease of remote access of the information necessary for diagnosis, treatment. The scoring is established as 1 - worst, 10 – best.
- **Objectiveness of the information**: Information presented by an architecture is objective if it is understandable to the remote troubleshooter and explainable by the local operator. The scoring is established as 1- worst, 10- best.
- Architecture facilitates cognitive reasoning without too many manipulations. The scoring is established as 1- does not facilitate, 10- very conducive
- Requirement of a skilled operator. The scoring is established as 1- very low, 10- very high

4.3 Experimental design

4.3.1 Experimental hypotheses

H_0 stands for null hypothesis and H_1 stands for the alternate hypothesis. The following experimental hypotheses for the main effects are formulated:

1. **H_0:** There is no difference in the troubleshooting performance with the three architectures.

   H_1: Architecture-3 induces better remote troubleshooting performance than architectures-2 and 1. Architecture-2 induces better remote troubleshooting performance than architecture-1.
2. **H_0:** There is no difference in the troubleshooting performance for all the four failures.

   H_1: Failures-1, 2 and 3 induce better troubleshooting performance than failure-4.
3. **H_0:** There is no difference in the remote troubleshooting performance with engineer or novice system operators.

   H_1: The remote troubleshooting performance is better with engineer operators than novice operators.
4. **H_0:** There is no difference in the remote troubleshooting performance with expert or novice troubleshooters (students).

   H_1: The remote troubleshooting performance with expert troubleshooters is better than that with novice troubleshooter (students).

4.3.2 Experiment plan

The experiment incorporates a repeated measure 3 x 4 x 2 mixed fractional factorial design similar to the one used in [21]. Repeated measures imply that each subject goes through multiple tasks. 3 x 4 x 2 means that there are 24 combinations of 3 independent variables. The three independent variables are architecture (X1), failure being diagnosed (X2) and type of operator (X3). It is a fractional factorial design in that each troubleshooter goes through only a partial set of the 24 combinations of the three variables. In this case, each troubleshooter goes through 3 of the 24 combinations. The experiment is mixed in the sense that while each troubleshooter only goes through three of the combinations, all the 24 combinations are represented in the total set of data. The schematic layout of the experiment plan is tabulated in **Table 10**. Two such sets of 24 experiments were performed, one with experts and the other with students (novice) as remote troubleshooters.

Table 10—Schematic representation of experiment plan

Troubleshooter	**A1**	**A2**	**A3**
Expert 1	4E	2E	3E
Expert 2	1E	4N	2N
Expert 3	3N	1N	4E
Expert 4	2E	3E	1E
Expert 5	4N	2N	3N
Expert 6	1N	4E	2E
Expert 7	3E	1E	4N
Expert 8	2N	3N	1N
NOTE— 1: Failure -1, 2: Failure -2, 3: Failure -3, 4: Failure -4, E: engineer operator, N: novice operator, A1-architecture-1, A2-architecture-3, A3-architecture-3			

4.4 Experimental protocol

The subjects involved in the study are 8 novices (students) and 8 experts in the field of factory automation. Novices are considered in [53] as the least experienced subjects. The experts are experienced operators who have practiced their skills over a longer time, thereby accumulating expertise. The novice troubleshooters were among Bachelor students of Texas A&M University who had studied a course on automation and PLC based systems. Experts were among professionals in the field of automation. Each subject sequentially participated in training sessions, experimental tasks and survey sessions as follows:

A. Introduction.
B. Training tutorial about the automated system.
C. Break for 5 minutes.
D. Training about the capabilities on architecture-1 for experiment 1.
E. Performing experiment 1.
F. Survey for experiment 1.
G. Break for 5 minutes.

H. Training about the capabilities on architecture-2 for experiment 2
I. Performing experiment 2.
J. Survey for experiment 2.
K. Break for 5 minutes.
L. Training about the capabilities on architecture-3 for experiment 3.
M. Performing experiment 3.
N. Survey for experiment 3.
O. Final Survey.

A website was used in order to train the remote subjects about the systems and the capabilities on the architectures. The goals of the research were mentioned in this section. The troubleshooters were informed that their screen was recorded by the computer along with the conversation on Skype. The experimental protocol was then described. The introductory training session introduced the troubleshooter to the automated system, the components, layout, operation etc. The individual training sessions were task specific and introduced the troubleshooter to the architecture to be used for that particular experiment, the tools available to them on the architecture and the operators available to them for the particular task. The troubleshooter was allowed to get to know the tools by practicing them and implementing them.

The website created was used to launch the various tools required on each of the architectures. While the tasks were performed on the computer of the troubleshooter, the screen was recorded throughout the experiments. Once the experiments were complete the recording was stopped. There were two types of surveys, one after each task (experiment) and a final survey. The survey after each task was to understand the search strategy and the sources of difficulty in the task for the troubleshooter. The final survey was designed to get a subjective assessment of the tasks from both the troubleshooter.

4.5 Analysis of results

In this study, there are different quantitative attributes established to evaluate the troubleshooting performance. These attributes are cost attributes [54] which implies that lower the value, greater its preference. Consequently, a good performance for any scenario is one that minimizes each of these attributes. However, there are also cases where the performance measured is slightly better on one attribute but worse on a couple of others. The performance is not equally distributed over all these attributes. As a result, it is difficult to make a preference decision aggregating all these attributes at the same time while taking into account, the quality of the architecture.

Another significant lead is that all these attributes are not equally weighted. It can be seen from **Table 11** that the quality of architecture and the time taken to diagnose the failure are more important than the amount of searching required and the number of diagnostic tests performed. This presents the need for an analytical model that is capable of combining the effect of all these quantitative and qualitative measures with their weights to provide a common scale with which to compare the performance of the alternatives. For this purpose, multiple attribute decision making is used.

4.5.1 Multiple attribute decision making

Measurement of operational performance and decision making under multiple attributes are shown to be essentially the same process [55]. Multiple attribute decision making refers to making preference decisions (e.g. evaluation, prioritization, selection) over the available alternatives. In every MADM problem, there exist a finite number of alternatives that needs to be ranked or prioritized. The word alternative is synonymous with option, policy, action or candidate among others. Attributes refer to criteria upon which to evaluate the alternatives. Measurement/ evaluation of goal accomplishment, performance is achieved by means of attributes [54]. Alternatives are contrasted over the attributes in order to make meaningful recommendations. In our case, the attributes are the measures of performance and the alternatives are the combinations of architecture, operator and failure. This process involves the following steps:

- Attribute generation
- Attribute weighting
- Normalization of attribute ratings

- Establishing the overall score

4.5.1.1 Attribute weighting

The role of weight serves to express the importance of each attribute relative to others. The weights are normalized to the sum $\sum w_j = 1$. Weights can be assigned based on ranking procedure. In this study, the subjects that performed the experiments were asked to rank the performance measures in the survey post experiments. Then, these rankings were transformed into weights by using the following equation:

$$w_j = \frac{\frac{1}{r_j}}{\sum_{k=1}^{n} \frac{1}{r_k}}$$

Here r_j is the rank of the jth attribute. Once the weights are obtained from the rankings given by each expert, they were averaged over the eight experts to obtain the final weights of the performance measures. Although, the weights were also calculated from the evaluation by the students, only the experts' rating was used in the calculation of the performance score using the model. The aggregate weights for the performance measures are tabulated in **Table 11**.

Table 11—Weights of performance measures- experts and students

Attribute (performance measure)	**Average weight (%)- Experts**	**Average weight (%)- Students**
Time taken to diagnose the failure	30.863	29.933
Amount of searching	17.327	17.980
Number of diagnostic tests	17.823	24.156
Quality of architecture	33.986	27.931
Sub- attributes: amount of searching		
Number of information sources consulted	40.641	47.874
Number of questions asked by the troubleshooter	59.359	52.126
Sub attributes: Quality of architecture		
Reliability	19.133	13.625
Quality of treatments performed by the operator	11.224	14.150
Accessibility	14.201	11.453
Objectiveness of the information	16.582	24.179
Architecture facilitates cognitive reasoning	22.449	22.528
Requirement of a skilled operator	19.930	14.156

4.5.1.2 Normalization of attribute ratings

In order to compare inter and intra attributes that do not have the same units, normalization is applied to obtain common scales for attributes. The normalization process is affected by the nature of the attributes. In this respect, the attributes can be classified as either benefit attributes or cost attributes:

Benefit attributes: The greater the attribute value, the more its preference. The benefit attributes among the performance measures are:

- Reliability (closeness between remote diagnosis and onsite diagnosis) 1- unreliable, 10 - very reliable
- Quality of treatments performed by the operator 1 - demanding high technical skill, 10 - very little technical skill
- Accessibility (ease of remote access) of the information necessary for diagnosis, treatment 1 - worst, 10 - best
- Objectiveness of the information (explainable and understandable) 1- worst, 10- best

- Architecture facilitates cognitive reasoning without too many manipulations 1- does not facilitate, 10- very conducive

Cost attributes: The greater the attribute value, the less its preference. The cost attributes among the performance measures are:

- Requirement of a skilled operator 1- very low, 10- very high.
- Time taken to diagnose the failure
- Amount of searching and its sub attributes
- Number of diagnostic tests

Normalization of benefit attributes is done using the linear normalization procedure that divides the ratings of a certain attribute by its maximum value. The normalized value of x_{ij} is given as:

$$r_{ij} = x_{ij} \, / \, x_j^*, \ \ (\mathrm{i} = 1, \ldots\ldots \mathrm{m}, \mathrm{j} = 1, \ldots . \mathrm{n})$$

Here x_j^* is the maximum value of the jth attribute. It is clear that $0 \leq rij \leq 1$ and the value is more favorable as the value of r_{ij} approaches 1.

Cost attributes can be transformed to benefit attributes by taking inverse ratings (i.e. $\frac{1}{x_{ij}}$). Then, the transformed benefit attribute (from cost) follows the same normalization process as for the benefit attribute.

4.5.1.3 Establishing the overall score

Scoring the alternatives with respect to the attributes is achieved with the simple additive weighting method (SAW) [54]. This method obtains an index by adding the contributions from each attribute. The common numerical scaling system obtained by normalization permits addition among attribute values with different units. The total score for each alternative then can be computed by multiplying the comparable rating for each attribute by the importance weight assigned to the attribute and then summing these products over all the attributes. The value for any alternative lies between 0 and 1 and can be formulated as

$$V_i = \sum_{j=1}^{n} w_j r_{ij} ,......i = 1,....m$$

Here r_{ij} is the comparable scale of x_{ij}, which can be obtained by a normalization process and the overall remote troubleshooting performance score obtained is such that $0 \leq V_i \leq 1$. The SAW method is used to calculate the overall performance scores that are tabulated in **Table 12**. The higher the performance score, better the alternative. The remote troubleshooting performance score obtained using the model is used to evaluate the variation in troubleshooting performance with respect to failures, operator and architecture. The scores from the model illustrate that architecture 3 induces the best remote troubleshooting performance among all the three architectures. This is due to the fact that architecture 3 has the most advanced configuration on hardware and software components.

Table 12—Overall performance scores- model output for experts and students

Failure	Architecture-1	Architecture-2	Architecture-3	Operator
Experts				
1	0.505*	0.474	0.517	Engineer
1	0.362	0.389	0.576	Novice
2	0.295	0.631	0.733	Engineer
2	0.277	0.421	0.490	Novice
3	0.366	0.369	0.579	Engineer
3	0.232	0.618	0.593	Novice
4	0.234	0.437	0.342	Engineer
4	0.280	0.282	0.361	Novice
Students				
1	0.320	0.454	0.635	Engineer
1	0.366	0.361	0.370	Novice
2	0.352	0.434	0.472	Engineer
2	0.354	0.493	0.487	Novice
3	0.316	0.414	0.490	Engineer
3	0.634	0.389	0.491	Novice
4	0.276	0.314	0.365	Engineer
4	0.226	0.338	0.346	Novice
NOTE—* indicates that failure was incorrectly diagnosed.				

4.5.2 Statistical analysis

Statistical analysis of the data involving all the four failures, architecture, operator and troubleshooter is performed using the general linear model of analysis of variance (ANOVA) in order to test if the different levels of any of the factors are statistically different in terms of the various performance metrics. Despite the variation between the categories of expert and novice troubleshooters (students), the data sets collected for experts and students were assumed to be replicates which allowed the ANOVA analysis to be performed considering the factors- architecture, failure and operator, three two-way interactions (architecture-failure, architecture-operator, failure-operator) and one three-way interaction (architecture-failure-operator). The hypotheses for the ANOVA analysis can be formulated as:

Null hypothesis (H_o): There is no significant effect of different levels of a factor on the true average performance.

Alternate hypothesis (H_1): At least one level of the factor has significant effect on the true average performance.

Under these conditions, as seen from the test statistics in **Table 13**, it is possible to support the experimental hypothesis that there is significant effect of the architecture based on the overall performance score ($p<0.05$), indicating that the performance with at least one of the levels of architectures is significantly different from that of the others. The overall performance score is calculated using the weighted sum of all the quantitative performance measures and the qualitative performance measures detailed in **Section 4.5.7**. It is also supported on the basis of both the performance score, time taken to diagnose the failure, number of times the system is cycled and number of questions asked that there is significant effect of failure diagnosed indicating that certain failures were more difficult to diagnose than the others, thus supporting the research hypothesis formulated earlier. However, the effect of the operator and the interactions between the factors are not seen to be as significant at 95% level of significance.

Table 13—Statistical testing using ANOVA

Source of variation (factor)	**T**	**PS**	**R**	**IS**	**Q**	**DT**
*Architecture (df=2)	F=0.61 p=0.549	*F=9.6 p=0.001	F=0.92 p=0.413	F=0.13 p=0.881	F=0.73 p=0.494	F=0.96 p=0.398
*Failure (df=3)	*F=13.64 p<0.001	*F=5.57 p=0.005	*F=37.6 p<0.001	F=0.61 p=0.617	*F=4.89 p=0.009	F=2.01 p=0.14
**Operator (df=1)	**F=3.62 p=0.069	F=0.73 p=0.4	F=0.09 p=0.768	F=0.12 p=0.732	F=0.11 p=0.74	F=1.04 p=0.319
Architecture × Failure (df=6)	F=1.17 p=0.355	F=0.57 p=0.747	F=0.78 p=0.597	F=0.37 p=0.888	F=0.64 p=0.701	F=1.32 p=0.286
Architecture × Operator (df=2)	F=0.08 p=0.923	F=0.38 p=0.685	F=0.6 p=0.554	F=1.73 p=0.918	F=0.94 p=0.404	F=0.17 p=0.848
Failure × Operator (df=3)	F=0.24 p=0.865	F=1.4 p=0.267	F=0.61 p=0.612	F=1.19 p=0.334	F=0.09 p=0.963	F=0.16 p=0.919
Architecture × Failure× Operator (df=6)	F=0.45 p=0.838	F=0.15 p=0.988	F=0.93 p=0.492	F=1.03 p=0.429	F=0.61 p=0.717	F=0.34 p=0.906
Error df	24	24	24	24	24	24
Total df	47	47	47	47	47	47

NOTE—*significant at 95% confidence interval.
**significant at 90% confidence interval.
T- time taken to diagnose the failure in seconds, PS- performance score calculated using the model, R- number of runs (cycling the system), Q- number of questions asked by the troubleshooter, IS- number of information sources consulted by the troubleshooter, DT- number of diagnostic treatments performed, df- degrees of freedom.

The ANOVA analysis only measures the extent to which the troubleshooting performance is affected by the factors but does not specify in detail the kind of effects that are caused by the levels of the factors and which of the means of the levels is significantly different [21]. Additionally, in the experiments performed, it was observed that there were certain combinations of the alternatives that yielded good troubleshooting performance and some that resulted in poor performance similar to the results seen by researchers in [40]. The effect of the operator and the interactions between factors although not found to be as statistically significant as the architecture and failures within the experiments performed (24 each with experts and students), there were some unusual observations observed, supported by the ANOVA analysis such as the diagnosis of failure-3 with novice operators. There was one occasion when failure-1 was incorrectly diagnosed with architecture-1. At times, abnormally high number of information sources was consulted by students with a novice operator when diagnosing failure-1 with architecture-3. Difficulty was faced in diagnosing failure-4 with the increase in capabilities of the architectures. There was reduced operator interaction with architecture-3 for some failures. The students, contrary to the experts faced difficulty using the information provided by the capabilities on architecture-3 and induced lesser performance when compared with architecture-2. So, in order to better understand the variation in troubleshooting performance with the different levels of the factors involved, the detailed statistical testing for the levels of each of the factors: failure, architecture, operator and troubleshooter is performed in the following sections.

4.5.3 Comparison between failures

The summary of the statistics of the different performance measures with the three architectures in the diagnosis of all the four failures by expert and novice troubleshooters working with engineer or novice local operators is detailed in **Table 14**. It can be understood that failure-4 was most difficult to diagnose and the increase in capabilities of the architectures did not help in the diagnosis of failure-4. Failures-1, 2 and 3 on the other hand induced similar performance with respect the various performance metrics for experts ad novices.

Table 14—Summary statistics for the performance measures over all the four failures

Parameter	**Expert**				**Novice**			
	F1	**F2**	**F3**	**F4**	**F1**	**F2**	**F3**	**F4**
T(mean)	1028.1	1133.1	1132.8	2229.1	1211.2	1099.8	958.8	2010.5
T (sd)	166.6	506.4	696.7	562	598.1	149	294.8	462.9
PS (mean)	0.470	0.474	0.459	0.323	0.418	0.432	0.456	0.311
PS (sd)	0.081	0.182	0.158	0.072	0.115	0.064	0.109	0.051
R (mean)	3	2.7	4.7	15.8	4	2.7	3.8	18.3
R (sd)	1.5	1.6	2.7	9.1	2.3	1.2	1.5	2.6
Q (mean)	4.1	7.3	7.2	16.1	9.8	8	5.8	14.3
Q (sd)	3.5	4.3	6.3	8.7	5	4.5	3.7	8.2
IS (mean)	20.3	31.5	19.8	26.1	45.1	41.5	31.3	34
IS (sd)	12.5	14.3	10	14.2	34.6	26.7	12	6.6
DT (mean)	3.3	3	4.7	4.7	3.8	2.5	4.7	4.2
DT (sd)	0.8	2	4	2.3	1.3	1	1.2	1.6

NOTE—T- time taken to diagnose the failure in seconds, PS- performance score calculated using the model, R- number of runs (cycling the system), Q- number of questions asked by the troubleshooter, IS- number of information sources consulted by the troubleshooter, DT- number of diagnostic treatments performed, F1-failure-1, F2-Failure-2, F3-failure-3, F4-Failure-4, sd-standard deviation.

In order to test the hypothesis for the difference in mean troubleshooting performance with the four failures, the 2-sample t-test was performed with a confidence interval of 95% (α=0.05). The following **Table 15** and **Table 16** detail the hypotheses tested and the results obtained.

Failure-4 was found difficult to diagnose remotely as seen from the significant differences obtained for hypotheses tested for the difference between failure-4 and failures-1, 2 or 3 in terms of time taken to diagnose the failure, number of questions asked and the overall remote troubleshooting performance. As justified by the results of the hypotheses tested for the number of times the system is cycled to diagnose the failure, failure-4 required most number of cycles of the system to diagnose because the failure occurred in a progression during 5 runs for each set of parts. The lack of additional information from the diagnosis tools available on the architectures and the nature of the failure resulted in the system being cycled most number of times for the diagnosis of failure-4. Failures-1, 2 and 3 were related to variables of the system that are measured or monitored and were caused due to a sensor fault, actuator fault, bad parts etc as tabulated in **Section 4.2.1.3**. The capabilities on the architectures were useful and helped improve the troubleshooting performance in the diagnosis of these failures. However, failure-4 was related to the hardware of the system, i.e. configuration of the part feeder and the nature of the parts.

Much of the additional information provided by architectures-2 and 3 was found to be redundant in the diagnosis of failure-4 as seen from the main effects plot for time taken to diagnose the failures with the three architectures in **Figure 18**. Since the failure was not related to any measured or monitored variable of the system, the additional information provided was not helpful. The stage diagram representation of the system in architecture-3 could be more beneficial than online ladder logic in order to determine that the system was logically executing the steps in the right order because it conveyed the information in a single screen rather than having to observe multiple ladder rungs while the system operated. The interaction with the operator to gain his intuition and good video feedback were the most efficient tools to remotely diagnose this type of failure. The combination of the stage diagram with the tools on architecture-1 would form the ideal tool set for the diagnosis of this type of failure.

From the hypotheses tested in **Table 16** for the number of information sources consulted, no significant difference was observed between all the four failures. In the diagnosis process of all the four failures, the troubleshooters would consult the available information sources and perform diagnostic tests in order to eliminate or create their fundamental hypotheses of the causes of failure. In the diagnosis of failure-4 once the basic tests were performed and the troubleshooters were convinced that the failure was purely hardware related, they spent more time interacting with the operator and observing the failure symptom via the webcam in order to further diagnose the failure. As a result, there was no significant difference between the failures on the basis of the number of information sources consulted and in case of experts, the number of diagnostic tests performed. The increased operator interaction is explained by the hypotheses tested for the number of questions asked.

Table 15—Statistical testing for difference in mean performance with failures

Null hypothesis ($\boldsymbol{H_0}$)	**Alternate hypothesis ($\boldsymbol{H_1}$)**	**Test Statistic (experts)**	**Outcome (experts)**	**Test Statistic (students)**	**Outcome (students)**
Time taken to diagnose the failure (seconds)					
$\bar{x}_{F1} - \bar{x}_{F2} = 0$	$\bar{x}_{F1} - \bar{x}_{F2} < 0$	t = -0.48 (p=0.323) df=6	Fail to reject	t =0.44 (p=0.662) df=5	Fail to reject
$\bar{x}_{F1} - \bar{x}_{F3} = 0$	$\bar{x}_{F1} - \bar{x}_{F3} < 0$	t= -0.36 (p=0.368) df=5	Fail to reject	t=0.93 (p=0.808) df=7	Fail to reject
$\boldsymbol{\bar{x}_{F1} - \bar{x}_{F4} = 0}$	$\boldsymbol{\bar{x}_{F1} - \bar{x}_{F4}} < 0$	t= -5.02 (p=0.002) df=5	**Reject**	t= -2.59 (p=0.015) df=9	**Reject**
$\bar{x}_{F2} - \bar{x}_{F3} = 0$	$\bar{x}_{F2} - \bar{x}_{F3} < 0$	t = 0 (p=0.500) df=9	Fail to reject	t = 1.05 (p=0.835) df=7	Fail to reject
$\boldsymbol{\bar{x}_{F2} - \bar{x}_{F4} = 0}$	$\boldsymbol{\bar{x}_{F2} - \bar{x}_{F4}} < 0$	t= -3.55 (p=0.003) df=9	**Reject**	t= -4.59 (p=0.002) df=6	**Reject**
$\boldsymbol{\bar{x}_{F3} - \bar{x}_{F4} = 0}$	$\boldsymbol{\bar{x}_{F3} - \bar{x}_{F4}} < 0$	t= -3 (p=0.007) df=9	**Reject**	t= -4.69 (p=0.001) df=8	**Reject**
Overall performance score					
$\bar{x}_{F1} - \bar{x}_{F2} = 0$	$\bar{x}_{F1} - \bar{x}_{F2} > 0$	t = -0.05 (p=0.520) df=6	Fail to reject	t = -0.26 (p=0.600) df=7	Fail to reject
$\bar{x}_{F1} - \bar{x}_{F3} = 0$	$\bar{x}_{F1} - \bar{x}_{F3} > 0$	t= 0.15 (p=0.443) df=7	Fail to reject	t= -0.58 (p=0.713) df=9	Fail to reject
$\boldsymbol{\bar{x}_{F1} - \bar{x}_{F4} = 0}$	$\boldsymbol{\bar{x}_{F1} - \bar{x}_{F4}} > 0$	t= 3.32 (p=0.004) df=9	**Reject**	t= 2.08 (p=0.042) df=6	**Reject**
$\bar{x}_{F2} - \bar{x}_{F3} = 0$	$\bar{x}_{F2} - \bar{x}_{F3} > 0$	t = 0.15 (p=0.440) df=6	Fail to reject	t= -0.46 (p=0.670) df=8	Fail to reject
$\boldsymbol{\bar{x}_{F2} - \bar{x}_{F4} = 0}$	$\boldsymbol{\bar{x}_{F2} - \bar{x}_{F4}} > 0$	t = 1.9 (p=0.053) df=6	**Reject**	t=3.59 (p=0.003) df=9	**Reject**
$\boldsymbol{\bar{x}_{F3} - \bar{x}_{F4} = 0}$	$\boldsymbol{\bar{x}_{F3} - \bar{x}_{F4}} > 0$	t= 1.92 (p=0.052) df=6	**Reject**	t= 2.93 (p=0.011) df=7	**Reject**
Number of times the system is cycled					
$\bar{x}_{F1} - \bar{x}_{F2} = 0$	$\bar{x}_{F1} - \bar{x}_{F2} < 0$	t = 0.36 (p=0.637) df=9	Fail to reject	t= 1.26 (p=0.877) df=7	Fail to reject
$\bar{x}_{F1} - \bar{x}_{F3} = 0$	$\bar{x}_{F1} - \bar{x}_{F3} < 0$	t= -1.3 (p=0.117) df=7	Fail to reject	t=0.15 (p=0.558) df=8	Fail to reject
$\boldsymbol{\bar{x}_{F1} - \bar{x}_{F4} = 0}$	$\boldsymbol{\bar{x}_{F1} - \bar{x}_{F4}} < 0$	t= -3.38 (p=0.010) df=5	**Reject**	t= -10.19 (p=0.000) df=9	**Reject**
$\bar{x}_{F2} - \bar{x}_{F3} = 0$	$\bar{x}_{F2} - \bar{x}_{F3} < 0$	t = -1.54 (p=0.081) df=8	Fail to reject	t= -1.5 (p=0.084) df=9	Fail to reject
$\boldsymbol{\bar{x}_{F2} - \bar{x}_{F4} = 0}$	$\boldsymbol{\bar{x}_{F2} - \bar{x}_{F4}} < 0$	t= -3.46 (p=0.009) df=5	**Reject**	t= -13.46 (p<0.001) df=7	**Reject**
$\boldsymbol{\bar{x}_{F3} - \bar{x}_{F4} = 0}$	$\boldsymbol{\bar{x}_{F3} - \bar{x}_{F4}} < 0$	t= -2.86 (p=0.018) df=5	**Reject**	t= -11.95 (p<0.001) df=7	**Reject**
NOTE—F1-failure-1, F2-failure-2, F3-failure-3, F4- failure-4, df-degrees of freedom, $\bar{x}$- sample mean.					

Table 16—Statistical testing for difference in mean performance with failures continued

Null hypothesis (H_0)	**Alternate hypothesis (H_1)**	**Test Statistic (experts)**	**Outcome (experts)**	**Test Statistic (students)**	**Outcome (students)**
Number of information sources consulted					
$\bar{x}_{F1} - \bar{x}_{F2} = 0$	$\bar{x}_{F1} - \bar{x}_{F2} < 0$	t = -1.44 (p=0.092) df=9	Fail to reject	t= 0.21 (p=0.579) df=9	Fail to reject
$\bar{x}_{F1} - \bar{x}_{F3} = 0$	$\bar{x}_{F1} - \bar{x}_{F3} < 0$	t= 0.07 (p=0.529) df=9	Fail to reject	t=0.93 (p=0.805) df=6	Fail to reject
$\bar{x}_{F1} - \bar{x}_{F4} = 0$	$\bar{x}_{F1} - \bar{x}_{F4} < 0$	t= -0.76 (p=0.235) df=9	Fail to reject	t= 0.78 (p=0.764) df=5	Fail to reject
$\bar{x}_{F2} - \bar{x}_{F3} = 0$	$\bar{x}_{F2} - \bar{x}_{F3} < 0$	t = 1.59 (p=0.927) df=9	Fail to reject	t= 0.85 (p=0.786) df=6	Fail to reject
$\bar{x}_{F2} - \bar{x}_{F4} = 0$	$\bar{x}_{F2} - \bar{x}_{F4} < 0$	t= 0.65 (p=0.734) df=9	Fail to reject	t=0.67 (p=0.733) df=5	Fail to reject
$\bar{x}_{F3} - \bar{x}_{F4} = 0$	$\bar{x}_{F3} - \bar{x}_{F4} < 0$	t= -0.87 (p=0.204) df=9	Fail to reject	t= -0.48 (p=0.324) df=7	Fail to reject
Number of questions asked by the troubleshooter					
$\bar{x}_{F1} - \bar{x}_{F2} = 0$	$\bar{x}_{F1} - \bar{x}_{F2} < 0$	t = -1.40 (p=0.098) df=9	Fail to reject	t= 0.67 (p=0.739) df=9	Fail to reject
$\bar{x}_{F1} - \bar{x}_{F3} = 0$	$\bar{x}_{F1} - \bar{x}_{F3} < 0$	t= -1.02 (p=0.171) df=7	Fail to reject	t=1.57 (p=0.925) df=9	Fail to reject
$\boldsymbol{\bar{x}_{F1} - \bar{x}_{F4} = 0}$	$\boldsymbol{\bar{x}_{F1} - \bar{x}_{F4}} < 0$	t= -3.13 (p=0.010) df=6	**Reject**	t= -1.15 (p=0.142) df=8	Fail to reject
$\bar{x}_{F2} - \bar{x}_{F3} = 0$	$\bar{x}_{F2} - \bar{x}_{F3} < 0$	t = 0.05 (p=0.521) df=8	Fail to reject	t= 0.92 (p=0.809) df=9	Fail to reject
$\boldsymbol{\bar{x}_{F2} - \bar{x}_{F4} = 0}$	$\boldsymbol{\bar{x}_{F2} - \bar{x}_{F4}} < 0$	t= -2.23 (p=0.030) df=7	**Reject**	t= -1.67 (p=0.070) df=7	Fail to reject
$\boldsymbol{\bar{x}_{F3} - \bar{x}_{F4} = 0}$	$\boldsymbol{\bar{x}_{F3} - \bar{x}_{F4}} < 0$	t= -2.05 (p=0.035) df=9	**Reject**	t= -2.33 (p=0.029) df=6	**Reject**
Number of diagnostic treatments performed					
$\bar{x}_{F1} - \bar{x}_{F2} = 0$	$\bar{x}_{F1} - \bar{x}_{F2} < 0$	t = 0.38 (p=0.641) df=6	Fail to reject	t= 1.93 (p=0.957) df=9	Fail to reject
$\bar{x}_{F1} - \bar{x}_{F3} = 0$	$\bar{x}_{F1} - \bar{x}_{F3} < 0$	t= -0.78 (p=0.234) df=5	Fail to reject	t= -1.14 (p=0.143) df=9	Fail to reject
$\bar{x}_{F1} - \bar{x}_{F4} = 0$	$\bar{x}_{F1} - \bar{x}_{F4} < 0$	t= -1.32 (p=0.118) df=6	Fail to reject	t= -0.39 (p=0.352) df=9	Fail to reject
$\bar{x}_{F2} - \bar{x}_{F3} = 0$	$\bar{x}_{F2} - \bar{x}_{F3} < 0$	t = -0.9 (p=0.200) df=7	Fail to reject	t= -3.31 (p=0.005) df=9	**Reject**
$\bar{x}_{F2} - \bar{x}_{F4} = 0$	$\bar{x}_{F2} - \bar{x}_{F4} < 0$	t= -1.33 (p=0.109) df=9	Fail to reject	t= -2.13 (p=0.033) df=8	**Reject**
$\bar{x}_{F3} - \bar{x}_{F4} = 0$	$\bar{x}_{F3} - \bar{x}_{F4} < 0$	t= 0 (p=0.500) df=7	Fail to reject	t= 0.29 (p=0.609) df=6	Fail to reject
NOTE—F1-failure-1, F2-failure-2, F3-failure-3, F4- failure-4, df-degrees of freedom, $\bar{x}$- sample mean.					

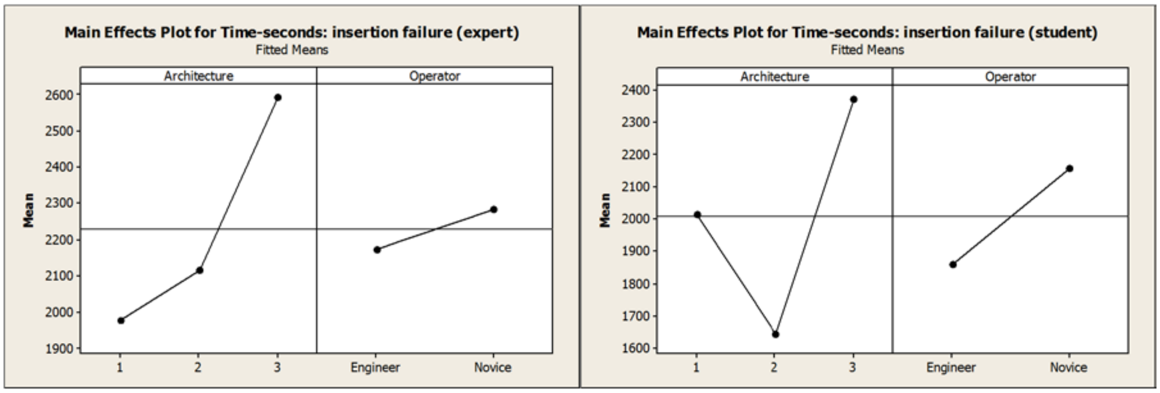


Figure 18—Main effects plot for time taken to diagnose the insertion failure- experts and students

The correlation between failure type and the various performance measures for both expert and novice troubleshooters is analyzed and the correlation coefficients are tabulated in **Table 17**. Failures-1, 2 and 3 were considered as one type while failure-4 was considered as another type because of the differences observed in the hypotheses tested for the performance measures between failure-4 and the other failures. The results obtained from the hypotheses tested for the significant reduction in troubleshooting performance in the diagnosis of failure-4 when compared to failures-1, 2 and 3 are supplemented by the moderately negative correlation coefficient for the overall remote troubleshooting performance score for both experts and novices. A moderately positive correlation between failure type and the time taken to diagnose the failure by both experts and novices supplements the fact that failure-4 took more time to diagnose in comparison to the failures-1, 2 and 3. The weak correlation between failure type and the information sources consulted and diagnostic tests performed supplements the fact that failure type does not affect these performance measures.

Table 17—Pearson correlation coefficient for failure type and performance measures

T		**PS**		**DT**		**Q**		**R**		**IS**	
E	N	E	N	E	N	E	N	E	N	E	N
0.71	0.72	-0.46	-0.55	0.17	0.15	0.61	0.47	0.76	0.96	0.07	-0.1
NOTE—T- time taken to diagnose the failure in seconds, PS- performance score calculated using the model, R- number of runs (cycling the system), Q- number of questions asked by the troubleshooter, IS- number of information sources consulted by the troubleshooter, DT- number of diagnostic treatments performed, E-expert troubleshooters, N-novice troubleshooters.											

Based on the analysis of the effect of failure, in order to study the effect of the other factors, the performance with respect to failure-4 is considered separate from that for failures-1, 2 and 3. The main effects and interactions plots for the mean time taken to diagnose failures-1, 2 and 3 for expert and novice troubleshooters are shown in **Figure 19**, **Figure 20** and **Figure 21**. The main effects plot for experts and students for mean number of times the system is cycled, diagnostic treatments performed by the operator, information sources consulted by the troubleshooter and questions asked are shown in **Figure 22**, **Figure 23**, **Figure 24** and **Figure 25** respectively. The number of information sources consulted and the number of questions asked constitute the amount of searching involved [21]. The number of times the system is cycled and the number of treatments performed by the operator together contribute to the number of diagnostic tests performed. It is important to note that all these quantitative measures are cost attributes which means that lower the value of these attributes, better the performance. The main effects and interactions plots for the performance score obtained using the model are shown in **Figure 26**, **Figure 27** and **Figure 28** respectively. In case of the model, higher the score, better the troubleshooting performance for any alternative.

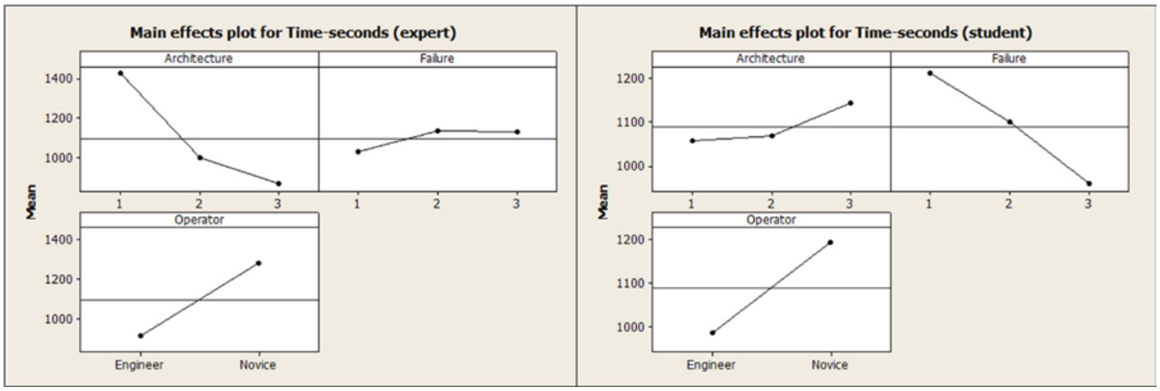


Figure 19—Main effects plots for time taken to diagnose the failure- experts and students

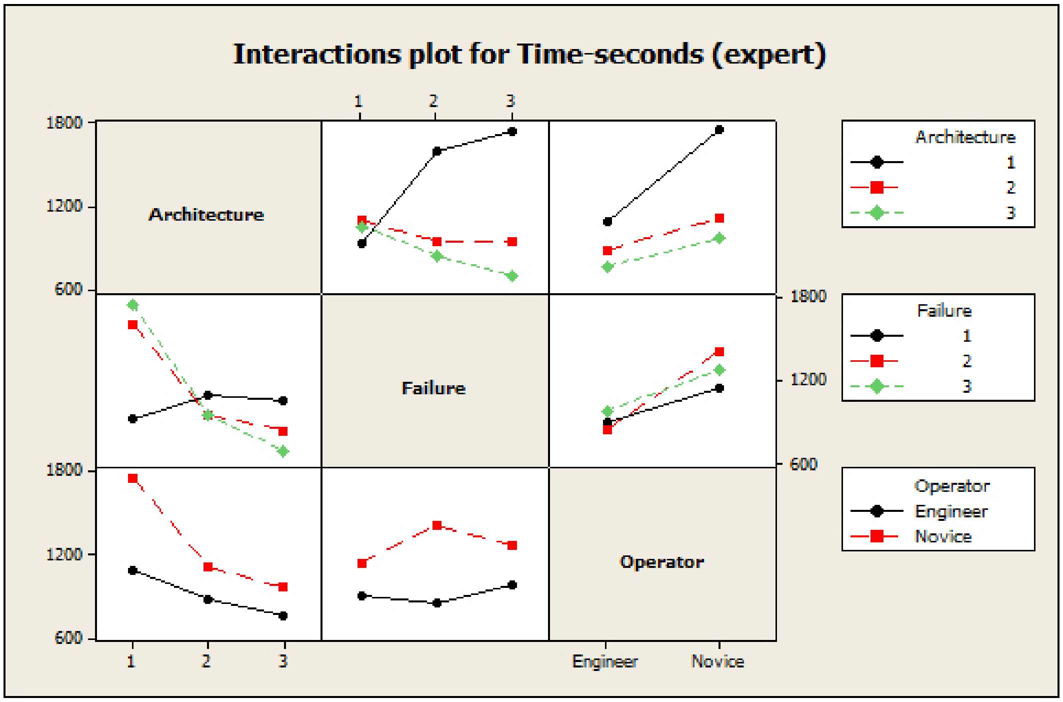


Figure 20—Interactions plot for time taken to diagnose the failure- experts

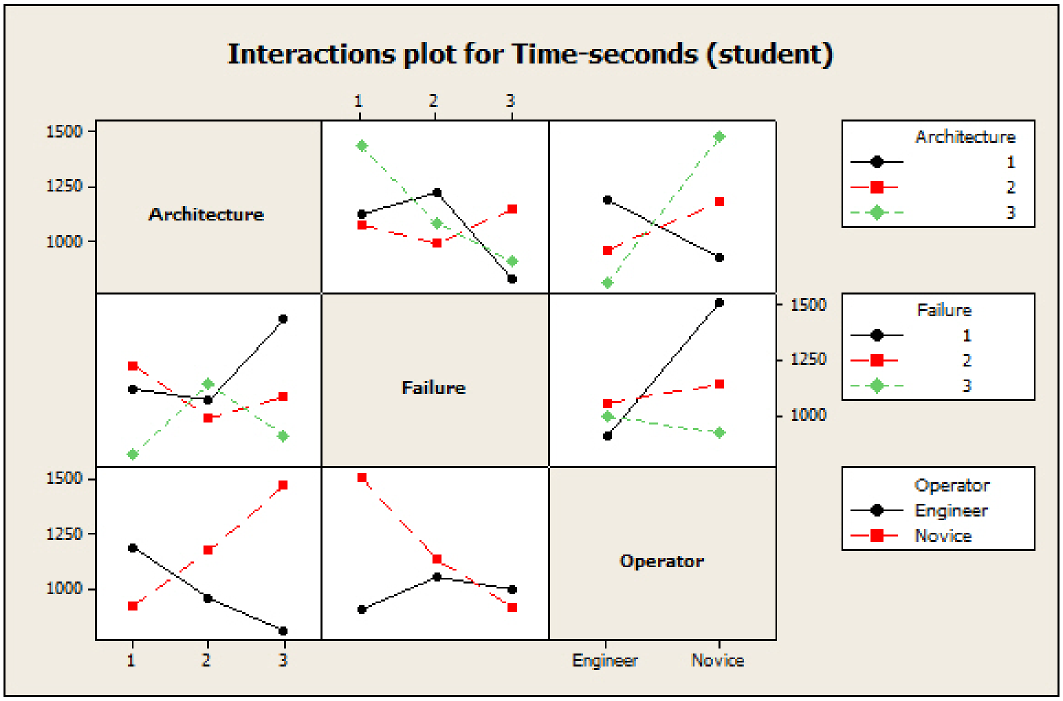


Figure 21—Interactions plot for time taken to diagnose the failure- students

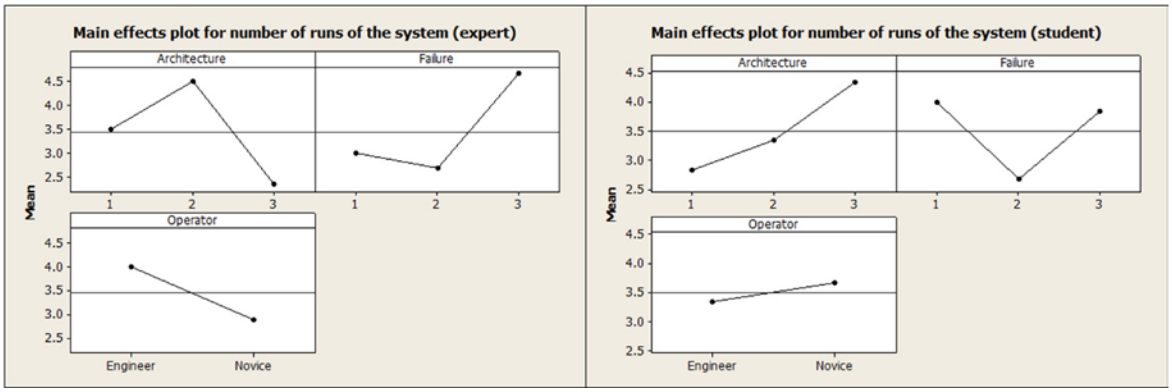


Figure 22—Main effects plot for number times the system is cycled: experts and students

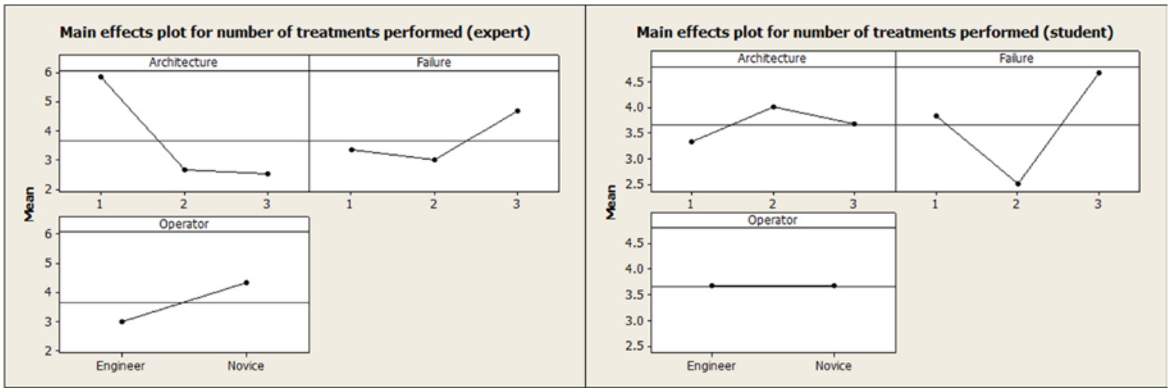


Figure 23—Main effects plot for number of diagnostic treatments performed- experts and students

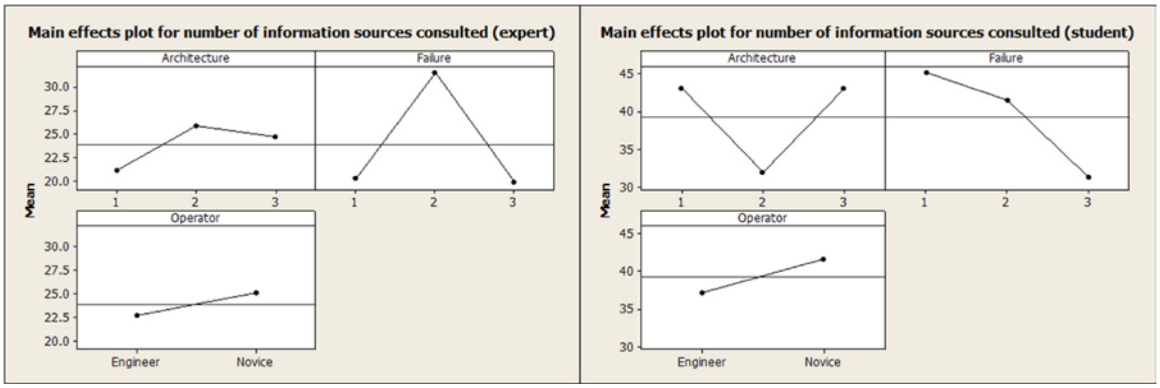


Figure 24—Main effects plot for number of information sources consulted- experts and students

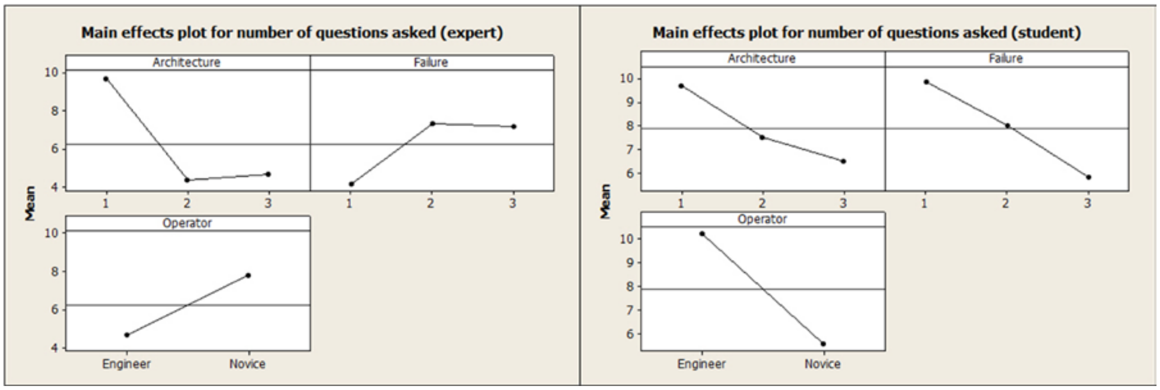


Figure 25—Main effects plot for number of questions asked- experts and students

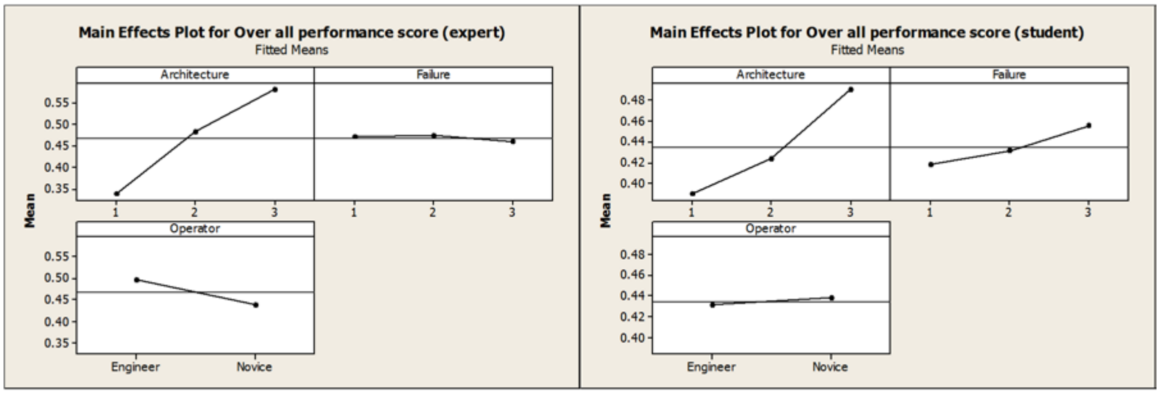


Figure 26—Main effects plot for overall performance measure- experts and students

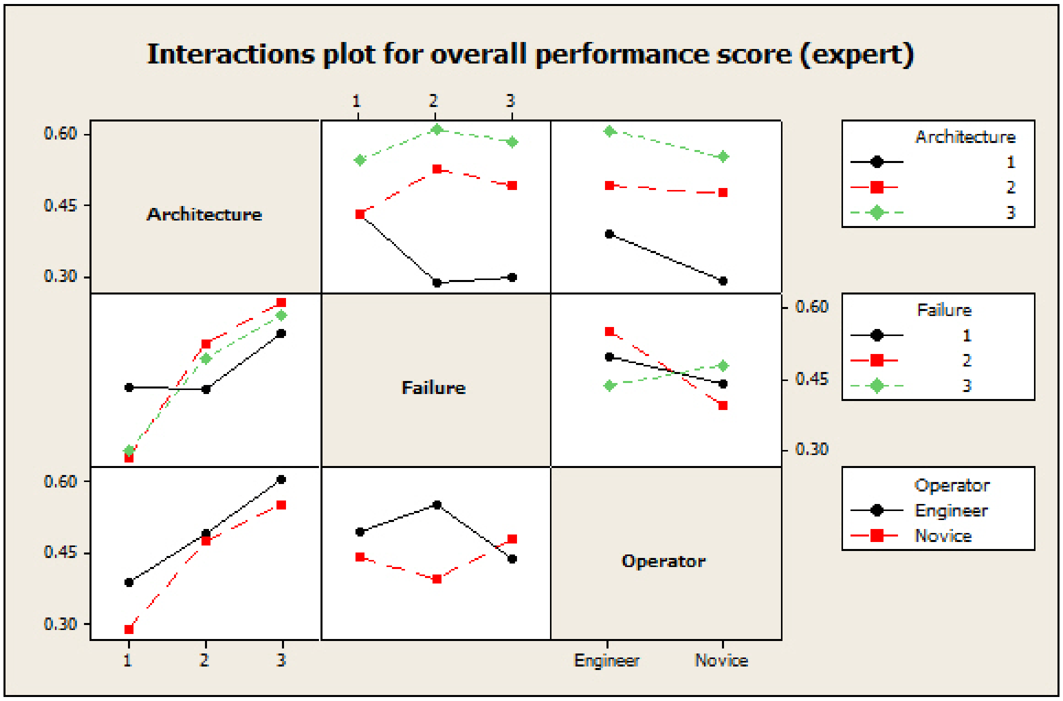


Figure 27—Interactions plot for overall performance scores- experts

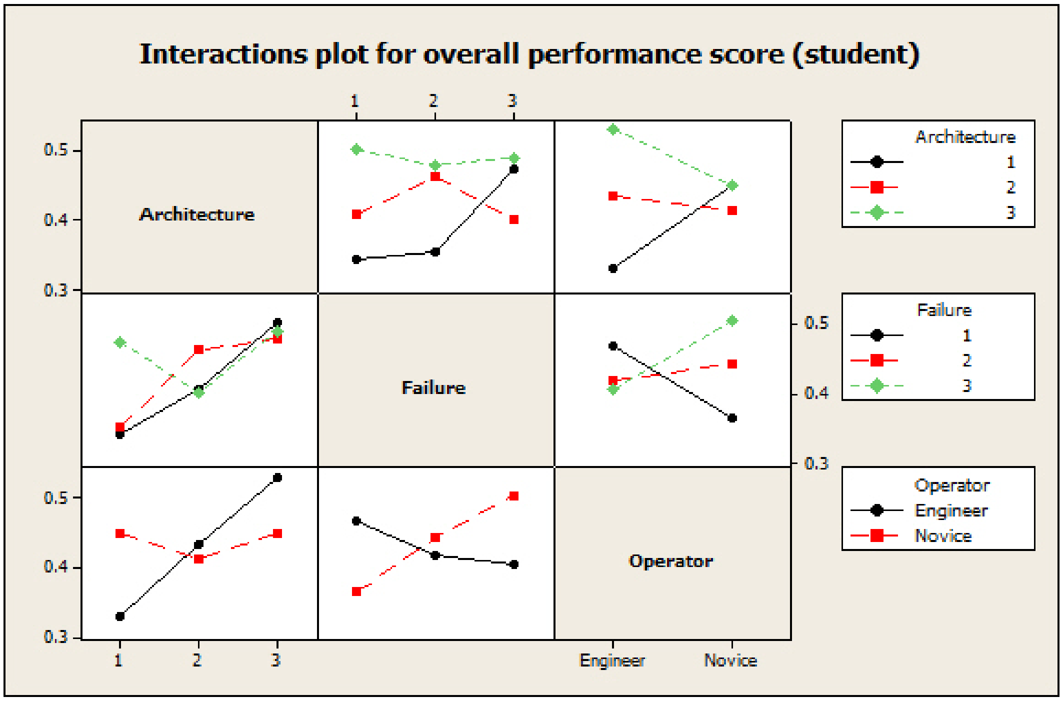


Figure 28—Interactions plot for overall performance scores- students

4.5.4 Comparison between architectures

The summary of the statistics of the different performance measures with the three architectures for failures-1, 2 and 3 is detailed in **Table 18** and for failure-4 in **Table 19**. Due the significant difference in performance between failure-4 and the other three failures as discussed in the previous section, to analyze the variation in the performance with the architectures, failures-1, 2 and 3 are considered separate from failure-4.

For the diagnosis of the failures-1, 2 and 3, from the view point of all the quantitative performance measures and the overall performance scores detailed in **Figure 19** to **Figure 28** it is clear that architecture-3 induced the best troubleshooting performance for the diagnosis of the first three failures especially in case of experts. The time taken to diagnose the failures was the maximum with architecture-1, reduced with architecture-2 and the least with architecture-3. The increase in capabilities of the architectures enabled the expert troubleshooters to diagnose the failures within lesser time. A similar trend was reflected in the overall performance score, where in architecture-3 induced the maximum performance score and architecture-1 induced the least.

Table 18—Summary statistics for the performance measures over the three architectures for failures-1, 2 and 3

Parameter	**Experts**			**Students**		
	A1 (n=6)	**A2 (n=6)**	**A3 (n=6)**	**A1 (n=6)**	**A2 (n=6)**	**A3 (n=6)**
Failures-1, 2 and 3						
T(mean)	1426.2	1000.7	867.3	1057.5	1069.2	1143.2
T (sd)	618.2	370.4	233.7	326.3	196.7	595.8
T (range)	1718	945	530	895	465	1712
PS (mean)	0.340	0.484	0.581	0.390	0.424	0.490
PS (sd)	0.096	0.115	0.084	0.121	0.047	0.085
PS (range)	0.273	0.262	0.243	0.318	0.131	0.265
PS-EW (mean)	0.316	0.434	0.524	0.358	0.371	0.414
PS-EW (sd)	0.114	0.115	0.098	0.119	0.053	0.085
PS-EW (range)	0.323	0.278	0.282	0.320	0.151	0.264
R (mean)	3.5	4.5	2.3	2.8	3.3	4.3
R (sd)	2.2	2.4	1.4	1.2	1.9	2.0
R (range)	5	8	3	0	3	0
Q (mean)	9.7	4.3	4.7	9.7	7.5	6.5
Q (sd)	5.9	3.2	3.2	6	3.9	3.2
Q (range)	11	8	8	17	10	7
IS (mean)	21.2	25.8	24.7	43	32	43
IS (sd)	10	13.8	16.7	27.8	16.3	32.3
IS (range)	30	35	39	76	38	94
DT (mean)	5.8	2.7	2.5	3.3	4	3.6
DT (sd)	3.3	1.5	1.4	1.6	1.1	1.7
DT (range)	9	4	3	4	3	5

NOTE—A1- architecture-1, A2- architecture-2, A3- architecture-3, n-number of samples, sd- standard deviation, T- time taken to diagnose the failure in seconds, PS- performance score calculated using the model, PS-EW- performance score calculated using the model for equal weights, R- number of runs (cycling the system), Q- number of questions asked by the troubleshooter, IS- number of information sources consulted by the troubleshooter, DT- number of diagnostic treatments performed.

The performance score with architecture-2 was better than that obtained with architecture-1 but lesser than that with architecture-3. It is also seen that with the increase in the capabilities of the architectures, there was reduced requirement to recreate the failure by cycling the system to diagnose the failure by the expert troubleshooters. Architecture-3 enabled the diagnosis of the failure with least requirement to cycle the system while architecture-1 maximized it. In case of the novice troubleshooters however, despite the increased capabilities on architecture-3, there was maximum tendency to cycle the system to diagnose the failure, completely opposite to that observed with experts.

Table 19—Summary statistics for the performance measures over the three architectures for failure-4

Parameter	**Experts**			**Students**		
	A1 (n=2)	**A2 (n=2)**	**A3 (n=2)**	**A1 (n=2)**	**A2 (n=2)**	**A3 (n=2)**
Failure-4						
T(mean)	1978	2117.5	2592	2015.5	1643.5	2372.5
T(sd)	364.8	951	356.4	580.5	292	342.9
PS (mean)	0.257	0.359	0.351	0.251	0.326	0.355
PS (sd)	0.032	0.110	0.013	0.05	0.024	0.019
PS-EW (mean)	0.223	0.331	0.276	0.213	0.262	0.279
PS-EW (sd)	0.038	0.151	0.014	0.034	0.014	0.014
R (mean)	10	15	22.5	17.5	20	17.5
R (sd)	7.1	14.1	3.5	3.5	0	3.5
Q (mean)	15	24	9.5	12.5	8	22.5
Q (sd)	1.4	12.7	0.7	10.6	0	0.7
IS (mean)	32.5	18.5	27.5	34	41	27
IS (sd)	16.3	21.9	7.8	4.2	1.4	1.4
DT (mean)	4.5	2.5	7	3.5	5	4
DT (sd)	0.7	2.1	1.4	0.7	1.4	2.8
NOTE—A1- architecture-1, A2- architecture-2, A3- architecture-3, n-number of samples, sd- standard deviation, T- time taken to diagnose the failure in seconds, PS- performance score calculated using the model, PS-EW- performance score calculated using the model for equal weights, R- number of runs (cycling the system), Q- number of questions asked by the troubleshooter, IS- number of information sources consulted by the troubleshooter, DT- number of diagnostic treatments performed.						

On similar lines, the increased capabilities of the architectures brought about a reduction in the operator interaction as seen from the decrease in the number of questions asked by the expert troubleshooters and the number of explicit diagnostic tests performed. Architectures-2 and 3 brought about the lesser operator interaction, while architecture-1 resulted in maximum operator interaction. Consistent with the behavior observed for the other performance metrics, the students displayed only a slight reduction in operator interaction with the increase in the capabilities of the architectures. The number of information screens viewed was not very different in case of the three architectures, however, the students displayed more transitions between the information screens than the experts.

In the diagnosis of failure-4, as seen from **Table 19**, the increase in the capabilities of the architectures brought about a reduction in troubleshooting performance for both expert and novice troubleshooters. The time taken to diagnose the failure and the number of times the system is cycled to diagnose the failure was maximum with architecture-3 and least with architecture-1 for the experts. The interaction with the operator was found to be the most important tool to diagnose the failure as seen from the number of questions asked and the number of explicit diagnostic tests performed. The overall performance score which is slightly better for architecture-3 could be attributed to the fact that it incorporates the quality of architecture, which is higher for architecture-3. In order to test the hypothesis for the difference in means of the performance metrics for the three architectures for failures-1, 2 and 3 and failure-4, the two sample t-test was performed with a confidence interval of 95% (α=0.05). The details of the hypotheses tested are tabulated in **Table 20** and **Table 21**.

From **Table 20**, it can be ascertained that the time taken by the experts to diagnose failures-1, 2 and 3 is significantly lesser with architecture-3 when compared to architecture-1. Although, the difference between architecture-2 and 3 and combination of architecture-1 and 2 were not found to be significant at 95% confidence interval, it is observed from **Table 18** that the average time taken to diagnose the failure with architecture-3 was lesser that that with architecture-2. This supports the fact that architecture-3 induced enhanced performance in terms of time taken to diagnose the failure. A similar relationship is observed when comparing the overall performance score for experts except that on the basis of the score, the performance between architecture-1 and 2 is also found to be significantly different. The difference between the overall performance scores of architecture-2 and 3 is significant at 90% confidence interval. Apart from inducing better performance over the different quantitative metrics, architectures-2 and 3 were also considered to be better than architecture-1 in terms of the qualitative performance metrics shown in figures on pages 113 and 114 in Section **4.5.7.**

On architectures-2 and 3 there is reduced interaction with the operator by the experts to diagnose the failures when compared to architecture-1 as seen from the hypotheses tested for the number of questions asked and explicit diagnostic treatments (tests) performed. This implies that with the increased capabilities of architectures-2 and 3, the expert troubleshooters could diagnose the failures with lesser dependence on the operator, consistent with the basis of the levels of remote diagnosis proposed in the standards [2]. The operator interaction and the video feedback were the only major information sources available to the remote troubleshooter when using architecture-1. The performance with architecture-1 was hence affected by the operator to a greater extent as shown in the interaction plot for time in **Figure 20** and confirmed by the interaction plot of scores from the model in **Figure 27**.

From the hypotheses tested for the number of times the system is cycled, it is clear that the tools available on architecture-2 required the system to be cycled in order to provide necessary information to the troubleshooter. The nature of the tools available on architecture-2 required real time monitoring when the system was in operation. The tools such as the stage diagram, video playback and the time based record of events available on architecture-3 enabled the diagnosis of the failure with reduced requirement to cycle the system. The average number of times the system was cycled with architecture-3 is seen to be lesser than that with architecture-1 as seen from **Figure 22**. However, this difference is not seen to be statistically significant.

From the hypotheses tested for the number of diagnostic treatments performed, it is seen that with architectures-2 and 3, significantly lesser number of explicit diagnostic treatments are performed when compared with architecture-1 which could be attributed to the additional information provided by the tools on these architectures. Most diagnostic treatments were performed with architecture-1 because of the lack of availability of additional information sources to verify or reduce the hypothesized causes of failure held by the troubleshooter.

Table 20—Statistical testing for difference in mean performance with the architectures for failures-1, 2 and 3

Null hypothesis ($\boldsymbol{H_0}$)	**Alternate hypothesis ($\boldsymbol{H_1}$)**	**Test Statistic (experts)**	**Outcome (experts)**	**Test Statistic (students)**	**Outcome (students)**
Time taken to diagnose the failure (seconds)					
$\boldsymbol{\bar{x}_3 - \bar{x}_1 = 0}$	$\boldsymbol{\bar{x}_3 - \bar{x}_1} < 0$	t = -2.17 (p=0.042) df=6	**Reject**	t= 0.31 (p=0.617) df=7	Fail to reject
$\bar{x}_2 - \bar{x}_1 = 0$	$\bar{x}_2 - \bar{x}_1 < 0$	t= -1.45 (p=0.093) df=8	**Reject***	t=0.08 (p=0.529) df=8	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 < 0$	t= -0.75 (p=0.239) df=8	Fail to reject	t= 0.29 (p=0.609) df=6	Fail to reject
Overall performance score					
$\boldsymbol{\bar{x}_3 - \bar{x}_1 = 0}$	$\boldsymbol{\bar{x}_3 - \bar{x}_1} > 0$	t = 4.64 (p=0.001) df=9	**Reject**	t = 1.67 (p=0.067) df=8	**Reject***
$\boldsymbol{\bar{x}_2 - \bar{x}_1 = 0}$	$\boldsymbol{\bar{x}_2 - \bar{x}_1} > 0$	t= 2.36 (p=0.021) df=9	**Reject**	t=0.64 (p=0.273) df=6	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 > 0$	t= 1.68 (p=0.064) df=9	**Reject***	t=1.69 (p=0.067) df=7	**Reject***
Number of times the system is cycled					
$\bar{x}_3 - \bar{x}_1 = 0$	$\bar{x}_3 - \bar{x}_1 < 0$	t= -1.12 (p=0.149) df=8	Fail to reject	t=1.61 (p=0.927) df=8	Fail to reject
$\bar{x}_2 - \bar{x}_1 = 0$	$\bar{x}_2 - \bar{x}_1 < 0$	t= 0.75 (p=0.764) df=9	Fail to reject	t=0.56 (p=0.704) df=8	Fail to reject
$\boldsymbol{\bar{x}_3 - \bar{x}_2 = 0}$	$\boldsymbol{\bar{x}_3 - \bar{x}_2} < 0$	t= -1.9 (p=0.049) df=7	**Reject**	t=0.9 (p=0.805) df=9	Fail to reject
Number of information sources consulted					
$\bar{x}_3 - \bar{x}_1 = 0$	$\bar{x}_3 - \bar{x}_1 < 0$	t= 0.44 (p=0.664) df=8	Fail to reject	t=0 (p=0.500) df=9	Fail to reject
$\bar{x}_2 - \bar{x}_1 = 0$	$\bar{x}_2 - \bar{x}_1 < 0$	t= 0.67 (p=0.740) df=9	Fail to reject	t= -0.84(p=0.213) df=8	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 < 0$	t= -0.13 (p=0.449) df=9	Fail to reject	t=0.74 (p=0.760) df=7	Fail to reject
Number of questions asked by the troubleshooter					
$\boldsymbol{\bar{x}_3 - \bar{x}_1 = 0}$	$\boldsymbol{\bar{x}_3 - \bar{x}_1} < 0$	t= -1.8 (p=0.057) df=7	**Reject***	t= -1.13(p=0.147) df=7	Fail to reject
$\boldsymbol{\bar{x}_2 - \bar{x}_1 = 0}$	$\boldsymbol{\bar{x}_2 - \bar{x}_1} < 0$	t= -1.93 (p=0.047) df=7	**Reject**	t= -0.74(p=0.241) df=8	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 < 0$	t= 0.18 (p=0.569) df=9	Fail to reject	t= -0.49(p=0.319) df=9	Fail to reject
Number of diagnostic tests performed					
$\boldsymbol{\bar{x}_3 - \bar{x}_1 = 0}$	$\boldsymbol{\bar{x}_3 - \bar{x}_1} < 0$	t= -2.31 (p=0.030) df=6	**Reject**	t=0.34 (p=0.630) df=9	Fail to reject
$\boldsymbol{\bar{x}_2 - \bar{x}_1 = 0}$	$\boldsymbol{\bar{x}_2 - \bar{x}_1} < 0$	t= -2.17 (p=0.034) df=7	**Reject**	t=0.83 (p=0.785) df=8	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 < 0$	t= -0.2 (p=0.423) df=9	Fail to reject	t= -0.4 (p=0.351) df=8	Fail to reject

NOTE—1-architecture-1, 2-architecture-2, 3-architecture-3, df-degrees of freedom, $\bar{x}$- sample mean, *-significant at 90% confidence interval.

Table 21—Statistical testing for difference in mean performance with the architectures for failure-4

Null hypothesis (H_0)	**Alternate hypothesis (H_1)**	**Test Statistic (experts)**	**Outcome (experts)**	**Test Statistic (students)**	**Outcome (students)**
Time taken to diagnose the failure (seconds)					
$\bar{x}_3 - \bar{x}_1 = 0$	$\bar{x}_3 - \bar{x}_1 < 0$	t = 1.7 (p=0.831) df=1	Fail to reject	t= 0.75 (p=0.705) df=1	Fail to reject
$\bar{x}_2 - \bar{x}_1 = 0$	$\bar{x}_2 - \bar{x}_1 < 0$	t = 0.9 (p=0.561) df=1	Fail to reject	t= -0.81 (p=0.283) df=1	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 < 0$	t= -0.66 (p=0.686) df=1	Fail to reject	t= 2.29 (p=0.869) df=1	Fail to reject
Overall performance score					
$\bar{x}_3 - \bar{x}_1 = 0$	$\bar{x}_3 - \bar{x}_1 > 0$	t = 2.86 (p=0.081) df=1	Fail to reject	t = 3.92 (p=0.079) df=1	Fail to reject
$\bar{x}_2 - \bar{x}_1 = 0$	$\bar{x}_2 - \bar{x}_1 > 0$	t= 1.26 (p=0.213) df=1	Fail to reject	t=2.72 (p=0.112) df=1	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 > 0$	t= -0.1 (p=0.532) df=1	Fail to reject	t=1.93 (p=0.152) df=1	Fail to reject
Number of times the system is cycled					
$\bar{x}_3 - \bar{x}_1 = 0$	$\bar{x}_3 - \bar{x}_1 < 0$	t= 2.24 (p=0.866) df=1	Fail to reject	t=0 (p=0.5) df=2	Support
$\bar{x}_2 - \bar{x}_1 = 0$	$\bar{x}_2 - \bar{x}_1 < 0$	t= 0.45 (p=0.634) df=1	Fail to reject	t=1.18 (p=0.776) df=1	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 < 0$	t= 0.73 (p=0.7) df=1	Fail to reject	t= -1.18 (p=0.224) df=1	Fail to reject
Number of information sources consulted					
$\bar{x}_3 - \bar{x}_1 = 0$	$\bar{x}_3 - \bar{x}_1 < 0$	t= -0.39 (p=0.381) df=1	Fail to reject	t=-2.21 (p=0.135) df=1	Fail to reject
$\bar{x}_2 - \bar{x}_1 = 0$	$\bar{x}_2 - \bar{x}_1 < 0$	t= -0.73 (p=0.3) df=1	Fail to reject	t= 2.21(p=0.865) df=1	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 < 0$	t= 0.55 (p=0.659) df=1	Fail to reject	t= -9.9 (p=0.005) df=2	**Reject**
Number of questions asked by the troubleshooter					
$\bar{x}_3 - \bar{x}_1 = 0$	$\bar{x}_3 - \bar{x}_1 < 0$	t= -4.92(p=0.064) df=1	Fail to reject	t= 1.33(p=0.795) df=1	Fail to reject
$\bar{x}_2 - \bar{x}_1 = 0$	$\bar{x}_2 - \bar{x}_1 < 0$	t= 0.99 (p=0.749) df=1	Fail to reject	t= -0.53(p=0.344) df=1	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 > 0$	t= -1.61 (p=0.0.823) df=1	Fail to reject	t= 19.8(p=0.001) df=2	**Reject**
Number of diagnostic tests performed					
$\bar{x}_3 - \bar{x}_1 = 0$	$\bar{x}_3 - \bar{x}_1 < 0$	t= 2.24 (p=0.866) df=1	Fail to reject	t=0.24 (p=0.576) df=1	Fail to reject
$\bar{x}_2 - \bar{x}_1 = 0$	$\bar{x}_2 - \bar{x}_1 < 0$	t= -1.26 (p=0.213) df=1	Fail to reject	t=1.34 (p=0.796) df=1	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 < 0$	t= 2.5 (p=0.879) df=1	Fail to reject	t= -0.45 (p=0.366) df=1	Fail to reject

NOTE—1-architecture-1, 2-architecture-2, 3-architecture-3, df-degrees of freedom, $\bar{x}$- sample mean.

From the hypotheses tested for the number of information sources consulted, no significant difference is observed between the three architectures. With the increase in the levels of the architectures, there were more information sources available to the remote expert. Despite this increase, the expert troubleshooters made nearly the same number of transitions between the information sources. The methodology of utilization of the information screens by the experts (discussed in the following section) resulted in a consistent distribution of information sources over the three architectures.

While the hypotheses tested reveal differences in performance between the architectures for the diagnosis of failures-1, 2 and 3 with the experts, there is no significant difference observed in performance between the architectures with the novice troubleshooters (students), except the overall performance score at 90% confidence interval. This indicates that the additional capabilities of the architectures were not found to be as useful to improve troubleshooting performance by the students. They tended to generate similar levels of performance with all the three architectures. The increase in capabilities also resulted in a slight reduction in performance in terms of time taken to diagnose the failure and number of times the system is cycled, which points towards the difficulty faced by the novice troubleshooters in using the tools provided on architectures-2 and 3. Ideally, with the capabilities available on architecture-3, a decrease in the requirement to cycle the system is expected. The comparison between the troubleshooting strategies of the expert and novice troubleshooters is addressed in the following S**ection 4.5.5** and their ineffectiveness in using the tools on the architectures is also discussed.

With regards to the diagnosis of failure-4 by the experts, there is no significant difference in performance between the architectures as seen from the results of the hypotheses tested in **Table 21**. As established in the previous **Section 4.5.3**, there is a significant reduction in the overall troubleshooting performance in the diagnosis of failure-4 when compared with the other three failures. Consistent with this conclusion, failure-4 was found to be difficult to diagnose with all the three architectures. The increased capabilities brought about a reduction in troubleshooting performance which illustrates that in cases where the failure is not related to measured or monitored variable of the system, additional diagnostic capabilities that automate the information gathering process tend to introduce additional failure causes to be hypothesized which could increase the overall time taken to troubleshoot the failure and the number of diagnostic tests performed.

In the case of novice troubleshooters, a similar trend is observed except the fact that the students asked significantly more number of questions despite consulting lesser information sources with architecture-3 over architecture-2. This once again suggests that the information provided by architecture-3 was not really helpful in the diagnosis of failure-4 and operator interaction was more important.

4.5.5 Comparison between expert and novice troubleshooters

Based on the discussion comparing the difference between the architectures, it was observed that the novice troubleshooters did not find the increased capabilities of the architectures to be as useful as the expert troubleshooters to diagnose the failures. Unlike the experts, the students generated similar levels of troubleshooting performance with all the three architectures. The results of the hypotheses tested for the comparison of the various performance metrics between the expert and novice troubleshooters (students) for failures-1, 2, 3 and failure-4 using the two sample t-test (α=0.05) are presented in **Table 22** and **Table 23** respectively.

From the hypotheses tested for the difference in performance between the expert and novice troubleshooters with architecture-1 in **Table 22**, the performance of the experts and students is not found to be significantly different on the basis of time taken to diagnose the failure, overall performance score, number of times the system is cycled, number of questions asked and the number of diagnostic treatments performed. The novice troubleshooters generated better performance than the experts as seen from the reduced mean time taken to diagnose the failure and the increased mean overall performance score as seen in **Table 18**. However, based on the hypothesis tested for the number of information sources consulted, there is evidence to suggest that the experts consulted lesser information sources than the students during the diagnosis process. The number of transitions made between the information screens was higher for the novice troubleshooters when compared with that of the expert troubleshooters given the limited information sources available on architecture-1.

In case of architecture-2, despite the improved performance of the expert troubleshooters over the novice troubleshooters illustrated from the mean values of the various performance metrics in **Table 18**, only the number of explicit diagnostic tests performed was found to be significantly lesser for the experts in comparison to that of the novice troubleshooters. Despite the additional capabilities provided, the novice troubleshooters depended more on the local operator to perform diagnostic treatments to justify their hypothesized failure causes. The reduction in operator interaction with the additional capabilities is noticed mainly for the experts and not the novice troubleshooters.

The ineffectiveness of the novice troubleshooters to utilize the increased capabilities is confirmed from the hypotheses tested over architecture-3 for the number of times the system is cycled. The availability of time based record of events and other tools on architecture-3 enabled the experts to cycle the system significantly fewer times to diagnose the failure in comparison with the novice troubleshooters. The time taken to diagnose the failure, the interaction with the operator (questions and diagnostic tests) and the number of information screens viewed are lesser for experts than novices as seen from **Table 18**. However, these differences were not found to be statistically significant at 95% confidence interval. But the fact that the overall performance score for the experts was found to be significantly higher than that for the students emphasizes that the expert troubleshooters were more efficient in using the additional information to improve troubleshooting performance.

Table 22—Statistical testing for difference in mean performance between expert and novice troubleshooters for failures-1, 2 and 3

Null hypothesis (H_0)	**Alternate hypothesis (H_1)**	**Par**	**Test Statistic**	**Outcome**
Architecture-1				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	T	t= 1.29 (p=0.881) df=7	Fail to reject
$\sigma_{expert} - \sigma_{student} = 0$	$\sigma_{expert} - \sigma_{student} < 0$	T	f= 3.59 (p=0.187)	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	PS	t=-0.8 (p=0.779) df=9	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	R	t= 0.66 (p=0.736) df=7	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	**IS**	t= -1.81 (p=0.06) df=6	**Reject***
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	Q	t= 0 (p=0.5) df=9	Support
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	DT	t= 1.68 (p=0.932) df=7	Fail to reject
Architecture-2				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	T	t= -0.4 (p=0.351) df=7	Fail to reject
$\sigma_{expert} - \sigma_{student} = 0$	$\sigma_{expert} - \sigma_{student} < 0$	T	f= 3.55 (p=0.191)	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	PS	t=-1.18 (p=0.142) df=6	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	R	t= 0.93 (p=0.813) df=9	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	IS	t= -0.71 (p=0.248) df=9	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	**Q**	t= -1.54 (p=0.079) df=9	**Reject***
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	**DT**	t= -1.75 (p=0.057) df=9	**Reject***
Architecture-3				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	T	t= -1.06 (p=0.166) df=6	Fail to reject
$\sigma_{expert} - \sigma_{student} = 0$	$\sigma_{expert} - \sigma_{student} < 0$	**T**	f= 0.15 (p=0.061)	**Reject***
$\boldsymbol{\bar{x}_{expert} - \bar{x}_{student} = 0}$	$\boldsymbol{\bar{x}_{expert} - \bar{x}_{student}} > 0$	**PS**	t=1.86 (p=0.048) df=9	**Reject**
$\boldsymbol{\bar{x}_{expert} - \bar{x}_{student} = 0}$	$\boldsymbol{\bar{x}_{expert} - \bar{x}_{student}} < 0$	**R**	t= -2.05 (p=0.037) df=8	**Reject**
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	IS	t= -1.23 (p=0.128) df=7	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	Q	t= -0.98 (p=0.176) df=9	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	DT	t= -1.28 (p=0.116) df=9	Fail to reject

NOTE— Par- parameter, expert-expert troubleshooter, student-novice troubleshooter, df-degrees of freedom. T- time taken to diagnose the failure in seconds, PS- performance score calculated using the model, R- number of runs (cycling the system), Q- number of questions asked by the troubleshooter, IS- number of information sources consulted by the troubleshooter, DT- number of diagnostic treatments performed, $\bar{x}$- sample mean, σ-sample variance, *- significant at 90% confidence interval.

Table 23—Statistical testing for difference in mean performance between expert and novice troubleshooters for failure-4

Null hypothesis (H_0)	**Alternate hypothesis (H_1)**	**Par.**	**Test Statistic**	**Outcome**
Architecture-1				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	T	t= -0.08 (p=0.475) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	PS	t= 0.18 (p=0.443) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	R	t= -1.34 (p=0.204) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	IS	t= -0.13 (p=0.460) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	Q	t= 0.33 (p=0.602) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	DT	t= 1.41 (p=0.854) df=2	Fail to reject
Architecture-2				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	T	t= -0.67 (p=0.689) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	PS	t= 0.43 (p=0.372) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	R	t= -0.55 (p=0.34) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	IS	t= -1.45 (p=0.192) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	Q	t= 1.72 (p=0.832) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	DT	t= -1.39 (p=0.199) df=1	Fail to reject
Architecture-3				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	T	t= 0.63 (p=0.678) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	PS	t= -0.29 (p=0.59) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	R	t= 1.41 (p=0.854) df=2	Reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	IS	t= 0.09 (p=0.528) df=1	Fail to reject
$\boldsymbol{\bar{x}_{expert} - \bar{x}_{student} = 0}$	$\boldsymbol{\bar{x}_{expert} - \bar{x}_{student}} < 0$	**Q**	t= -18.38 (p=0.001) df=2	**Reject**
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	DT	t= 1.34 (p=0.796) df=1	Fail to reject

NOTE— Par- parameter, expert-expert troubleshooter, student-novice troubleshooter, df-degrees of freedom. T- time taken to diagnose the failure in seconds, PS- performance score calculated using the model, R- number of runs (cycling the system), Q- number of questions asked by the troubleshooter, IS- number of information sources consulted by the troubleshooter, DT- number of diagnostic treatments performed, $\bar{x}$- sample mean.

In case of failure-4 however, the difference between experts and the novice troubleshooters is not found to be statistically significant which could be largely attributed to the nature of the failure. However, the number of questions asked with architecture-3 is higher for the students in comparison with the experts illustrating the continued reliance on the information from the operator by the novice troubleshooters.

In order to further investigate the difference in troubleshooting strategies of the expert and novice troubleshooters, the timestamps of all the activities performed during the diagnosis process by the experts and novices were collected from the video recorded of the screen viewed by the troubleshooter and the audio recorded of the conversation between the troubleshooters and the operators. Based on the descriptive model of troubleshooting proposed in [21], the time spent in the troubleshooting process can be classified into two main categories: information collection and hypothesis testing. Information collection refers to the assessment of the situation in order to gain knowledge while hypothesis testing involves the use of the knowledge gained to support or reject hypothesized causes of failure.

In the experiments performed in this study, information collecting activities mainly include viewing the system via the webcam (W), observing the information contained in the I/O list (IL), offline (Off-L) and online ladder diagrams (On-L), labview® voltage waveforms (LV), the stage diagram (S) and detailed view of each station (SD), time based record of events (TR) and the playback of the failed operation (VP). The hypothesis testing activities include the questions asked by the troubleshooter of the operator, the explicit diagnostic tests performed and the cycling of the system. It must be noted that even while performing hypothesis testing, the troubleshooters are engaged in information collection as they view the screen of the computer in front of them. As a result, it is not possible to ascertain if at any given time hypothesis testing was performed without information collection. The average time spent by the two types of troubleshooters in information collection activities, hypothesis testing activities and the time taken to test the first hypothesized cause of failure is tabulated in **Table 24**.

As seen from **Table 24**, in the diagnosis of failures-1, 2 and 3, the time spent in information collection activities alone by the experts displays a reducing trend from architecture-1 to architecture-3 with architecture-2 being in the middle. This indicates that the experts were able to gather information necessary for the diagnosis of failures-1, 2 and 3 within lesser time with the increase in the capabilities of the architectures. On the other hand, a completely opposite trend was observed with novice troubleshooters as they required more time to gather the information necessary to formulate or validate hypothesized causes of failure with the increase in the capabilities of the architectures. This is an indication of the difficulty faced by the novice troubleshooters in handling the tools and gathering the information necessary for failure diagnosis with architecture-2 and architecture-3 to a greater extent.

Table 24—Summary statistics for time spent in diagnosis activities by expert and novice troubleshooters

Parameter	**Experts**			**Students**		
	A1	**A2**	**A3**	**A1**	**A2**	**A3**
Failures-1, 2 and 3						
IC (mean)	955	755	650	698	747	815
IC (sd)	502	345	305	313	188	594
HT (mean)	471	246	217	397	322	328
HT (sd)	238	151	102	123	190	197
FT (mean)	156	297	322	245	325	422
FT (sd)	158	151	187	217	232	458
Failure-4						
IC (mean)	1511	1341	1961	1326	834.7	1341
IC (sd)	222	619	462	423	44.7	122
HT (mean)	467	777	631	689	809	1032
HT (sd)	143	332	106	157	337	465
FT (mean)	210	103	230	532	296	414.5
FT (sd)	148	69.3	148	5.7	129	7.8
NOTE— All the time spent is measured in seconds. IC- time spent in information collection alone, HT- time spent in hypothesis testing while viewing information screen, FT- time spent before performing the first test of the hypothesized case of failure.						

In the diagnosis of failures-1, 2 and 3 it is seen that with the increase in capabilities of the architectures, the experts spent lesser time than the novice troubleshooters in hypothesis testing activities. Most of the hypothesis testing activities involved interaction with the operator and the dependence on the operator to cycle the system, perform diagnostic tests and replies to questions asked. It is thus possible to conclude that there was reduced dependence on the information provided by the operator with the increase in levels of architectures in case of expert troubleshooters. However, the novice troubleshooters tended to place similar amount of importance on the interaction with the operator via the hypothesis testing activities despite the increase in the capabilities of the architectures. The experts on an average, spent lesser time before testing their first hypothesis of the cause of failure. This could be attributed to the higher knowledge levels of the experts and their experience with troubleshooting automated systems coupled with their ability to extract information better than the novices from the information collection activities. It can also be seen that the time taken to perform the first test that it increases with the capabilities of the architectures for both expert and novice troubleshooters. This indicates that with the increase in the capabilities of the architectures, the troubleshooters spent more time formulating or eliminating their hypothesized causes of failure before testing.

In case of failure-4 however, there is increased time spent on both information collection and hypothesis testing activities with all the three architectures owing to the level of difficulty involved in the diagnosis of the failure. The marked difference between the experts and novices in the diagnosis of failures-1, 2 and 3 was not observed with this failure indicating that failure-4 was difficult to diagnose for experts and novices.

The results of the hypotheses tested for the comparison of the time spent in diagnosis activities between the architectures by expert and novice troubleshooters (students) for failures-1, 2, 3 and failure-4 using the two sample t-test ($\alpha=0.05$) are presented in **Table 25**. It is observed that for failures-1, 2 and 3, the time spent in hypothesis testing activities by the experts with architectures-2 and 3 is significantly lesser than that with architecture-1. However, this is not true in case of novice troubleshooters. This supplements the fact that, with the increase in the levels of architectures, the experts were able to reduce the dependence on the operator for information required to diagnose he failure. The novices on the other hand could not utilize the additional information provided as effectively and relied to a similar extent on the information given by the operator. The experts spent significantly (at 90% confidence interval) more time before performing their first test of the hypothesized cause of failure with architectures-2 and 3 when compared to architecture-1 indicating that they devoted time to integrate the information provided before testing. The novices, spent more time than the experts with architectures-2 and 3 than architecture-1 before performing their first test, but this difference is not found to be significant.

The results of the hypotheses tested for the comparison of the time spent in diagnosis activities between expert and novice troubleshooters (students) for failures-1, 2, 3 and failure-4 using the two sample t-test (α=0.05) are presented in **Table 26**. The difference observed between the experts and novices in terms of the time spent in diagnosis activities in **Table 24** was not found to be statistically significant.

Table 25—Statistical testing for difference in mean time spent in diagnosis activities between the three architectures for expert and novice troubleshooters

Null hypothesis (H_0)	**Alternate hypothesis (H_1)**	**Test Statistic (experts)**	**Outcome (experts)**	**Test Statistic (students)**	**Outcome (students)**
Failures-1, 2 and 3					
Time spent in information collection activities (seconds)					
$\bar{x}_3 - \bar{x}_1 = 0$	$\bar{x}_3 - \bar{x}_1 < 0$	t= -1.27 (p=0.120) df=8	Fail to reject	t= 0.43 (p=0.659) df=7	Fail to reject
$\bar{x}_2 - \bar{x}_1 = 0$	$\bar{x}_2 - \bar{x}_1 < 0$	t= -0.8 (p=0.222) df=8	Fail to reject	t= 0.33 (p=0.626) df=8	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 < 0$	t= -0.56 (p=0.296) df=9	Fail to reject	t= 0.26 (p=0.599) df=5	Fail to reject
Time spent in hypothesis testing activities (seconds)					
$\bar{\boldsymbol{x}}_{\boldsymbol{3}} - \bar{\boldsymbol{x}}_{\boldsymbol{1}} = \boldsymbol{0}$	$\bar{\boldsymbol{x}}_{\boldsymbol{3}} - \bar{\boldsymbol{x}}_{\boldsymbol{1}} < 0$	t= -2.41 (p=0.026) df=8	**Reject**	t= -0.72 (p=0.246) df=8	Fail to reject
$\bar{\boldsymbol{x}}_{\boldsymbol{2}} - \bar{\boldsymbol{x}}_{\boldsymbol{1}} = \boldsymbol{0}$	$\bar{\boldsymbol{x}}_{\boldsymbol{2}} - \bar{\boldsymbol{x}}_{\boldsymbol{1}} < 0$	t= -1.96 (p=0.043) df=8	**Reject**	t= -0.81 (p=0.221) df=8	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 < 0$	t= -0.39 (p=0.355) df=8	Fail to reject	t= 0.06 (p=0.523) df=9	Fail to reject
Time spent before performing the first test (seconds)					
$\bar{x}_3 - \bar{x}_1 = 0$	$\bar{x}_3 - \bar{x}_1 > 0$	t= 1.66 (p=0.066) df=9	**Reject***	t= 0.85 (p=0.211) df=7	Fail to reject
$\bar{x}_2 - \bar{x}_1 = 0$	$\bar{x}_2 - \bar{x}_1 > 0$	t= 1.58 (p=0.074) df=9	**Reject***	t= 0.61 (p=0.278) df=9	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 > 0$	t= 0.25 (p=0.404) df=9	Fail to reject	t= 0.46 (p=0.329) df=7	Fail to reject
Failure-4					
Time spent in information collection activities (seconds)					
$\bar{x}_3 - \bar{x}_1 = 0$	$\bar{x}_3 - \bar{x}_1 < 0$	t= 1.24 (p=0.784) df=1	Fail to reject	t= 0.05 (p=0.515) df=1	Fail to reject
$\bar{x}_2 - \bar{x}_1 = 0$	$\bar{x}_2 - \bar{x}_1 < 0$	t= -0.37 (p=0.388) df=1	Fail to reject	t= -1.63 (p=0.175) df=1	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 < 0$	t= 1.14 (p=0.770) df=1	Fail to reject	t= 5.49 (p=0.943) df=1	Fail to reject
Time spent in hypothesis testing activities (seconds)					
$\bar{x}_3 - \bar{x}_1 = 0$	$\bar{x}_3 - \bar{x}_1 < 0$	t= 1.30 (p=0.791) df=1	Fail to reject	t= 0.99 (p=0.748) df=1	Fail to reject
$\bar{x}_2 - \bar{x}_1 = 0$	$\bar{x}_2 - \bar{x}_1 < 0$	t= 1.21 (p=0.78) df=1	Fail to reject	t= 0.46 (p=0.636) df=1	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 < 0$	t= -0.59 (p=0.330) df=1	Fail to reject	t= 0.55 (p=0.66) df=1	Fail to reject
Time spent before performing the first test (seconds)					
$\bar{x}_3 - \bar{x}_1 = 0$	$\bar{x}_3 - \bar{x}_1 > 0$	t= -0.92 (p=0.263) df=1	Fail to reject	t= -17.28 (p=0.018) df=1	**Reject**
$\bar{x}_2 - \bar{x}_1 = 0$	$\bar{x}_2 - \bar{x}_1 > 0$	t= 0.13 (p=0.542) df=1	Fail to reject	t= -2.59 (p=0.117) df=1	Fail to reject
$\bar{x}_3 - \bar{x}_2 = 0$	$\bar{x}_3 - \bar{x}_2 > 0$	t= 1.1 (p=0.765) df=1	Fail to reject	t= 1.3 (p=0.791) df=1	Fail to reject

Table 26—Statistical testing for difference in mean time spent in diagnosis activities between expert and novice troubleshooters

Null hypothesis (H_0)	**Alternate hypothesis (H_1)**	**Par**	**Test Statistic**	**Outcome**
Failures-1, 2 and 3				
Architecture-1				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	IC	t= 1.06 (p=0.159) df=8	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	HT	t= 0.68 (p=0.742) df=7	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	FT	t= -0.82 (p=0.218) df=9	Fail to reject
Architecture-2				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	IC	t= 0.05 (p=0.482) df=7	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	HT	t= -0.77 (p=0.231) df=9	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	FT	t= -0.24 (p=0.406) df=8	Fail to reject
Architecture-3				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	IC	t= -0.6 (p=0.717) df=7	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	HT	t= -1.23 (p=0.129) df=7	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	FT	t= -0.49 (p=0.319) df=6	Fail to reject
Failure-4				
Architecture-1				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	IC	t= 0.55 (p=0.341) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	HT	t= -1.48 (p=0.19) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	FT	t= -3.06 (p=0.1) df=1	Fail to reject
Architecture-2				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	IC	t= 1.15 (p=0.228) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	HT	t= -0.09 (p=0.470) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	FT	t= -1.87 (p=0.156) df=1	Fail to reject
Architecture-3				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	IC	t= 1.83 (p=0.159) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	HT	t= -1.19 (p=0.223) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	FT	t= -1.77 (p=0.164) df=1	Fail to reject
NOTE— Par- parameter, expert-expert troubleshooter, student-novice troubleshooter, df-degrees of freedom, $\bar{x}$- sample mean.				

The total time spent in each of the information collection activities by the expert and novice troubleshooters for the diagnosis of failures-1, 2, 3 and failure-4 is tabulated in **Table 27** and **Table 28** respectively.

Table 27—Summary statistics for time spent in each information collection activity by expert and novice troubleshooters for the failures-1, 2 and 3

Parameter	**Experts**			**Students**		
	A1	**A2**	**A3**	**A1**	**A2**	**A3**
Failures-1, 2 and 3						
W(mean)	624	275	154	484	333	220
W (sd)	583	211	129	213	227	145
IL (mean)	323	110.6	151	239	126.7	153
IL (sd)	330	48.5	109	159	69.9	162
Off-L (mean)	465	NA	NA	349	NA	NA
Off-L (sd)	338	NA	NA	129	NA	NA
On-L (mean)	NA	388	153	NA	246	207
On-L (sd)	NA	202	111	NA	118	218
LV (mean)	NA	257	60	NA	320	113.5
LV (sd)	NA	138	-	NA	193	57.1
S (mean)	NA	NA	114	NA	NA	113.4
S (sd)	NA	NA	96.9	NA	NA	87.8
SD (mean)	NA	NA	79.8	NA	NA	135
SD (sd)	NA	NA	74.7	NA	NA	110
TR (mean)	NA	NA	277	NA	NA	248
TR (sd)	NA	NA	205	NA	NA	109

NOTE— All the time spent is in measured in seconds. W- time spent observing the system using the webcam, IL- time spent observing the I/O list, Off-L- time spent observing the offline ladder program, On-LV- time spent observing the online ladder program, L- time spent observing the labview® voltage waveforms, S- time spent observing the stage diagram, SD- time spent observing the detailed view of a station with stage diagram, TR- time spent observing the time based record of events, NA-not applicable.

The results of the hypotheses tested for the comparison of the time spent in each of the information collection activities between the expert and novice troubleshooters (students) for failures-1, 2, 3 and failure-4 using the two sample t-test ($\alpha=0.05$) are presented in **Table 29** and **Table 30**. It can be observed that with architecture-1 and architecture-2, the experts tended to spend more amount of time observing the ladder programs (online and offline) in the diagnosis of all the four failures as seen from **Table 27** and **Table 28**. This emphasizes that the experts spent more time developing an understanding of the interaction between inputs and the outputs in the system in the process of generating and testing hypothesized causes of failures. However, from the hypotheses tested for the time spent per information collection activity, it is observed that there is no significant difference in between the expert and novice troubleshooters for the diagnosis of failures-1, 2, 3 and failure-4 as seen in **Table 29** and **Table 30**.

Table 28—Summary statistics for time spent in each information collection activity by expert and novice troubleshooters for the failure-4

Parameter	**Experts**			**Students**		
	A1	**A2**	**A3**	**A1**	**A2**	**A3**
			Failure-4			
W(mean)	1208	688	1218.2	1293	887	1237
W (sd)	74	770	30.8	148	127	302
IL (mean)	152	187	92.6	236	161	75
IL (sd)	175	263	27.5	133	4.6	45.6
Off-L (mean)	559	NA	NA	486	NA	NA
Off-L (sd)	32	NA	NA	399	NA	NA
On-L (mean)	NA	942	624	NA	398	702
On-L (sd)	NA	508	-	NA	202	151
LV (mean)	NA	604	-	NA	197	-
LV (sd)	NA	-	-	NA	223	-
S (mean)	NA	NA	500	NA	NA	236.5
S (sd)	NA	NA	445	NA	NA	24.4
TR (mean)	NA	NA	-	NA	NA	-
TR (sd)	NA	NA	-	NA	NA	-
V (mean)	NA	NA	168	NA	NA	60
V (sd)	NA	NA	-	NA	NA	1.3

NOTE— All the time spent is in measured in seconds. W- time spent observing the system using the webcam, IL- time spent observing the I/O list, Off-L- time spent observing the offline ladder program, On-LV- time spent observing the online ladder program, L- time spent observing the labview® voltage waveforms, S- time spent observing the stage diagram, SD- time spent observing the detailed view of a station with stage diagram, TR- time spent observing the time based record of events, Q- time spent asking questions and receiving replies, T- time spent in performing diagnostic treatments, R- time spent in cycling the system, NA-not applicable.

Table 29—Statistical testing for difference in mean time spent in each activity between expert and novice troubleshooters for failures-1, 2 and 3 with architectures-1 and 2

Null hypothesis (H_0)	**Alternate hypothesis (H_1)**	**Par**	**Test Statistic**	**Outcome**
Failures-1, 2 and 3				
Architecture-1				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	W	t= 0.55 (p=0.301) df=6	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	IL	t= 0.57 (p=0.295) df=7	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	Off-L	t= 0.79 (p=0.231) df=6	Fail to reject
Architecture-2				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	W	t= -0.46 (p=0.673) df=9	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	IL	t= -0.46 (p=0.672) df=8	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	On-L	t= 1.49 (p=0.088) df=8	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	LV	t= -0.63 (p=0.728) df=8	Fail to reject
Architecture-3				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	W	t= -0.84 (p=0.789) df=9	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	IL	t= -0.02 (p=0.506) df=5	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	On-L	t= -0.52 (p=0.689) df=7	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	LV	t= -2.01 (p=0.909) df=2	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	S	t= 0.01 (p=0.496) df=9	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	SD	t= -1.02 (p=0.83) df=8	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	TR	t= 0.31 (p=0.383) df=8	Fail to reject

Table 30—Statistical testing for difference in mean time spent per activity between expert and novice troubleshooters for failure-4

Null hypothesis (H_0)	**Alternate hypothesis (H_1)**	**Par**	**Test Statistic**	**Outcome**
Failure-4				
Architecture-1				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	W	t= -0.73(p=0.701) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	IL	t= -0.54 (p=0.659) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	Off-L	t= 0.34 (p=0.395) df=1	Fail to reject
Architecture-2				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	W	t= -0.36 (p=0.61) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	IL	t= 0.13 (p=0.458) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	On-L	t= 1.41 (p=0.197) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	LV	t= 0.31 (p=0.405) df=1	Fail to reject
Architecture-3				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	W	t= -0.09 (p=0.527) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	IL	t= 0.47 (p=0.361) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	On-L	t= -1.18 (p=0.777) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	S	t= 0.83 (p=0.279) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	V	t= 0.28 (p=0.412) df=1	Fail to reject

The difference in the performance of the expert and novice troubleshooters with respect to the number of transitions to a particular information source is tabulated in **Table 31** and the results of the hypotheses tested for the difference in the number of transitions between the information sources by the experts and novices in the diagnosis of failures-1, 2 and 3 and failure-4 are tabulated in **Table 32** and **Table 33**.

Table 31—Summary statistics for number of times information source is consulted by expert and novice troubleshooters

Parameter	**Experts**			**Students**		
	A1	**A2**	**A3**	**A1**	**A2**	**A3**
Failures-1, 2 and 3						
W_n (mean)	5.2	3.8	3.2	13	9.8	9.7
W_n (sd)	2.9	2	1.5	7.2	5.9	8
IL_n (mean)	7.3	5	8.2	13.7	5.7	9.6
IL_n (sd)	4.8	3.1	6.5	6.7	2.7	10.6
Off-L_n (mean)	8.3	NA	NA	14.3	NA	NA
Off-L_n (sd)	5.5	NA	NA	10.3	NA	NA
On-L_n (mean)	NA	7.3	9.2	NA	7.8	11.3
On-L_n (sd)	NA	3.4	7.5	NA	4.8	13.7
LV_n (mean)	NA	4.4	4	NA	7.3	5
LV_n (sd)	NA	3.4	-	NA	4.5	4.7
S_n (mean)	NA	NA	2.3	NA	NA	2
S_n (sd)	NA	NA	1.2	NA	NA	0.7
SD_n (mean)	NA	NA	3.2	NA	NA	4.2
SD_n (sd)	NA	NA	1.9	NA	NA	3
TR_n (mean)	NA	NA	3.7	NA	NA	6.4
TR_n (sd)	NA	NA	3.4	NA	NA	6.2
Failure-4						
W_n (mean)	7	6	7	8.5	11.5	10
W_n (sd)	4.2	7.1	4.2	0.7	0.7	1.4
IL_n (mean)	12	4	3.5	10	11.5	1.5
IL_n (sd)	5.7	5.7	0.7	0.4	2.1	0.7
Off-L_n (mean)	12.5	NA	NA	13	NA	NA
Off-L_n (sd)	4.9	NA	NA	4.2	NA	NA
On-L_n (mean)	NA	3	5.5	NA	7	6.5
On-L_n (sd)	NA	1.4	7.8	NA	1.4	2.1
LV_n (mean)	NA	5.5	-	NA	7.5	-
LV_n (sd)	NA	7.8	-	NA	3.5	-
S_n (mean)	NA	NA	6	NA	NA	5
S_n (sd)	NA	NA	7.1	NA	NA	5.7
TR_n (mean)	NA	NA	7	NA	NA	2
TR_n (sd)	NA	NA	-	NA	NA	-

NOTE—W_n- number of times the webcam is viewed, IL_n- number of times the I/O list is viewed, Off-L_n- number of times the offline ladder program is viewed, On-L_n- number of times the online ladder program is viewed, LV_n- number of times the labview® voltage waveforms is viewed, S_n- number of times the stage diagram is viewed, SD_n- number of times the detailed view of a station with stage diagram is viewed, TR_n- number of times the time based record of events is viewed, NA-not applicable.

Table 32—Statistical testing for difference in mean number of times information source is consulted between expert and novice troubleshooters for failures-1, 2 and 3

Null hypothesis (H_0)	**Alternate hypothesis (H_1)**	**Par**	**Test Statistic**	**Outcome**
Failures-1, 2 and 3				
Architecture-1				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	W$_n$	t= -2.47 (p=0.024) df=6	**Reject**
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	IL$_n$	t= -1.88 (p=0.046) df=9	**Reject**
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	Off-L$_n$	t= -1.26 (p=0.124) df=7	Fail to reject
Architecture-2				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	W$_n$	t= -2.35 (p=0.029) df=6	**Reject**
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	IL$_n$	t= -0.4 (p=0.351) df=9	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	On-L$_n$	t= -0.21 (p=0.420) df=8	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	LV$_n$	t= -1.22 (p=0.128) df=8	Fail to reject
Architecture-3				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	W$_n$	t= -1.96 (p=0.054) df=5	**Reject***
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	IL$_n$	t= -0.25 (p=0.405) df=6	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	On-L$_n$	t= -0.31 (p=0.383) df=7	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	LV$_n$	t= -0.94 (p=0.208) df=3	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	S$_n$	t= 0.57 (p=0.707) df=8	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	SD$_n$	t= -0.57 (p=0.298) df=5	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	TR$_n$	t= -0.88 (p=0.206) df=6	Fail to reject

NOTE— Par- parameter, expert-expert troubleshooter, student-novice troubleshooter, df-degrees of freedom, $\bar{x}$- sample mean, *- significant at 90% confidence interval.

In the diagnosis of failures-1, 2 and 3, it is seen from the average number of transitions made to information sources that the experts in general made fewer transitions than the novice troubleshooters (**Table 31**). Despite the reduced number of transitions made by the expert troubleshooters on most of the information sources, only the number of times the webcam was consulted is found to be statistically significant for all the three architectures as seen in **Table 32**. In case of architecture-1 however, the number of times the I/O list is consulted is also significantly higher for the novices over the experts.

Table 33—Statistical testing for difference in mean number of times information source is consulted between expert and novice troubleshooters for failure-4

Null hypothesis (H_0)	**Alternate hypothesis (H_1)**	**Par**	**Test Statistic**	**Outcome**
Failure-4				
Architecture-1				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	W_n	t= -0.49(p=0.354) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	IL_n	t= 0.49 (p=0.356) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	Off-L_n	t= -0.11 (p=0.466) df=1	Fail to reject
Architecture-2				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	W_n	t= -1.01 (p=0.236) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	IL_n	t= -1.76 (p=0.115) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	On-L_n	t= -2.83 (p=0.053) df=2	**Reject***
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	LV_n	t= -0.33 (p=0.398) df=1	Fail to reject
Architecture-3				
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	W_n	t= -0.95 (p=0.259) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	IL_n	t= 2.83 (p=0.947) df=2	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	On-L_n	t= -0.18 (p=0.445) df=1	Fail to reject
$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	S_n	t= 0.16 (p=0.451) df=1	Fail to reject

NOTE— Par- parameter, expert-expert troubleshooter, student-novice troubleshooter, df-degrees of freedom, $\bar{x}$- sample mean, *- significant at 90% confidence interval.

In the diagnosis of failure-4, it is seen with architecture-2 that the number of transitions made to the ladder program is significantly lesser for the experts in comparison with the novices. This illustrates that even in case of failures that are difficult to diagnose, the experts possessed a deeper search strategy. Apart from this, no significant difference was observed between the experts and novices as seen in **Table 33.**

A comparison of the time spent in per information screen and the number of transitions made between the information screens by both expert and novice troubleshooters in the diagnosis of all the failures using the three architectures is considered in **Table 34** and the results for the hypotheses tested for the difference in the means of these parameters is shown in **Table 35**.

Table 34—Summary statistics for comparison of time spent per information screen and number of transitions

Parameter	**Experts (n=24)**	**Novices (n=24)**
T/IS (mean)	89	47
T/IS (sd)	105	27.5
Transitions (mean)	23.2	38.1
Transitions (sd)	12.5	23.8
NOTE—T/IS- Time spent per information screen in seconds, n-number of samples.		

Table 35—Statistical testing for comparison of time spent per information screen and number of transitions

Parameter	**Null hypothesis (H_0)**	**Alternate hypothesis (H_1)**	**Test statistic**	**Outcome**
T/IS	$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} > 0$	t= -0.95 (p=0.259) df=1	**Reject**
Transitions	$\bar{x}_{expert} - \bar{x}_{student} = 0$	$\bar{x}_{expert} - \bar{x}_{student} < 0$	t= 2.83 (p=0.947) df=2	**Reject**

It can be seen that the experts spent significantly more time per information screen than the novice troubleshooters and made significantly lesser transitions between the information sources (number of information sources consulted) than the novices. It is also seen that the number of transitions made by the novice troubleshooters in the diagnosis of failures-1, 2 and 3 exhibits a strong positive correlation with the time taken to diagnose the failure as shown in **Table 36**. This indicates that the novice troubleshooters who made more transitions between the information sources spent more time to diagnose the failure. This correlation however, is not observed with expert troubleshooters.

Table 36—Pearson correlation coefficient for correlation between the various performance measures

Parameter	**T**		**PS**		**DT**		**Q**		**R**	
	E	N	E	N	E	N	E	N	E	N
Failures-1, 2 and 3										
PS	-0.85	-0.66								
DT	0.87	-0.08	-0.78	0						
Q	0.75	0.34	-0.64	-0.67	0.69	-0.17				
R	0.44	0.77	-0.46	-0.47	0.47	0.19	0.32	0.29		
IS	0.14	**0.76**	-0.12	**-0.52**	-0.05	-0.1	-0.08	0.46	-0.02	0.64
Failure-4										
PS	-0.37	-0.08								
DT	0.77	0.28	-0.33	0.01						
Q	-0.75	0.9	0.56	0.18	-0.9	-0.02				
R	0.98	0.24	-0.22	-0.15	0.71	0.81	-0.69	0.08		
IS	0.73	-0.83	-0.85	-0.16	0.64	0.17	-0.73	-0.92	0.63	0.18

NOTE— E-expert troubleshooter, N-novice troubleshooter, T- time taken to diagnose the failure in seconds, PS- performance score calculated using the model, R- number of runs (cycling the system), Q- number of questions asked by the troubleshooter, IS- number of information sources consulted by the troubleshooter, DT- number of diagnostic treatments performed.

Based on these observations, it is possible to conclude that the experts required fewer transitions between information sources, possessed better information retention and developed a deeper understanding of the failure situation while formulating or testing their hypotheses of the cause of failure. This also indicates that the ability of the students to utilize any given information source faded in comparison to the experts. In order to gather information to formulate or test any hypothesized cause of failure, the students would shuffle through several information screens indicating a diluted search strategy as illustrated in the timeline plot generated for a novice who induced the least performance with architecture-3 as shown in figure on page 107. This is substantiated from the findings in [53] that the experts possess better inferences of monitored parameters and seek further detailed information while devoting more time towards building up a representation whereas the novices focus mainly on the directly available data. The information selection, gathering and recall of the experts were far superior to that of novices.

It is also understood from [53] that novice troubleshooters are less selective with information collection and tend to formulate their hypotheses as information is encountered, whereas the experts formulate a limited number of assumptions after they have built a relationship pattern taking into account the entire information to come up with a framework of the causes of failure. This difference in diagnosis strategy of the experts and the novices also explains the ability of the experts to use the information sources efficiently especially in situations providing more information and the ineffectiveness of the students magnified with architecture-3.

Variability is considered to be an indicator of good performance because a good architecture is also one that facilitates consistent performance. The range and the standard deviation in a set of data are considered to be good indicators of variability [21]. It is seen from the range and the standard deviation for the time taken to diagnose the failures in **Table 18** and the hypothesis tested with architecture-3 in **Table 22** that architecture-3 induced the most consistent performance with the experts while it gave least consistent results with the students over all the performance metrics. This indicates that in order to effectively use architecture-3, a certain level of technical skill was necessary. The experts were able to use the capabilities on architecture-3 far better than the novices who found the information given by the tools very difficult to work with. A similar behavior was observed in the findings in [43] when comparing subjects with different levels of competence, wherein competent diagnosticians produced more consistent results. This is supplemented by the qualitative ratings of the architectures by the students in figures on pages 113 and 114 and the responses from the subjects on the survey detailed in **Appendix 2**, wherein the novices reported difficulties using architecture-3 and the experts on the other hand found it to be the best suited for diagnosis.

In the study performed [56] comparing the engineering design processes of freshmen and senior year students, it was found that the seniors produced higher quality of designs by gathering more information, considering more alternative solutions, performing more frequent transitions between the various design steps and spending more time in the design process. Although, engineering design and fault diagnosis are principally different processes, there exist some similarities and differences between the strategies employed by personnel with higher levels of expertise in both fields.

Similar to the behavior displayed by the senior design engineers in [56], the experts possessed better information gathering capabilities and applied the information conveyed by the various tools available on the architectures better than the novices during the diagnosis. The overall troubleshooting performance of the experts was better than the novice troubleshooters (students) in the diagnosis of failures-1, 2 and 3 with architecture-3, similar to the higher quality of designs created by senior design engineers in comparison to freshman engineers. In contrast, in our study, the experts made significantly fewer transitions between the various information screens and spent significantly more time per information screen in comparison to novice troubleshooters. In case of senior design engineers, more transitions between design steps and lesser time spent in a given design step where correlated to a higher quality score in [56]. The senior engineers made more transitions between design steps and consulted more information sources than freshman engineers.

In order to illustrate the difference in diagnosis strategies of the experts and novices, the timeline plots for two experts (best and least performance score) and two novice troubleshooters (best and least performance score) when diagnosing failures (1, 2 or 3) with architecture-3 are shown in **Figure 29**, **Figure 30**, **Figure 31** and **Figure 32** below. The timeline plots corresponding to all the 48 experiments performed with expert and novice troubleshooters can be found in **Appendix 3**.

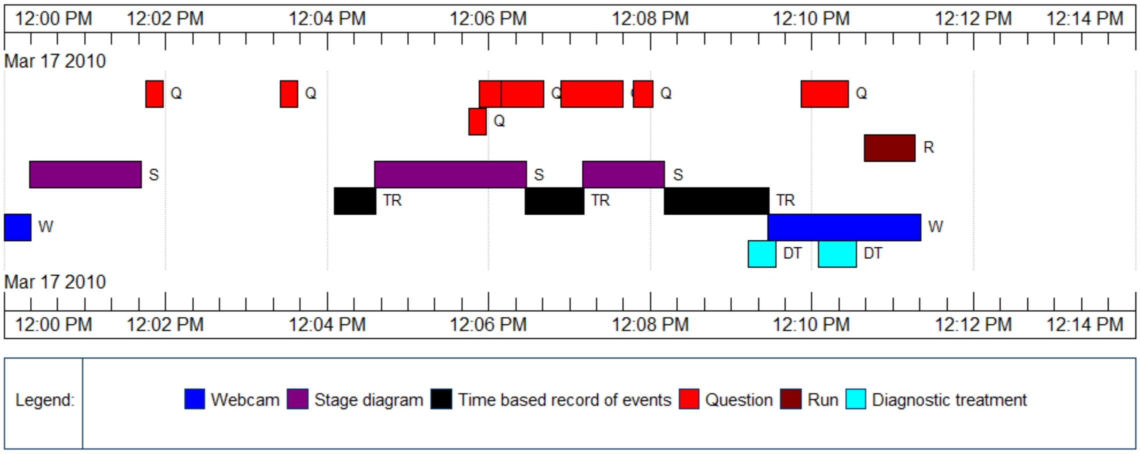


Figure 29—Timeline plot for distribution of activities in the diagnosis of failure-2 with architecture-3 by expert-8 (overall performance score 0.733)

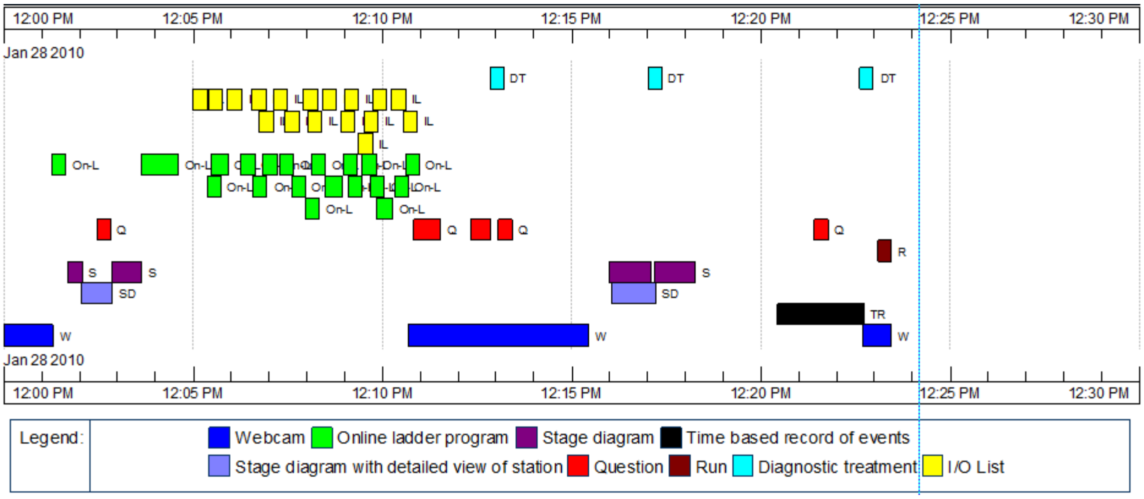


Figure 30—Timeline plot for distribution of activities in the diagnosis of failure-2 with architecture-3 by expert-4 (overall performance score: 0.490)

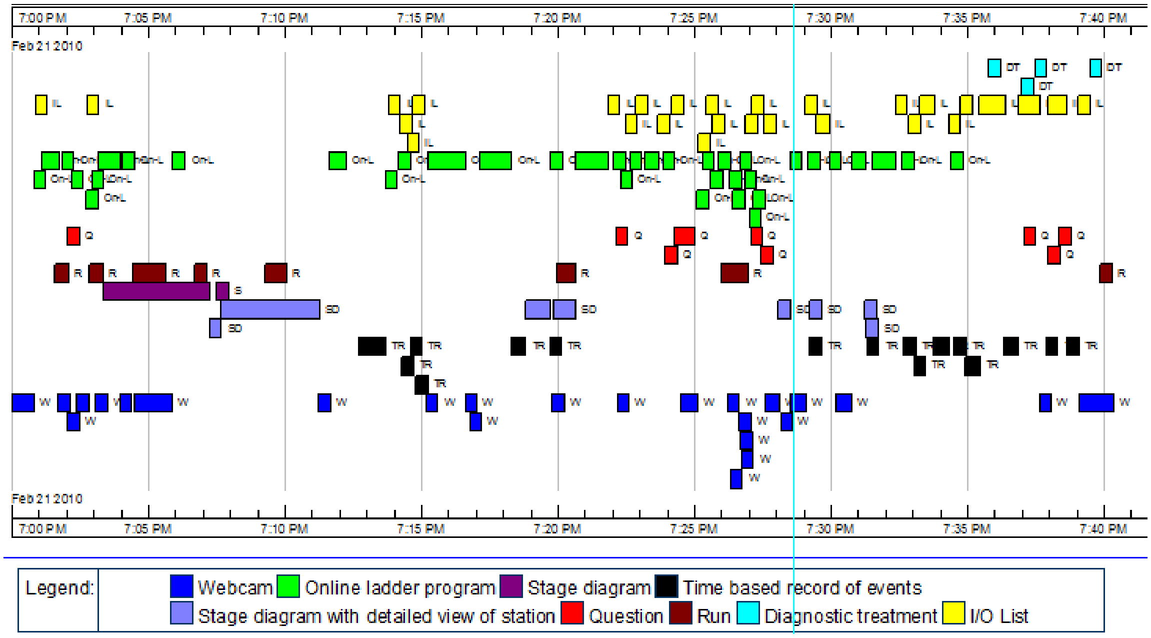


Figure 31—Timeline plot for distribution of activities in the diagnosis of failure-1 with architecture-3 by student-1 (overall performance score: 0.370)

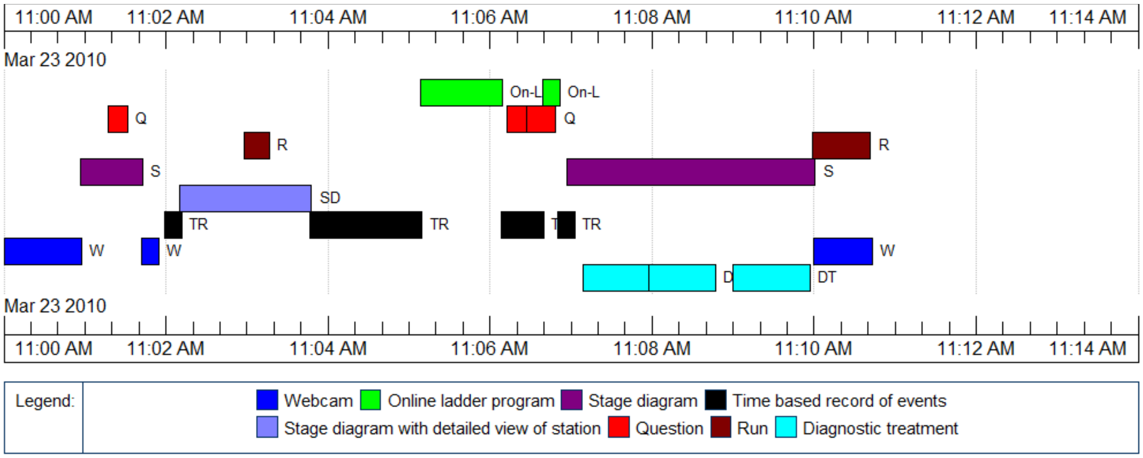


Figure 32—Timeline plot for distribution of activities in the diagnosis of failure-1 with architecture-3 by student-8 (overall performance score: 0.635)

From these timeline plots, it is possible to support that in order to diagnose failures effectively using architecture-3, which provides the maximum number of capabilities, an economy of transition between information sources is helpful. By transitioning lesser between information sources, remote troubleshooters were capable of extracting more relevant information and were able to formulate fewer and better hypotheses of the failure causes. This translated into a better overall remote troubleshooting performance by the experts in the diagnosis of failures with the architectures providing more capabilities. The effectiveness of such a strategy is also supported by the performance of a novice troubleshooter who tended to spend more time per information source as seen from **Figure 32**. In general, the novice troubleshooters, tended to scan through the information screens in order to observe any apparent abnormalities based on which they could formulate or test their hypotheses of the cause of failure. With the increase in the capabilities of the architectures, the number of information sources available increased. As a result, the novice troubleshooters experienced difficulties in utilizing the information available and exhibited a reduction in troubleshooting performance with the increased capabilities of the architectures.

4.5.6 Comparison between local engineer and novice operators

Interaction with the local operator is an important consideration in a remote diagnosis environment and the ability of the remote troubleshooter to make use of the skills of the operator could affect the troubleshooting performance. The operator is a source of information, intuition and an agent for the troubleshooter to carry out treatments. Especially in case of failure-4 that was difficult to troubleshoot, the intuition of the operator was used by the remote troubleshooter to come up with plausible hypotheses about the failure cause. The statistics for the different performance measures categorized on the basis of engineer and novice troubleshooters is tabulated in **Table 37**. The summary of the results of the hypotheses tested using the two sample t-test ($\alpha=0.05$) for the various performance metrics obtained between engineer and novice operators, when working with expert and novice troubleshooters is tabulated in **Table 38**.

.

Table 37—Summary statistics for comparison between engineer and novice operator

Parameter	**Expert**		**Student**	
	Engineer	**Novice**	**Engineer**	**Novice**
Failures-1, 2 and 3				
T (mean)	915.2	1280.9	987.6	1192
T (sd)	321.7	554.8	213.5	496.6
PS (mean)	0.497	0.440	0.465	0.427
PS (sd)	0.140	0.139	0.111	0.100
R (mean)	3	4.3	3.7	3.7
R (sd)	2.2	2	1.1	2.2
Q (mean)	4.7	7.8	10.2	5.6
Q (sd)	3.3	5.7	4.4	3.3
IS (mean)	22.7	25.1	37.1	41.6
IS (sd)	14.3	12.5	25.1	26.9
DT (mean)	3.0	4.3	3.7	3.7
DT (sd)	1.9	3.2	1.5	1.5
T-QT	187	338	288	256
T-QT (sd)	101	241	157	130
Failure-4				
T (mean)	2175	2283	1862	2159
T (sd)	701.5	537.2	262.7	632.7
PS (mean)	0.338	0.307	0.318	0.303
PS (sd)	0.102	0.046	0.045	0.067
R (mean)	15	16.7	16.7	20
R (sd)	10	10.4	2.9	0.6
Q (mean)	19	13.3	11.7	17
Q (sd)	12.3	3.8	9.1	7.9
IS (mean)	26.7	25.7	35.7	32.3
IS (sd)	21.2	7.2	7.1	7.1
DT (mean)	4.7	4.7	3.7	4.7
DT (sd)	3.5	1.1	2.1	1.2
T-QT	343	349	559	492
T-QT (sd)	140	117	213	306
NOTE—T- time taken to diagnose the failure in seconds, PS- performance score calculated using the model, R- number of runs (cycling the system), Q- number of questions asked by the troubleshooter, IS- number of information sources consulted by the troubleshooter, DT- number of diagnostic treatments performed, T-QT-time spent in asking questions and performing diagnostic treatments in seconds (interacting with the operator).				

It is observed that the average time taken to diagnose failures-1, 2 and 3 with engineer operator is lesser than that of novice operator over all the three architectures. This difference is greater with experts than novice troubleshooters. The time spent in asking questions and performing diagnostic treatments is lesser with an engineer operator, in the case of expert troubleshooters. Another important consideration is that the expert troubleshooters tended to ask lesser questions of engineer operators than novice operators. The students on the other hand asked more questions of the engineer operator than the novice operator indicating their greater dependence on the responses given by the operators.

In the diagnosis of the failures-1, 2 and 3 with expert troubleshooters alone, it can be seen from the hypothesis tested for time taken to diagnose the failure and the average time taken to diagnose the failures with engineer operators that, engineer operators enabled to diagnose the failure in significantly lesser time than novice operators. The engineer operators possessed better technical capabilities such as checking the status of the I/Os using the PLC program when online with the PLC. They were capable of understanding the technical jargon used by the troubleshooter and could perform diagnostic treatments with reduced guidance during treatment. In case of the novice operators, however, the troubleshooter needed to provide more elaborate explanation to enable him understand the treatment the troubleshooter required to be performed by him and better guidance to perform the treatment in comparison to engineer operators. This is supplemented by the significantly lesser time spent in asking questions and performing diagnostic treatments with an engineer operator in comparison to a novice operator, in the case of expert troubleshooters in the diagnosis of failures-1, 2 and 3 (**Table 38**).

The hypotheses related to the number of information sources consulted, the number of questions asked, number of times the system is cycled and the number of diagnostic treatments performed reveal that there is no significant difference between the diagnosis with engineer and novice operators. This could be attributed to the fact that these parameters are more dependent on the architecture used and the strategies of the troubleshooter as seen from the reduced difference over these metrics in **Table 37** with operators. This is also reflected in the inability to reject the null hypothesis tested for the overall performance score which incorporates the qualitative attributes along with the quantitative performance metrics.

Table 38—Statistical testing for difference in mean performance with engineer and novice operators

Null hypothesis (H_0)	**Alternate hypothesis (H_1)**	**Par**	**Test Statistic (experts)**	**Outcome (experts)**	**Test Statistic (students)**	**Outcome (students)**
Failures-1, 2 and 3						
$\boldsymbol{\bar{x}_{engineer} - \bar{x}_{novice} = 0}$	$\boldsymbol{\bar{x}_{engineer} - \bar{x}_{novice}} < 0$	**T**	t= -1.71 (p=0.056) df=12	**Reject***	t= -1.14 (p=0.141) df=10	Fail to reject
$\bar{x}_{engineer} - \bar{x}_{novice} = 0$	$\bar{x}_{engineer} - \bar{x}_{novice} > 0$	PS	t=0.86 (p=0.201) df=15	Fail to reject	t=-0.76 (p=0.230) df=15	Fail to reject
$\bar{x}_{engineer} - \bar{x}_{novice} = 0$	$\bar{x}_{engineer} - \bar{x}_{novice} < 0$	R	t= -1.12 (p=0.860) df=15	Fail to reject	t= -0.4 (p=0.348) df=11	Fail to reject
$\bar{x}_{engineer} - \bar{x}_{novice} = 0$	$\bar{x}_{engineer} - \bar{x}_{novice} < 0$	IS	t= -0.39 (p=0.352) df=15	Fail to reject	t= -0.36 (p=0.361) df=15	Fail to reject
$\bar{x}_{engineer} - \bar{x}_{novice} = 0$	$\bar{x}_{engineer} - \bar{x}_{novice} > 0$	Q	t= -1.42 (p=0.909) df=12	Fail to reject	t= 2.54 (p=0.012) df=14	**Reject**
$\bar{x}_{engineer} - \bar{x}_{novice} = 0$	$\bar{x}_{engineer} - \bar{x}_{novice} < 0$	DT	t= -1.09 (p=0.149) df=12	Fail to reject	t= 0 (p=0.500) df=16	Fail to reject
$\bar{x}_{engineer} - \bar{x}_{novice} = 0$	$\bar{x}_{engineer} - \bar{x}_{novice} < 0$	T-QT	t= -1.73 (p=0.057) df=10	**Reject***	t= 0.47 (p=0.677) df=15	Fail to reject
Failure-4						
$\bar{x}_{engineer} - \bar{x}_{novice} = 0$	$\bar{x}_{engineer} - \bar{x}_{novice} < 0$	T	t= -0.21 (p=0.423) df=3	Fail to reject	t= -0.75 (p=0.265) df=2	Fail to reject
$\bar{x}_{engineer} - \bar{x}_{novice} = 0$	$\bar{x}_{engineer} - \bar{x}_{novice} > 0$	PS	t=0.47 (p=0.344) df=2	Fail to reject	t=-0.32 (p=0.384) df=3	Fail to reject
$\bar{x}_{engineer} - \bar{x}_{novice} = 0$	$\bar{x}_{engineer} - \bar{x}_{novice} < 0$	R	t= -0.2 (p=0.427) df=3	Fail to reject	t= -2.16 (p=0.082) df=2	Fail to reject
$\bar{x}_{engineer} - \bar{x}_{novice} = 0$	$\bar{x}_{engineer} - \bar{x}_{novice} < 0$	IS	t= 0.08 (p=0.527) df=2	Fail to reject	t= 0.58 (p=0.702) df=4	Fail to reject
$\bar{x}_{engineer} - \bar{x}_{novice} = 0$	$\bar{x}_{engineer} - \bar{x}_{novice} < 0$	Q	t= 0.76 (p=0.737) df=2	Fail to reject	t= -0.77 (p=0.25) df=3	Fail to reject
$\bar{x}_{engineer} - \bar{x}_{novice} = 0$	$\bar{x}_{engineer} - \bar{x}_{novice} < 0$	DT	t= 0 (p=0.5) df=2	Support	t= -0.73 (p=0.26) df=3	Fail to reject
$\bar{x}_{engineer} - \bar{x}_{novice} = 0$	$\bar{x}_{engineer} - \bar{x}_{novice} < 0$	T-QT	t= -0.06 (p=0.478) df=3	Fail to reject	t= 0.31 (p=0.612) df=3	Fail to reject
NOTE— Par- parameter, df-degrees of freedom. $\bar{x}$-sample mean, *-significant at 90% confidence interval.						

In the diagnosis of failure-4, however, there was no significant difference observed in terms of the performance measures between local engineer and novice operators for both expert and novice troubleshooters.

The effect of the operator is seen to be more critical with architecture-1 because of the lack of many additional sources of information for the remote expert. The number of questions asked of the operator and the number of explicit diagnostic treatments performed are maximum with architecture-1 as seen from **Table 18**. The ANOVA analysis pointed out unusual observations for the diagnosis of failure-3 wherein a novice operator introduced difficulty in the diagnosis with architecture-1. On the diagnosis of failure-3, the search path adopted by the troubleshooter with a novice operator is interesting as seen from the interaction plot for time on **Figure 20** and **Figure 21**. An expert, who adopted a search path by first eliminating software issues before the hardware took the longest time to diagnose the failure but a student who started with the hardware was able to diagnose the failure in the least time. This discrepancy was eliminated with an engineer operator.

Despite the additional information sources available on architectures-2 and 3, it was seen that the students were more reliant on the information given by the operator as seen from the hypothesis tested for the difference in the number of questions asked and the difference in performance with engineer and novice operators in the architecture-operator interaction plot for time on **Figure 20**. When this is combined with the difference in the number of information sources consulted between the experts and students it could be concluded that the quality and usefulness of treatments performed and questions answered by the engineer operator with expert troubleshooters was better than that by a novice operator. The expert troubleshooters were capable of utilizing the engineer operator better than the students.

4.5.7 Qualitative ratings of the architectures

The qualitative attributes of the architectures were scored by the experts and the students in the survey performed after the experiments. The results are shown in **Figure**

33 and **Figure 34**. Reliability, accessibility, objectiveness, quality of treatments performed by the operator and cognitive reasoning are benefit attributes and so higher the score, the greater the preference. The requirement of a skilled operator on the other hand is a cost attribute indicating that lesser the value, better the preference.

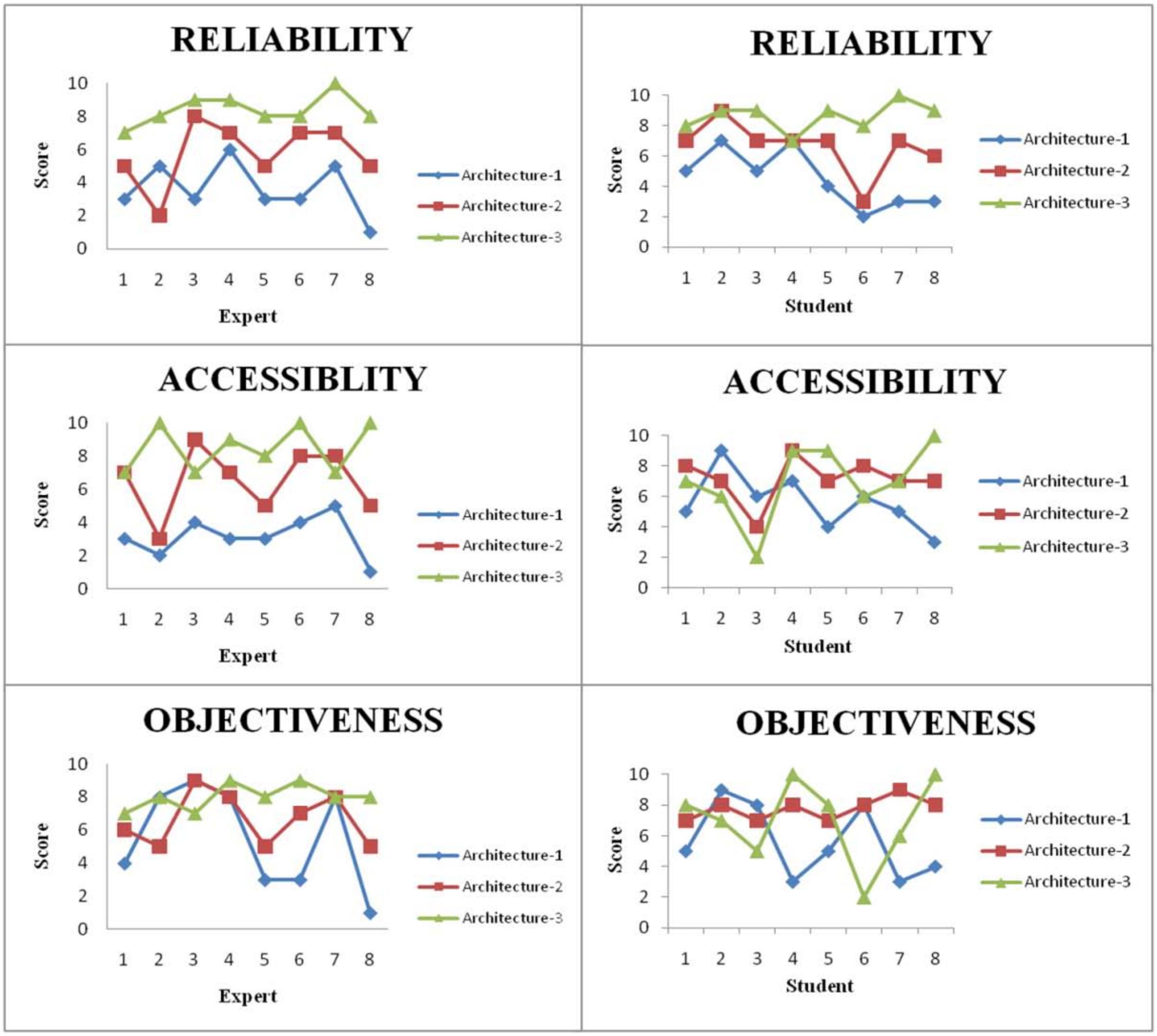


Figure 33—Qualitative ratings of architectures by experts and students part 1

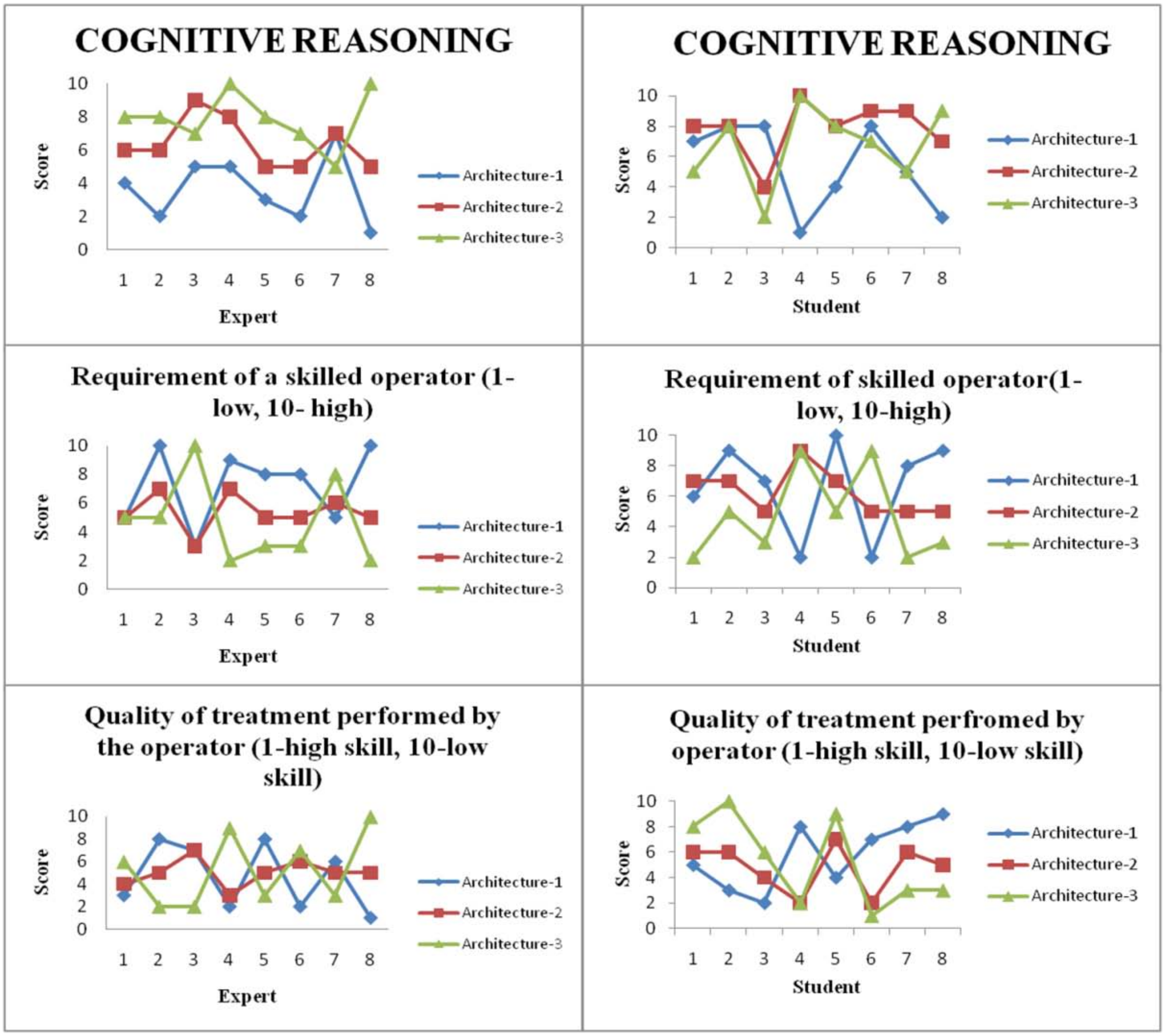


Figure 34—Qualitative ratings of architecturesby experts and students part 2

As discussed earlier, the students in general were not at ease with using some of the advanced capabilities on architecture-3 and found architecture-2 easier to use and this is reflected in this scoring. The experts on the other hand found architecture-3 to be the best of the three architectures and were able to gain maximum information from the tools with reduced operator interaction. Although both experts and students felt that the reliability of architecture-3 was higher, the students found architecture-3 to be less objective and less supportive of cognitive reasoning. Among the experts, expert 3 and 7 diagnosed failure-4 with atchitecture-3. Their qualitative ratings for architecture-3 were not better than that of architecture-2 which is an indicator of the difficulty faced in the diagnosis of failure-4 with the increased capabilities of architecture-3. The additional comments made by the experts and the students about the architectures can be found in **Appendix 2**.

The subjects were asked to rank the architectures in terms of their suitability to diagnose the failures. The following **Figure 35** represents the overall rankings of the architectures by the experts. A ranking of 1 indicates that the architecture is most suitable and a ranking of 3 indicates that the architecture is least suitable. This supplements that the information available on architecture-3 was not really helpful in the diagnosis of failure-4.

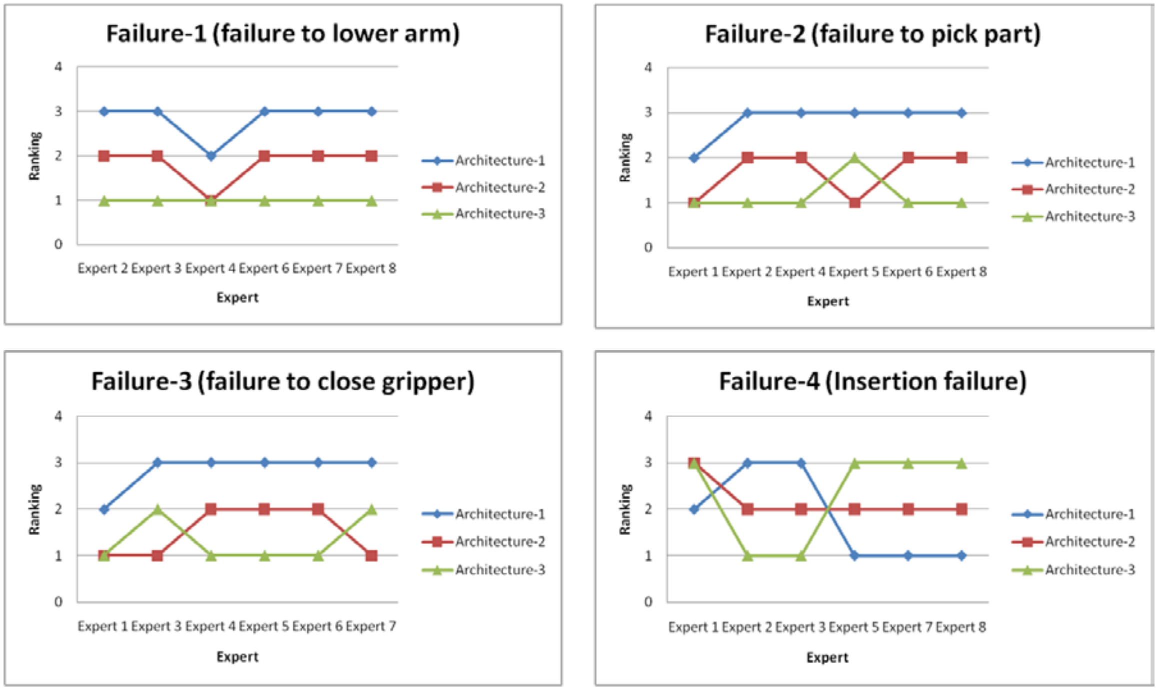


Figure 35—Overall suitability rankings of the architectures by experts

In another survey performed, the experts and the students were asked their preference of the remote diagnosis tools in terms of the percentage of tools they would prefer to use in the diagnosis of each of the failures. The average percentages of these preferences are tabulated in **Table 39**. It can be seen that in the diagnosis of failures-1, 2 and 3 by expert troubleshooters, the conversation with the operator and the live video feedback are considered important by the subjects. This is followed by the ladder diagram of the PLC and the stage diagram based representation of the system. The ladder diagram is necessary to understand the interaction between the various inputs and outputs in the system and the logical sequence of operation. The time based record of the system is considered most important for the diagnosis of failure-3 involving a loss of communication between the PLC's output module and the solenoid valve that controls the gripper. This is because the time based record of events records both the status of the PLC's I/Os and the analog voltage values, thereby facilitating the isolation of the cause of failure-3 with lesser searching.

The voltage waveforms is similar to the time based record except that it does not depict the status of the I/Os along with the analog voltage values. This tool was hence less preferred in comparison to the time based record of events. The use of the video playback of the failed operation was another useful tool that was used by the remote troubleshooters to gain a better understanding of the failure without the requirement to recreate the failure by running the system. In case of failure-4, however, the interaction with the operator and the video feedback were found to be the most important tools supplementing some of the previous discussion related to the diagnosis of failure-4.

The novice troubleshooters on the other hand tended to place a greater deal of importance on the interaction with the operator than the experts despite the additional tools provided, consistent with the discussion comparing the expert and the novice troubleshooters.

Table 39—Preference of remote diagnosis tools by expert and novice troubleshooters

F	Con	N	Ladder diagram	Voltage waveform	Pressure sensor	Load cell	Stage diagram	Time based record	Video playback	Forcing I/O
Expert troubleshooters										
1	21.7	13.3	20	5.8			22.5	10	4.2	1.7
2	30.8	12.5	20	8.3			15	11.7	1.7	
3	24.2	7.5	25.8	2.5	0.8		13.3	19.2	4.2	1.7
4	48.3	22.5	14.2	1.7		1.7	6.7		3.3	1.7
Novice troubleshooters (students)										
1	39.2	20.8	5.8	8.3	0.8		7.5	5.8	3.3	0.8
2	40	18.3	16.7	10.8			5.8	6.7	1.7	
3	36.7	18.7	15	14.2	1.7		5	9.2	0.8	0.8
4	45	28.3	15.8			1.7	5.8		2.5	0.8
NOTE—F- failure type, Con-conversation with the operator, N-network camera. The numbers in the cells represent the average percentage (n=6/ failure).										

4.5.8 Design for remote diagnosis

The design for remote diagnosis involves identification of the most effective alternative to achieve optimum troubleshooting performance in a remote diagnosis environment based on the performance of the different combinations of architecture, type of failure and skill level of the operator. Architecture-3, which is representative of the level-2 type architectures proposed in [2], has the most capabilities and the tools needed to diagnose the different failures that occur in the PLC based discrete automated system. However, the additional capabilities in architecture-3 are not always required and the decision regarding the choice of architecture can be made based on cost, existing conditions of the skill level of the operator, nature of failure and level of competence of the troubleshooter. Owing to the increased capabilities, architecture-3 is the highest in terms of cost, followed by architecture-2. Architecture-1, with the minimum tools, is least expensive. The prescribed architecture for the various alternatives in this study is presented in **Table 40**.

Table 40—Prescribed architecture for expert and student troubleshooters

Failure	Operator	Recommended architecture
Expert troubleshooters		
1	Engineer	2
1	Novice	3
2	Engineer	2
2	Novice	3
3	Engineer	2
3	Novice	3
4	Engineer	1 and stage diagram
4	Novice	1 and stage diagram
Less competent troubleshooters (students)		
1	Engineer	2
1	Novice	2
2	Engineer	2
2	Novice	2
3	Engineer	3
3	Novice	3
4	Engineer	2
4	Novice	2

In case of failures that render the system incapable of being operated or if the cost of running the system under the partially failed condition is too high, then architecture-3 would be recommended because of the availability of time-based record of the events and reduced requirement to cycle the system.

The expert troubleshooters considered architecture-3 to be most helpful because of their ability to use the tools effectively. If the system operators are novices, the architecture with most tools is recommended for failures related to system variables that can be measured or monitored because it serves to reduce the interaction with the novice operator, reduces the requirement to cycle the system and provides a more reliable decision making platform for the experts. The presence of engineer operators however, allows the deployment of slightly lower cost architecture, with slightly lesser capabilities, largely because of the ability of an engineer operator to provide reliable technical assistance to the troubleshooter.

For the failures that are purely related to the hardware of the system and independent of any measured or monitored variables, such as failure-4, intuition of the operator and good quality of video feedback are of immense help in diagnosis. However, the stage diagram based human machine interface to the automated system allows the troubleshooter to eliminate any issues with the logic and create system awareness. So, the combination of the stage diagram and simple tools allowing video feedback with multiple views and un-interrupted communication with the operator would be the recommended tool set for this type of failure with both engineer and novice operators.

In the case of less competent troubleshooters (students) however, their discomfort in using the tools providing more information, suggests that architecture with lesser information sources but with certain capabilities allowing further processing of data and support for decision making would be recommended.

CHAPTER V

SUMMARY, CONCLUSIONS AND FUTURE SCOPE

5.1 Summary

Three levels of remote diagnosis architectures were considered for a PLC based automated assembly system. The capabilities of these levels of remote diagnosis were successfully implemented using a set of hardware and software tools. The different types of failures in automated assembly systems were categorized and commonly occurring failures were duplicated in an automated assembly line in the laboratory. Expert and novice troubleshooters used the three architectures to determine the root cause of these failures remotely, while working with novice and engineer operators. Measurements of the quantitative metrics for troubleshooting performance were made for each of the experiments performed. The qualitative measures were assessed by means of a survey performed with the subjects.

5.2 Conclusions

The experiments performed provide an insight into the factors that are involved in a remote diagnosis environment and how they affect the ability of a remote diagnosis architecture to facilitate the troubleshooter to remotely diagnose failures. The model for measuring remote troubleshooting performance has been successful in combining the performance of the alternatives over the multiple attributes to help with decision making. The model’s scoring is consistent with the general trend of the quantitative performance measures on most situations. The only exception was the diagnosis of failure-4, where the model predicted architecture-2 and 3 to be partially better than architecture-1. The main conclusions of this study are summarized as follows:

(i) In the diagnosis of failures-1, 2 or 3 by remote expert troubleshooters with local engineer or novice operators, the overall remote troubleshooting performance with architecture-3 is significantly better than that with architecute-2 and architecture-1.

(ii) In the diagnosis of failures-1, 2 or 3 by remote expert troubleshooters with local engineer or novice operators, the overall remote troubleshooting performance with architecture-2 is significantly better than that with architecture-1.

(iii) In the diagnosis of failure-4 by remote expert troubleshooters with local engineer or novice operators, no significant difference could be observed among the architectures in terms of overall troubleshooting performance on the basis of two samples. These suggest that for failures related to measured or monitored system variables, remote troubleshooting performance improves with the increase in capabilities of the remote diagnosis architectures.

(iv) In the diagnosis of failures-1, 2, 3 or 4 by remote novice troubleshooters with local engineer or novice operators, there is no significant difference among the architectures in terms of the overall remote troubleshooting performance. This suggests that the ability to utilize the remote diagnosis architectures to improve troubleshooting performance depends on the level of technical expertise of the remote troubleshooter.

(v) For expert or novice remote troubleshooters, in terms of overall remote troubleshooting performance, failure-4 is significantly lower than failures-1, 2 or 3 using the three architectures with local engineer or novice operators. This suggests that failures unrelated to measured or monitored system variables such as failure-4 are more difficult to diagnose remotely and induce lesser remote troubleshooting performance than the ones related to measured system variables (failures-1, 2 and 3).

(vi) In the diagnosis of failures-1, 2, 3 with local engineer or novice operators using architectures- 3, the overall remote troubleshooting performance of the experts is significantly better than that of remote novice troubleshooters. In the diagnosis of all the failure failures with local engineer or novice operators using architectures-1, 2 or 3, remote experts made significantly fewer transitions between information screens than novices. Also, the remote experts spent significantly more time per information screen than the novices. This suggests that the experts possessed better information gathering capabilities than novice troubleshooters and were better equipped to utilize some of the advanced capabilities to improve remote troubleshooting performance.

(vii) In the diagnosis of failures-1, 2, 3 or 4 with remote expert or novice troubleshooters, with the three architectures, no significant differences were observed between local engineer and novice operators in terms of overall remote troubleshooting performance.

5.3 Challenges faced and suggested improvements

The security of transactions over the internet is one of the most important challenges in a remote environment [37]. In this work, this was addressed by the use of VPN, where in the remote expert was allowed authorized access within the firewall of the local area network at the system site. However, in the industrial environment, customers would seldom allow an external connection to access their network and the components on their network owing to stringent security policies.

A fast and reliable internet connection is vital for the success of any remote diagnosis environment, especially with video transmission requirements. During the experiments, at times owing to lower connection speed and network traffic, the remote troubleshooter experienced difficulty viewing the continuous stream of the video through the network camera. This was compounded by the rapid movement of the actuators in the system resulting in a choppy or poor quality of video. However, relatively slow moving parts and still images were of acceptable visual quality. In order to compensate this, a video playback of the process was provided to the troubleshooter and better description of the failure was required from the system operator. So, a web camera with good image quality, compression and reliable transmission over the internet is crucial for a healthy remote diagnosis environment.

The experts also felt that that multiple views of the system using multiple cameras would be useful during diagnosis. This would enable the remote personnel to be able to see different views at the same time. E.g. in diagnosing failure-4 (insertion failure), the experts felt the need to view both the side of the track containing the pegs and the pickup operation of the gripper at the same time. During the experiments, we had to cycle the system additionally for the expert to get a different view of the operation.

During the experiments, the operator was given a headset that was wired to the computer in order to communicate verbally with the troubleshooter. However, when the troubleshooter instructed the operator to perform some treatment or gather information about a part of the system, the operator would go to the system site, leaving the headset behind. During this interval where the operator implements the instructions of the troubleshooter, the operator would not be in a position to speak to the troubleshooter. In cases involving failures that are difficult to diagnose, the remote troubleshooter tries to invoke the intelligence and intuition of the operator in order to come up with plausible explanations for the causes of failure. Thinking out loud by the operator while looking at the system and being in constant conversation with the operator at the failure site, enables to troubleshooter invoke the intuition of the operator. This could be accomplished by means of technology such as wireless headset and it enables the troubleshooter to be in constant touch with the operator, as the operator performs a prescribed treatment or looks at a specific component in the system. On similar lines, the use of a mobile camera (mounted to the operator) was also suggested because it would allow the expert to see what the operator's field of view and prescribe treatments dynamically.

Finally, the troubleshooters also felt the need for better integration of the video of the process with the other diagnosis tools such as the stage diagram, Labview® voltage waveforms and the online ladder program which give a real time status of the system. Without this, the troubleshooter would have to frequently shuttle between the video and the information sources which could result in him losing track of an event, missing an event that happened and so on. This was accomplished to a certain extent by resizing the windows on the computer screen. However, more efficient integration of the video feedback or multiple display screens for the troubleshooter would be desirable.

Based on the experiments performed with the remote diagnosis architectures, the following recommendations can be made for the design of automated systems to facilitate remote diagnosis. Any automated system is comprised of actuators, sensors and controller. In order to facilitate remote diagnosis, the following recommendations can be made for the automated system.

1. **Actuators**: In the automated system considered in the study, the actuators were either pneumatic or electrical. The electrical motors were driven by means of relays and the pneumatic actuators were driven by means of solenoid valves. During the remote diagnosis process, there were several occasions when the troubleshooters required the operators to check the status of a solenoid valve or relay when the system was in operation. Although engineer operators were capable of measuring the voltage across the points using electrical measurement devices, the novice operators in general found it difficult to perform these tests. The provision of indicators on these devices, however, would make it possible to easily identify the status of an actuator. Hence, to enable the local operators to identify whether the actuators receive the voltage necessary to run the system, the presence of indicators (LEDs) on the solenoid valves or relays is highly recommended. In the remote diagnosis of the system considered in this study, which is driven by pressurized air, the most commonly tested hypothesis for the failure cause was the sufficiency of air pressure in a pneumatic line. Although, it may not be feasible to install pressure sensors in every pneumatic line, the installation of pressure transducers on key pneumatic lines helps quickly identify issues related to pressure.
2. **Sensors**: There were situations encountered in this study when the lack of an indicator on a sensor introduced difficulty in the remote diagnosis process. Similar to the indicators for denoting the status of the actuators, sensors equipped

with indicators make remote troubleshooting easier even if the local operator is a novice.

3. **Controller**: In this study, the programmable logic controller was used for controlling the discrete event system. It was observed during the experiments that the failures caused due to inefficiently written programs tend to induce unexpected symptoms that could misguide the troubleshooter onto an incorrect search strategy. Fault tolerance [57] in control algorithms which involves accommodating for some basic failures with minor corrective actions, greatly reduces remote diagnosis complexity. Pictures of the programmable logic controller, chassis, I/O module allocations etc. help the remote troubleshooter guide operators better in verifying physical connections related to the PLC.
4. **System documentation**: System documentation with detailed descriptions of the system and pictorial layouts with the sensors and actuators labeled provide a quick reference to troubleshooters when they need to check of a particular variable is sensed or controlled. The detailed list of PLC inputs and outputs with the addresses and their description is also found useful. A record of replacements made and recent changes to the system, parts assembled or manufactured is useful especially with failures of the system that are not related to measured or monitored parameters.

5.4 Future directions

5.4.1 Automated system

Introduction of additional complexity in the automated system could make it susceptible to more failures, thereby, enabling a larger set of failures to be introduced. The existing PLC based robotic assembly system could be modified. Instead of single assembly operation at each assembly station, we could have multiple assembly operations. This would require a linear indexing mechanism at each assembly station. One example of a linear indexing mechanism is to have multiple part stoppers with different extensions on the cylinders. The base part could be modified to facilitate such indexing.

The rotational degree of the freedom of the gripper could be used to come up with additional variations in the assembly such as rotating the part prior to insertion or rotating a part after insertion into the hole in the base part. Likewise, the part feeder subsystem could be modified to have the parts automatically fed to the feeder instead of being manually placed.

The remote diagnosis of a PLC based discrete automated system is considered in this study with human troubleshooters. The work can be extended to the continuous domain, since many real world applications such as power plants and nuclear applications are popular avenues for remote diagnosis considering the accessibility and safety issues involved. However, control and diagnosis of continuous systems involves strategies different from that of discrete systems because in the continuous domain, the purpose of process control is to maintain uniformity of output by adjusting certain variables. Likewise an empirical evaluation of remote diagnosis architectures for computer controlled automated systems would involve different strategies for the development of these architectures.

5.4.2 Automated system failures

In this work, only a single point of failure is considered. When a failure is introduced in the system, it is not accompanied by any other failure. However, in industrial applications, there could be multiple failure causes [58] occurring at the same time. Although multiple failure causes are not common, they introduce additional set of challenges for the troubleshooter to encounter during diagnosis.

5.4.3 Experiments and factors

The fractional factorial design of experiments was used in this study and primarily the main effects of the factors were considered. However, there is scope to examine the effect of the interactions between the factors including the higher order interactions such as 3-way and 4-way interactions. This could provide deeper insight into the combination of factors that would optimize remote troubleshooting performance. However, in order to study the interactions more experiments need to be performed with more replicates in the data sets. Additionally, experiments can be modified to accommodate voice protocol analysis [56] to understand the thought process of the remote troubleshooter during the diagnosis process.

5.4.4 Remote diagnosis environment

Automated failure diagnosis based on system models in order to identify abnormal events is widely used in the industry. In this work, a timing based model of the system is used in order to identify the failed event in the system. Further advanced methodologies such as petri- net may be used for building system models [31], [35]. The use of alarm messages can be analyzed to incorporate automated failure alarming that can be trusted by the remote troubleshooter. The stage diagram based representation of the system could be improved to incorporate the abnormality detection for the inputs (sensors) in the system.

Collaborative maintenance [12], [32], [33] is widely used in the industry. In the case of integrated automated systems, it is possible to have subsystems from different suppliers/ manufacturers [4] and system integrators. In such cases, there may be certain failures that are beyond the scope of the system integrator and their diagnosis may require multiple parties to work together at the same time. This provides another dimension for the analysis of the remote diagnosis problem that is, how the diagnosis tools for a single remote expert system would fare on the collaborative platform and what additional tools and communication media would be necessary. Also the interactions between multiple troubleshooters would present an interesting proposition.

Use of a physical replica of the system for failure diagnosis is very common in the space operations domain [36], [37], [38]. Any treatment that is suggested for the system is first tested on the replica before being implemented on the system. However, it is suggested that a replica would be of real use in situations where the cost of implementing a treatment or the cost of an incorrect treatment is high, typical of space stations and related systems. Although, a replica of the pick and place robot was built, it was not used owing to the simplicity of the system and shipping cost involved. However, there is scope to examine the utility of a replica of the system in the remote diagnosis of complex automated systems. In cases where security considerations are too prohibitive, a replica could be useful. Also, in cases where the cost of implementing a treatment is high or a subsystem is permanently damaged, the replica of the system could be used to gain important knowledge about how the system is supposed to behave or how it operates in the failed state. The use of 3D animated models of the system could also be considered for remote diagnosis, apart from the physical replica.

REFERENCES

[1] R. Wei, W. Jian, Z. Hao, Y. Junwei and W. Qidi, "The research of remote fault diagnosis system in manufacturing," in *Proc. 3rd World Congress on Intelligent Control and Automation*, 2000, vol.1, pp. 712-714.

[2] H. Wohlwend, "E-diagnostics guidebook: Revision 2.1," *International Sematech Manufacturing Initiative, ISMI*, July 15th 2005, [Online], Available: http://ismi.sematech.org. [Accessed: Jan.20, 2010]

[3] A. Wolfram and R. Isermann, "Component based tele-diagnosis approach to a textile machine," *Control Engineering Practice,* vol. 10, no. 11, pp. 1251-1257, 2002.

[4] K. Feldmann and J. Gohringer, "Multimedia system for remote diagnosis of complex placement machines," *The International Journal of Advanced Manufacturing Technology,* vol. 15, no. 10, pp. 722-729, 1999.

[5] M. Mori, M. Fujishima, M. Komatsu, B. Zhao and Y. Liu, "Development of remote monitoring and maintenance system for machine tools," *CIRP Annals - Manufacturing Technology*, vol. 57, no. 1, pp. 433-436, 2008.

[6] F. Qiao, H. Schlange, H. Meier and W. Massberg, "Internet-based remote access for a manufacturing-oriented teleservice," *The International Journal of Advanced Manufacturing Technology*, vol. 31, no. 7, pp. 825-832, 2007.

[7] Y. Kwon, R. Chiou and L. Stepanskiy, "Remote, condition-based maintenance for web-enabled robotic system," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 3, pp. 552-559, 2009.

[8] S. K. Ong, N. An and A. Y. C. Nee, "Web-based fault diagnostic and learning system," *The International Journal of Advanced Manufacturing Technology,* vol. 18, no. 7, pp. 502-511, 2001.

[9] C. Wang, L. Xu and W. Peng, "Conceptual design of remote monitoring and fault diagnosis systems," *Inf. Syst.*, vol. 32, no. 7, pp. 996-1004, 2007.

[10] W. Wang, P. Tse and J. Lee, “Remote machine maintenance system through internet and mobile communication,” *The International Journal of Advanced Manufacturing Technology*, vol. 31, no. 7, pp. 783-789, 2007.

[11] P. E. Miyagi and L. A. M. Riascos, “Modeling and analysis of fault-tolerant systems for machining operations based on petri nets,” *Control Engineering Practice*, vol. 14, pp. 397-408, 2006.

[12] J. F. Wang, P. W. Tse, L. S. He and R. W. Yeung, “Remote sensing, diagnosis and collaborative maintenance with web-enabled virtual instruments and mini-servers,” *The International Journal of Advanced Manufacturing Technology,* vol. 24, no. 9, pp. 764-772, 2004.

[13] Y. Lei, Z. He, Y. Zi and Q. Hu, “Fault diagnosis of rotating machinery based on multiple anfis combination with gas,” *Mechanical Systems and Signal Processing,* vol. 21, no. 5, pp. 2280-2294, 2007.

[14] S. F. Yuan and F. L. Chu, “Support vector machines-based fault diagnosis for turbo-pump rotor,” *Mechanical Systems and Signal Processing,* vol. 20, no. 4, pp. 939-952, 2006.

[15] S. S. Mandroli, A. K. Shrivastava and Y. Ding, “A survey of inspection strategy and sensor distribution studies in discrete-part manufacturing processes,” *IIE Transactions,* vol. 38, no. 4, pp. 309 - 328, 2006.

[16] M. Bhushan and R. Rengaswamy, “Design of sensor location based on various fault diagnostic observability and reliability criteria,” *Computers and Chemical Engineering,* vol. 24, pp. 735-741, 2000.

[17] H. M. Hsiung, C. K. Yii, H. R. Wen and C. F. Tien, “Development of an e-diagnostics/maintenance framework for semiconductor factories with security considerations,” *Advanced Engineering Informatics*, vol. 17, pp. 165-78, 2003.

[18] A. Muller, A. C. Marquez and B. Iung, “On the concept of e-maintenance: Review and current research,” *Reliability Engineering & System Safety*, vol. 93, no. 8, pp. 1165-1187, 2008.

[19] H. Li, T. Shi, S. Yang, Z. Li, Y. Tao *et al.*, "Internet-based remote diagnosis: Concept, system architecture and prototype," in *Proc. 3rd World Congress on Intelligent Control and Automation*, 2000, vol.1, pp. 719-723.

[20] S. Deb, S. Ghoshal, V. N. Malepati and D. L. Kleinman, "Tele-diagnosis: Remote monitoring of large-scale systems," in *Proc. Aerospace Conference, IEEE*, 2000, vol.6, pp. 31-42.

[21] S. R. Bereiter and S. M. Miller, "Troubleshooting and human factors in automated manufacturing systems," *Park Ridge, NJ: Noyes Data Corp.*, 1989.

[22] J. Lee, "E-manufacturing-fundamental, tools, and transformation," *Robotics and Computer-Integrated Manufacturing*, vol. 19, no. 6, pp. 501-507, 2003.

[23] W. Hudetz, "Teleservice-state-of-the-art and future developments," in *Industrial Electronics Society*, 1998, in *Proc. 24th Annual Conference of the IEEE*, 1998, vol.4, pp. 2103-2107.

[24] S. J. Chang, F. DiCesare and G. Goldbogen, "Failure propagation trees for diagnosis in manufacturing systems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 4, pp. 767-776, 1991.

[25] M. J. Stanton, "A fault monitoring architecture for the diagnosis of hardware and software faults in manufacturing systems," in *Proc. 7th IEEE International Conference on Emerging Technologies and Factory Automation*, 1999, vol.1, pp. 693-701.

[26] L. E. Holloway and B. H. Krogh, "Fault detection and diagnosis in manufacturing systems: A behavioral model approach," in *Proc. Rensselaer's Second International Conference on Compter Integrated Manufacturing*, 1990, pp. 252-259.

[27] C. Ming, Z. Jianzhi and Q. Wenhan, "Fault diagnosis system for automated assembly line," in *Proc. IEEE International Conference on Intelligent Procesing Systems*, 1997, vol.2, pp. 1478-1482.

[28] N. Hardy, D. Barnes and M. Lee, "Automatic diagnosis of task faults in flexible manufacturing systems," Robotica, vol.7, pp. 25-35, 1989.

[29] K. Althoefer, B. Lara, Y. H. Zweiri and L. D. Seneviratne, “Automated failure classification for assembly with self-tapping threaded fastenings using artificial neural networks,” in *Proc. Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* 2008, vol. 222, no. 6, pp. 1081-1095.

[30] C. Linderstam and B. A. T. Soderquist, “Monitoring the generic assembly operation for impact from gripping to finished insertion,” in *Proc. IEEE International Conference on Robotics and Automation*, 1996, vol.4, pp. 3330-3335.

[31] A. W. L. Yao, “Design and implementation of web-based diagnosis and management system for an fms,” *The International Journal of Advanced Manufacturing Technology,* vol. 26, no. 11, pp. 1379-1387, 2005.

[32] R. S. Lee, J. P. Tsai, J. N. Lee, Y. C. Kao, G. Lin *et al.*, “Collaborative virtual cutting verification and remote robot machining through the internet,” in *Proc. Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* 2000, vol. 214, no. 7, pp. 635-644.

[33] J. P. Tsai, Y. C. Kao and R. S. Lee, “Development of a remote collaborative forging engineering system,” *The International Journal of Advanced Manufacturing Technology,* vol. 19, no. 11, pp. 812-820, 2002.

[34] T.H. Hou, W.L. Liu and L. Lin, “Intelligent remote monitoring and diagnosis of manufacturing processes using an integrated approach of neural networks and rough sets,” *Journal of Intelligent Manufacturing,* vol. 14, no. 2, pp. 239-253, 2003.

[35] C. Lin, W. Shutao and X. Xiaowen, “Modeling and analysis of remote diagnosis using petri nets,” in *Proc. IEEE International Conference on Robotics and Biomimetics, ROBIO*, 2007, pp. 2133-2137.

[36] M. A. D. C. R. Price and R. B. Berka, “Space station operations simulation in the automated robotic maintenance of space station facility,” in *Proc. AlAA Space Programs and Technologies Conference and Exhibit*, 1993.

[37] T. N. H. Ueno, M. Oda and N. Inaba, “Autonomous cooperative robots for space structure assembly and maintenance,” in *Proc.7th International Symposium on Artificial Intelligence, Robotics and Automation in Space: i-SAIRAS, NARA*, 2003.

[38] J. McGuire and B. Roberts, “Hubble robotic servicing and de-orbit mission: Risk reduction and mitigation,” in *Proc. AIAA Space 2007 Conference and Exposisiton*, 2007, pp. 2699-2736.

[39] P. Csaszar, T. Tirpak and P. Nelson, “Optimization of a high-speed placement machine using tabu search algorithms,” *Annals of Operations Research,* vol. 96, no. 1, pp. 125-147, 2000.

[40] D. H. Ham and W. C. Yoon, “The effects of presenting functionally abstracted information in fault diagnosis tasks,” *Reliability Engineering & System Safety,* vol. 73, no. 2, pp. 103-119, 2001.

[41] G. A. Jamieson, “Empirical evaluation of an industrial application of ecological interface design,” in *Proc. 46th Annual Meeting of the Human Factors and Ergonomics Society*, 2002, pp. 536-540.

[42] J. H. Kim and P. H. Seong, “The effect of information types on diagnostic strategies in the information aid,” *Reliability Engineering & System Safety,* vol. 92, no. 2, pp. 171-186, 2007.

[43] J. B. Brooke and K. D. Duncan, “A comparison of hierarchically paged and scrolling displays for fault finding,” *Ergonomics,* vol. 26, no. 5, pp. 465 - 477, 1983.

[44] S. J. Hsieh, “Re-configurable dual-robot assembly system design, development and future directions,” *Industrial Robot: An International Journal,* vol. 30, pp. 250-257, 2003.

[45] “Texas A&M University VPN 3000 for Windows,” Sept. 2009. [Online], Available: https://hdc.tamu.edu/reference/documentation/?section_id=156. [Accessed: Jan. 10, 2010].

[46] “Skype,” 2003. [Online], Available: http://www.skype.com. [Accessed: Jan. 5, 2010].

[47] “Real VNC personal edition 4.3,” 2007. [Online], Available: http://www.realvnc.com. [Accessed: Jan. 8, 2010].

[48] S. J. Hsieh, “Work in progress - integrating technology for e-diagnosis of automated manufacturing system,” in *Proc. 34th ASEE/ IEEE Frontiers in Education Conference*, 2004, vol. 1, pp. T1C-12-13.

[49] “Youtube-broadcast yourself,” 2005. [Online], Available: http://www.youtube.com. [Accessed: Jan. 4, 2010].

[50] T. E. Leonik, “Home automation basics: Practical applications using visual basic 6,” *Indianapolis, IN: Prompt Publications*, 2000.

[51] W. B. Rouse, “Human problem solving performance in a fault diagnosis task,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 8, no. 4, pp. 258-271, 1978.

[52] C. Kolski and P. Millot, “Problems in telemaintenance and decision aid criteria for telemaintenance system design,” *International Journal of Industrial Ergonomics*, vol. 11, no. 2, pp. 99-106, 1993.

[53] J. M. Cellier, H. Eyrolle and C. Marine, “Expertise in dynamic environments,” *Ergonomics*, vol. 40, pp. 28-50, 1997.

[54] K. P. Yoon and C. L. Hwang, “Multiple attribute decision making: An introduction,” *Thousand Oaks, CA: Sage Publications*, 1995.

[55] C. Parkan and M. L. Wu, “On the equivalence of operational performance measurement and multiple attribute decision making,” *International Journal of Production Research*, vol. 35, pp. 2963-2988, 1997.

[56] C. J. Atman, J. R. Chimka, K. M. Bursic and H. L. Nachtmann, “A comparison of freshman and senior engineering design processes,” *Design Studies*, vol. 20, no. 2, pp. 131-152, 1999.

[57] M. Blanke, R. I. Zamanabadi, S. A. Bogh and C. P. Lunau, “Fault-tolerant control systems - a holistic view,” *Control Engineering Practice,* vol. 5, no. 5, pp. 693-702, 1997.

[58] C. Baydar and K. Saitou, “Off-line error prediction, diagnosis and recovery using virtual assembly systems,” *Journal of Intelligent Manufacturing,* vol. 15, no. 5, pp. 679-692, 2004.

APPENDIX 1

EXPERIMENTAL DATA

The abbreviations used in the following tables are:

A1- ARCHITECTURE-1
A2- ARCHITECTURE-2
A3- ARCHITECTURE-3
1- FAILURE TO LOWER ARM (FAILURE-1)
2- FAILURE TO PICK PART (FAILURE-2)
3- FAILURE TO CLOSE GRIPPER (FAILURE-3)
4- INSERTION FAILURE/ SCRATCH (FAILURE-4)
E- ENGINEER OPERATOR
N- NOVICE OPERATOR
F- FAILURE
O- OPERATOR
T- TIME TAKEN TO DIAGNOSE THE FAILURE (SECONDS)
IS- NUMBER OF INFORMATION SOURCES CONSULTED BY THE TROUBLESHOOTER
Q- NUMBER OF QUESTIONS ASKED BY THE TROUBLESHOOTER
DT- NUMBER OF DIAGNOSTIC TREATMENTS PERFORMED BY THE OPERATOR
R- NUMBER OF RUNS (CYCLES) OF THE SYSTEM
Re.- RELIABILITY (CLOSENESS TO ONSITE MAINTENANCE) (1-LOW, 10-HIGH)
Q.Op- QUALITY OF TREATMENTS PERFORMED BY THE OPERATOR (1-LOW, 10-HIGH)
Acc.- EASE OF REMOTE ACCESS OF THE INFORMATION NECESSARY FOR DIAGNOSIS AND TREATMENT (1-LOW, 10-HIGH)

Obj.- OBJECTIVENESS OF THE INFORMATION AVAILABLE-UNDERSTANDABLE AND EXPLAINABLE (1-LOW, 10-HIGH)

Cog.- ARCHITECTURE FACILITATES COGNITIVE REASONING WITHOUT TOO MANY MANIPULATIONS (1-LOW, 10-HIGH)

R.Sk.- REQUIREMENT OF A SKILLED OPERATOR (1-HIGH, 10-LOW)

Table A1: Architecture-1 with expert troubleshooter

Expert	F.O	T	IS	Q	DT	R	Re.	Q.Op	Acc.	Obj.	Cog.	R.Sk.
1	4E	2236	44	14	5	15	3	3	3	4	4	5
2	1E	760*	3	3	3	2	5	8	2	8	2	10
3	3N	2478	23	18	12	7	3	7	4	9	5	3
4	2E	1492	25	10	4	5	6	2	3	8	5	9
5	4N	1720	21	16	4	5	3	8	3	3	3	8
6	2N	1712	33	14	6	3	3	2	4	3	2	8
7	3E	1020	19	3	6	3	5	6	5	8	7	5
8	1N	1095	24	10	4	1	1	1	1	1	1	10

Table A2: Architecture-2 with expert troubleshooter

Expert	F.O	T	IS	Q	DT	R	Re.	Q.Op	Acc.	Obj.	Cog.	R.Sk.
1	2E	480	15	2	1	4	5	4	7	6	6	5
2	4N	2790	34	15	4	25	2	5	3	5	6	7
3	1N	1254	14	7	4	5	8	7	9	9	9	3
4	3E	1235	19	9	5	9	7	3	7	8	8	7
5	2N	1425	34	5	2	2	5	5	5	5	5	5
6	3N	665	9	1	2	3	7	6	8	7	5	5
7	1E	945	35	2	2	4	7	5	8	8	7	6
8	4E	1445	3	33	1	5	5	5	5	5	5	5

Table A3: Architecture-3 with expert troubleshooter

Expert	F.O	T	IS	Q	DT	R	Re.	Q.Op	Acc.	Obj.	Cog.	R.Sk.
1	3E	705	39	3	1	1	7	6	7	7	8	5
2	2N	1110	46	5	3	1	8	2	10	8	8	5
3	4E	2844	33	10	8	25	9	2	7	7	7	10
4	1E	1020	13	2	4	4	9	9	9	9	10	2
5	3N	694	10	9	2	5	8	3	8	8	8	3
6	1N	1095	33	1	3	2	8	7	10	9	7	3
7	4N	2340	22	9	6	20	10	3	7	8	5	8
8	2E	580	7	8	2	1	8	10	10	8	10	2

Table A4: Architecture-1 with novice troubleshooter

Student	F.O	T	IS	Q	DT	R	Re.	Q.Op	Acc.	Obj.	Cog.	R.Sk.
1	2N	1360	28	9	2	2	5	5	5	5	7	6
2	3E	1180	36	8	6	4	7	3	9	9	8	9
3	1N	943	28	7	4	3	5	2	6	8	8	7
4	4N	2426	31	20	4	20	7	8	7	3	1	2
5	2E	1094	95	15	2	3	4	4	4	5	4	10
6	1E	1525	52	18	2	4	2	7	6	8	8	2
7	3N	465	19	1	4	1	3	8	5	3	5	8
8	4E	1605	37	5	3	15	3	9	3	4	2	9

Table A5: Architecture-2 with novice troubleshooter

Student	F.O	T	IS	Q	DT	R	Re.	Q.Op	Acc.	Obj.	Cog.	R.Sk.
1	3N	1311	49	4	3	5	7	6	8	7	8	7
2	1E	846	18	12	4	2	9	6	7	8	8	7
3	4E	1850	42	8	6	20	7	4	4	7	4	5
4	2N	920	23	2	4	1	7	2	9	8	10	9
5	1N	1303	55	10	6	5	7	7	7	7	8	7
6	3E	980	17	10	4	5	3	2	8	8	9	5
7	4N	1437	40	8	4	20	7	6	7	9	9	5
8	2E	1055	30	7	3	2	6	5	7	8	7	5

Table A6: Architecture-3 with novice troubleshooter

Student	F.O	T	IS	Q	DT	R	Re.	Q.Op	Acc.	Obj.	Cog.	R.Sk.
1	1N	2292	106	9	4	8	8	8	7	8	5	2
2	4N	2615	26	23	6	20	9	10	6	7	8	5
3	2E	1020	37	10	3	4	9	6	2	5	2	3
4	3N	987	30	3	5	4	7	2	9	10	10	9
5	3E	830	37	9	6	4	9	9	9	8	8	5
6	4E	2130	28	22	2	15	8	1	6	2	7	9
7	2N	1150	36	5	1	4	10	3	7	6	5	2
8	1E	580	12	3	3	2	9	3	10	10	9	3

APPENDIX 2

RESULTS OF THE SURVEY

SURVEY RESULTS: EXPERTS

Educational qualifications of the experts

Expert	Name	Company	Qualification
1	Andy Cannon	Automation Tool Company	MS in electrical engineering
2	Greg Michel	Wright Industries	BS in elec engg tech
3	Jim Johnson	Automation Tool Company	BS in electrical engineering, TTU 1994
4	Raymond Baca	Matrix Technologies	BS in Electrical Engineering
5	Rick Kolb	Motoman	controls engineer
6	Jeff Svetlik	Testengeer	BS in ENTC-Manufacturing
7	Kurt Oswalt	Motoman	BS Electronic Engineering
8	Sean Flanagan	Mikron	B.S. Electrical Engineering

Summary of experiments performed

Expert	Architecture-1	Architecture-2	Architecture-3
1	4E	2E	3E
2	1E	4N	2N
3	3N	1N	4E
4	2E	3E	1E
5	4N	2N	3N
6	1N	4E	2E
7	3E	1E	4N
8	2N	3N	1N

Survey done after experiment 1:

Question 1: What is your opinion of the failure diagnosed?

Very easy to diagnose, easy to diagnose, neither easy nor difficult to diagnose, difficult to diagnose, very difficult to diagnose

Expert	Response	Comments
1	Difficult to diagnose	Issues with marginal equipment design and/or marginal product components are always difficult to diagnose. Remote diagnosis is more difficult because of video latency and an inability to touch/feel the parts.
2	very easy to diagnose	
3	difficult to diagnose	
4	easy to diagnose	With the help of an operator present
5	neither easy nor difficult to diagnose	
6	difficult to diagnose	
7	neither easy nor difficult to diagnose	Easy because of seeing what happened but difficult because of not knowing the system and how it has been built
8	easy to diagnose	Program is very poorly written and poorly documented

Question 2: How did you go about solving the problem? Did you use any strategy or method?

Expert	Response
1	Eliminate simple control malfunctions first. Identify key inputs and outputs for the area of interest and determine if the system is receiving inputs and setting outputs appropriately. Follow the lead of the operator toward a solution. Operators, even novice operators, have an intuition about the equipment they operate. Listening to their descriptions will lead you to the solution.
2	I visually saw the problem on the webcam.
3	Check the program for the inputs that cause the valve to work. Checked the inputs and found they were OK. Then checked the outputs. Checked the mechanical operation, then traced the operation from the gripper back to the PLC.
4	Understanding the sequence of events, having a quality video of the problem, having an operator available and an offline copy of the program pointed me in the right direction for troubleshooting
5	Look at what can effect the position of the peg in the puck. Robot arm positioning, puck positioning, and then feeder positioning.
6	Cross reference the I/O list with the offline program to determine possible causes of the failure. I had the operator verify whether the sensors were showing the proper state.
7	First Air source, Second Program operation, third wiring
8	Talk to operator about what operation did not complete. Analyze program to figure out trigger for machine movement. Ask operator to check sensor

Question 3: Is there anything that would have made the problem easier to solve?

Expert	Response
1	Outside of being there in person, no. The remote tools are very good. If I had more tools such as online diagnostics of the PLC it would most likely have taken longer to diagnose this problem because I would have spent some time attempting to watch the sequence operate from with the PLC interface.
2	Probably if I had watched the process a few more times before trying to diagnose.
3	More indicators on the valve/ solenoid. Pictures of the PLC.
4	online with the plc program
5	
6	Being online with the program.
7	better knowledge of system operation
8	Better program with descriptions in code

Survey done after experiment 2

Question 1: What is your opinion of the failure diagnosed?

Very easy to diagnose, easy to diagnose, neither easy nor difficult to diagnose, difficult to diagnose, very difficult to diagnose

Expert	Response	Comments
1	easy to diagnose	It was a physical problem that could be diagnosed with physical feedback to the system.
2	very difficult to diagnose	I was expecting a controls/elect. problem, but was a mechanical issue.
3	easy to diagnose	
4	easy to diagnose	It was easier to diagnose this problem than in experiment 1. It was still difficult, because I am not very familiar with the system.
5	neither easy nor difficult to diagnose	
6	very easy to diagnose	
7	neither easy nor difficult to diagnose	A little more difficult because I would expect the arm not to continue to travel and release part if the advanced prox did not make.
8	difficult to diagnose	Problem with mechanical design of system

Question 2: How did you go about solving the problem? Did you use any strategy or method?

Expert	Response
1	Once the problem occurred I instructed the operator to run the trial again and to leave the system in the malfunctioning state once the problem re-occurred. This allowed me to monitor the system in a steady state of incorrect operation and detect the problem.
2	The only way to diagnose the problem was to see it happen on the webcam.
3	Used the PLC program to determine the inputs needed and find an input not coming on- verified with the voltage waveform. Then discussed with operator about checking the sensor.
4	Was able to go online with the program and verify that the output to gripper #2 was being energized. The waveform indicated that no voltage was getting to the gripper #2 solenoid, therefore a problem existed between the output card and the solenoid. (wiring problem)
5	Use sequence of operations and PLC program to guide operator through problem
6	Viewed the problem with the live webcam. Found logic that controls the output. Watched logic as the operator ran machine. Used the waveforms to show that the output voltage was not coming on when the output was activated.
7	Watching the failure a couple of times. Viewing PLC code to understand the travel sequence. Use of Waveform to view status of input on the extend.
8	Determine how system works. Then eliminate the possibilities until the only thing left was a mechanical problem.

Question 3: Is there anything that would have made the problem easier to solve?

Expert	Response
1	If the operator had not cleared the station after the first malfunction we could have isolated the problem quicker without needing to re-create the malfunction, but it would not have been easier.
2	Being there.
3	not from remote (that I can think of)
4	No
5	
6	No, very easy to diagnose.
7	Sequence of operation needs to stop during a motion failure and not continue to release part.
8	no

Survey done after experiment 3:

Question 1: What is your opinion of the failure diagnosed?

Very easy to diagnose, easy to diagnose, neither easy nor difficult to diagnose, difficult to diagnose, very difficult to diagnose

Expert	Response	Comments
1	neither easy nor difficult to diagnose	The symptom was physical and related to measured inputs/outputs
2	easy to diagnose	
3	very difficult to diagnose	
4	very easy to diagnose	
5	neither easy nor difficult to diagnose	
6	difficult to diagnose	
7	difficult to diagnose	The Stage Diagram was helpful to eliminate a controls programs issue quickly. The difficulty was with the parts and feed problem. Operator was a tremendous help with solving issue.
8	easy to diagnose	

Question 2: How did you go about solving the problem? Did you use any strategy or method?

Expert	Response
1	The operator indicated a failure of the gripper to close on Robot 2. Operator transmitted sequence record. An investigation of the measured vs programmed output voltage to the solenoid showed the PLC had logically set the output high but that Labview had not recorded an actual rise in voltage. This indicated a disconnected circuit which the operator located and repaired.
2	The sequence skipped a step and I thought of what should have triggered the skipped step and assumed that was the issue which it was.
3	Checked the outputs and inputs and looked at the running of the system. Discussed the issue with the engineer
4	Stage diagram pointed me to the the problem right away, the I/O diagram allowed me to determine which I/O was not coming on. The historical data plot showed me that nether the field or plc voltage was not being made, therefore it was a sensor failure
5	Use sequence of ops and with troubleshooting aids.
6	I used the stage diagram to determine where the failure occurred. From there I attempted to use the excel file showing states of the inputs/outputs, but the graphs were misleading. I ended up using ladder logic to determine the point of failure.
7	Talking with operator and discussing process and parts. Use of Stage diagram to view process and look for possible errors. Biggest strategy was the use of the operator for eyes on process.
8	Used the "stage" diagram to determine what steps did not complete. Was able to use the I/O monitor in the stage diagram to see that the sensor was not working.

Question 3: Is there anything that would have made the problem easier to solve?

Expert	Response
1	No.
2	Being there.
3	I should have asked if the system ever ran.
4	If the I/O input turned red when failed
5	
6	Difficulties with communications and use of tools caused some unexpected problems.
7	No, this kind of mechanical problem is difficult to diagnose since there is not feedback on part present on track.
8	In the state diagram, show the inputs required to progress to the next step and the ouputs that are executed at that step. In the code, do not progress to the next step until all input conditions have been met.

Final survey

Question 1: Of the failures diagnosed which one do you think was easier to solve and why?

Expert	Response
1	The Gripper not closing. Once the signals were evaluated it was obvious the PLC logically turned on the close output but the voltage was not received at the solenoid.
2	Experiment #1 was easiest, because I could visually see what happened and infer the problem.
3	Experiment #2 was easiest, because it was a problem with plenty of tools.
4	#3. Stage diagram and historical data plots are very intuitive diagnostic tools
5	Not Clamping part. Sequence is repeatable.
6	The 2nd failure. The failure was more evident and easier to detect. The tools were improved for the last failure, but the failure was simply more difficult to diagnose.
7	Item #2
8	Third one. The tools made it easier, even with a program that made machine behave strangely.

Question 2: Which architecture is better and why?

Expert	Response
1	Architecture 2. I suspect the cost of Architecture 2 is significantly lower than that of #3 but it included most of the tools necessary to diagnose most failures.
2	#3, because it provides the most tools.
3	The third is best in most cases in case you need the tools, but have to know the est tool to use. Being able to know what is going on helps and it has the most explanation of the real world.
4	#3. More tools available.
5	The architecture that allow the troubleshooter to have the most relevant info. IN each situation, having more info does not hurt, so the last one would be best.
6	The 3rd architecture is better because you have more tools to choose from, however, I feel some of them are over-kill and not necessary.
7	I really liked the stage diagnostic. I feel that is a good bridge between a Engineer and tech
8	Third one. Enough information was given.

Question 3: What additional features or diagnostic tools would you suggest for improving the remote troubleshooting performance?

Expert	Response
1	In a real situation more webcams would be helpful. These are fairly low cost additions and would give the troubleshooter multiple views. A wireless headset for operator would allow the troubleshooter to stay in constant contact with the operator; this would be similar to being on-site.
2	A better cross-reference between I/O sheet and naming and broken out by station. Preferably a hard copy.
3	Knowing more about the history of the machine….how it has performed in the past. This is probably more a discuss with the operator.
4	More webcams. One for each station. Better quality of webcams (resolution and less choppiness)
5	
6	The online PLC program, video camera, a form of communication to the operator, and waveforms are all the tools necessary to diagnose most failures remotely. The waveforms are a nice tool and should be used in industry in my opinion.
7	The biggest problem I had was with any software that uses symbols for I/O descriptions. I should be able to take cursor over a symbol and the address or full description should show. The use of a diagram showing the actual equipment helps a lot.
8	If the stage diagram showed what the prerequisites were for each step and the actions that are taken, the operator probably could diagnose the machine themselves without requiring the remote support.

Question 4: Would a replica of the robot/ the production line be of use in the diagnosis of the failures considered?

Expert	Response
1	Maybe, but it could be costly. A solid model in Solidworks for viewing to get familiar with the equipment.
2	Yes
3	Most likely not unless it is purely a design issue that cannot be solved with collaboration or webcam.
4	Not really needed for diagnostics as long as the process is explained thoroughly.
5	No. Descriptions of system were relevant.
6	No
7	yes, pictures / illustration go a long way. I have written a lot of manuals for systems and I have found that pictures and brief explanation go a long way.
8	No. The information given was sufficient.

Question 5: Are there some failures, the diagnosis for which, the replica would be useful? Are there any conditions that would warrant the use of the replica?

Expert	Response
1	Not really.
2	You could go offline and use the replica to discuss the failure with other minds and then remote again with the ideas developed.
3	Design issues that are very complex and not easy to see or explain.
4	No
5	Not that I can think of
6	No, the failure would not exist on the replica, therefore it would be pointless. The engineer should be familiar enough with the equipment to use the other tools to diagnose the problem without a replica.
7	The use of the stage diagram
8	Yes, but only on very complex systems that take hours to troubleshoot.

Question regarding ranking of architectures: For the failures you diagnosed, give your overall ranking of the three architectures. In other words, your assessment of which architecture is more suitable to diagnose a type of failure. Ranking: 1-most suitable, 2-is in the middle, 3-least suitable.

Expert	Failure diagnosed	Architecture-1	Architecture-2	Architecture-3
1	Failure to pick part	2	1	1
	Failure to close gripper	2	1	1
	Insertion failure/ scratch	2	3	3
	Comments: On the insertion failure, more remote information would have been a distraction because the real information needed to come from the feedback of the operator. The failure was not related to a monitored or measured input to the system.			
2	Failure to lower arm	3	2	1
	Failure to pick part	3	2	1
	Insertion failure/ scratch	3	2	1
	Comments: Operator interaction seemed to be the most important tool used.			
3	Failure to lower arm	3	2	1
	Failure to close gripper	3	1	2
	Insertion failure/ scratch	3	2	1
4	Failure to lower arm	2	1	1
	Failure to pick part	3	2	1
	Failure to close gripper	3	2	1
5	Failure to pick part	3	1	2
	Failure to close gripper	3	2	1
	Insertion failure/ scratch	1	2	3
6	Failure to pick part	3	2	1
	Failure to close gripper	3	2	1
	Failure to lower arm	3	2	1
7	Failure to lower arm	3	2	1
	Failure to close gripper	3	1	2
	Insertion failure/ scratch	1	2	3
8	Failure to lower arm	3	2	1
	Failure to pick part	3	2	1
	Insertion failure/ scratch	1	2	3
	Comments: For the insertion failure/scratch the additional functionality provided by Architecture 2 and 3 would not have helped			

Question regarding ranking of attributes: On a scale of 1 to 5, rank the following attributes in terms of their importance in evaluating troubleshooting performance. 1-most important, 5-least important.

Parameter	1	2	3	4	5	6	7	8
Time taken to diagnose the failure	2	2	2	2	3	2	1	1
Amount of searching	2	5	4	2	3	3	3	3
Number of diagnostic tests performed	2	2	3	3	3	4	2	4
Number of incorrect diagnoses	3	4	1	4	1	1	2	2
Quality of architecture	1	1	5	1	1	5	1	1

Comments: Expert 1 The operator and the architecture tools are the eyes, ears, and hands of the remote troubleshooter. The quicker they can get into the problem the easier it is to solve. The tools should be intuitive, the ones used in the experiment give the troubleshooter the pertinent information without too much extra distraction.

Question regarding ranking of sub-attributes: On a scale of 1 to 5, rank the following sub attributes under amount of searching in terms of their importance in evaluating troubleshooting performance. 1-most important, 5-least important.

Parameter	1	2	3	4	5	6	7	8
Number of information screens	5	5	4	2	3	4	3	5
Number of views of the system	3	3	1	1	5	3	2	4
Number of questions asked by the expert	4	1	3	4	4	2	2	3
Number of questions answered correctly by the operator	1	2	2	3	2	2	1	2
Number of treatments performed without operator involvement	2	4	5	5	1	1	3	1

Comments: Expert1: Accurate responses from the on-site (local) person are very important, incorrect answers can be misleading and cause additional downtime. Problems related to measured signals (I/O) can be easily diagnosed from computer resources and this may be easier than consulting the operator, but it is important during troubleshooting to keep your operator engaged in the process so their intuition is available. If a troubleshooter leaves the operator in the dark they will not feel a part of the process and may not be as helpful when called upon.

Question regarding ranking of sub-attributes: On a scale of 1 to 7, rank the following sub attributes under quality of architecture in terms of their importance in evaluating troubleshooting performance. 1-most important, 7-least important.

Parameter	1	2	3	4	5	6	7	8
Reliability (how close to on site maintenance)	5	7	4	2	3	1	2	2
Quality of treatments performed by the operator	7	4	3	5	4	5	4	7
Accessibility (ease of remote access) of the information necessary for diagnosis, treatment	3	5	6	3	7	2	2	3
Objectiveness of the information (explainable and understandable)	4	2	7	7	5	3	2	1
Architecture facilitates cognitive reasoning without too many manipulations	1	6	5	1	1	4	2	5
Requirement of a skilled operator	6	1	1	6	6	7	5	4
Quality of guidance given to the operator (Graphic guidance and advice given to the operator on site)	2	3	2	4	2	6	2	6

Comments: Expert 1: This was a difficult ranking. All of these items are important to successfully correcting a problem from a remote location.

SURVEY RESULTS: STUDENTS

Educational qualification

Student	Name	Qualification
1	Matt Svotek	I have taken an introductory PLC course
2	Danny	I am a Texas A&M student who took the course ENTC 410. This course goes over automation and robots. Throughout the course we worked with programmable logic controllers and how to use them.
3	Robert Fleming	Associates in Science, BS in Engineering in progress
4	Sam Lutz	Student in Manufacturing and Mechanical Engineering Technology. I completed ENTC 410 in the Fall 2009 semester.
5	Daniel Turrubriartes	Education in egnineering technology
6	Juan	I am a senior MMET student.
7	Aron Carpenter	Senior in Engeineering Technology. Coursework in industrial automation.
8	Shawn McCoy	Engineering Technology student, completed ENTC380, 383, 410.

Summary of experiments performed

Student	Architecture-1	Architecture-2	Architecture-3
1	2N	3N	1N
2	3E	1E	4N
3	1N	4E	2E
4	4N	2N	3N
5	2E	1N	3E
6	1E	3E	4E
7	3N	4N	2N
8	4E	2E	1E

Survey done after experiment 1:

Question 1: What is your opinion of the failure diagnosed?

Very easy to diagnose, easy to diagnose, neither easy nor difficult to diagnose, difficult to diagnose, very difficult to diagnose

Student	Response	Comments
1	Easy to diagnose	
2	neither easy nor difficult to diagnose	It was a moderate problem given the tools at my disposal.
3	very easy to diagnose	
4	difficult to diagnose	
5	easy to diagnose	
6	difficult to diagnose	
7	Easy to diagnose	
8	neither easy nor difficult to diagnose	

Question 2: How did you go about solving the problem? Did you use any strategy or method?

Student	Response
1	I first opened the ladder diagram to try to understand how it works. I then gave up on that and thought only about the physical parts of the system. Then the problem became evident
2	I looked at the system as a whole and tried to cut it down into its major components. The first being the programming, second actually mechanical work, and the third electrical communication.
3	Identify portion of the system experiencing failure, work backwards.
4	I first started by finding which variables were fixed and then finally figured out what variable was changing.
5	I first considered looking at the logistics code. Then I looked at the physical errors
6	At first I was to preoccupied with the RSLogix code, but then I realized the problem was more physical. It took a little bit of understanding of the code to realized that the sensor needed for lowering the gripper was controlled by the x axis sensor. After that, once I knew how the sensor worked, I had the correct sensor moved to a spot where it could accurately read the position of the x axis arm.
7	I had each physical connection checked, in order, starting with those connected directly to the gripper and moving towards the connection to the PLC.
8	After determining the failure point, I checked the code for obvious problems. Then I had the webcam set to monitor the subsystem that was producing the failure. It was then possible to see the way the parts interacted. Having the operator manipulate the feed timer allowed me to see how the timing affected the output.

Question 3: Is there anything that would have made the problem easier to solve?

Student	Response
1	Nothing I can think of.
2	If I could see the actual program run on my screen so that i may follow it myself. Also, a clearer description on the programming would help.
3	technician with knowledge to operate DVOM
4	Knowing what I could and could not change in order to solve the problem. Knowing what everything was in the ladder diagram for the experiment.
5	no
6	Some former knowledge on how the hall sensors worked would have been extremely helpful.
7	being able to see the connection to the PLC.
8	Real-time monitoring of the PLC programming, smooth video link for easier diagnostics.

Survey done after experiment 2:

Question 1: What is your opinion of the failure diagnosed?

Very easy to diagnose, easy to diagnose, neither easy nor difficult to diagnose, difficult to diagnose, very difficult to diagnose

Student	Response	Comments
1	easy to diagnose	
2	easy to diagnose	
3	difficult to diagnose	
4	neither easy nor difficult to diagnose	It was easier to diagnose this problem than in experiment 1. It was still difficult, because I am not very familiar with the system.
5	difficult to diagnose	
6	neither easy nor difficult to diagnose	
7	neither easy nor difficult to diagnose	
8	easy to diagnose	Having real-time waveform data made it much easier.

Question 2: How did you go about solving the problem? Did you use any strategy?

Student	Response
1	I first watched the ladder diagram as the operator ran the program. I saw that program was energizing the output that controls the gripper, so I know the failure is not because of a sensor. I then determined the problem was between the plc and the system. From there it was easy.
2	First saw that since the system went down the first time it was receiving power and that the logic was working. So then i looked at the sensors and from the voltage wave form i saw the second hall sensor was not coming on. I then asked the operator to inspect it and he found that it was look and the problem was solved.
3	Watched the webcam, followed the ladder logic. Adjusted timing and checked results. Changed position of webcam to observe any mechanical problems
4	I started by checking the obvious, i.e. seeing if there were any mechanical problems. From there I knew that the problem had to be with the gripper not picking up the part. I checked to see if the arm moved in the Z-axis. I then moved to check the sensors. The optical sensor for the block never gained a voltage which showed that there was a problem with the block.
5	Checked voltage and then pressure of the x axis hall effect sensor
Student	Response
6	Once I discovered that there was no change in the voltage for the second gripper, I knew that it was a miscommunication between the PLC and the machine. After the operator told me that there was a loose wire by O:2/14, I knew that was the problem.
7	First, I observed what was occurring. I observed that the last part was not hitting the hole correctly. When the camera was moved I observed that it was a problem with the feeding, and that parts were feeding differently. Second, I attempted to adjust the feed time to ensure all parts fed such that they fit into the slot, but the feed distance was still changing and parts were dragging on both sides of the slot, leaving potential for parts missing the pocket. At this point, I observed that the feed distance seemed to be related to drag, and that as there were fewer parts causing drag parts fed further. After observing the system run again I saw that the parts were not feeding in a straight line. This in conjunction with the parts being made out of a high drag material led me to the conclusion that the system must be modified in order to eliminate the problem.
8	After watching the failure occur, I was able to look over the code and determine that there were two rungs of interest that controlled the arm motion. After the operator determined correct operation of the hall effect sensors, the only remaining sensor was the part ready photo sensor. Looking at the historical waveform, I was able to see that the sensor was not triggering when the part was in place. The operator was able to inspect the part and determined that the retro-reflective tape was damaged. When the operator replaced the part, the machine operated correctly.

Question 3: Is there anything that would have made the problem easier to solve?

Student	Response
1	Nothing I can think of.
2	The tools like the voltage waveforms made it easier to diagnose. No other tools come to mind at this time.
3	Multiple computer monitors.
4	If the system would have stopped as soon as it reached the problem. The system went one step further than where the failure was which forced me to check that step and then move backwards.
5	none
6	Not really, except may be the understanding of the solenoid/ actuator
7	Additional views of the line would have been helpful.
8	A smoother user interface would have been beneficial.

Survey done after experiment 3:

Question 1: What is your opinion of the failure diagnosed?

Student	Response	Comments
1	Very difficult to diagnose	
2	Very difficult to diagnose	
3	Neither Easy nor difficult to diagnose	
4	Neither Easy nor difficult to diagnose	This was easier than the first two, but the answer to fix the problem was not obvious at all.
5	Neither Easy nor difficult to diagnose	
6	Very difficult to diagnose	
7	Neither Easy nor difficult to diagnose	
8	Very easy to diagnose	

Question 2: How did you go about solving the problem? Did you use any strategy or method?

Student	Response
1	I began by looking at the ladder diagram to see if the system was running in the correct order. I spent a lot of time looking through that and didn't come to a conclusion. I then looked at the record of data for each input and output and noticed the data for the problem sensor did not follow the pattern of the same sensor in the first arm. At that point I asked the operator to examine the sensor and when he did he shifted it and the problem was fixed
2	By looking very closely at the system. First it was determined that sensors and timers were working correctly. So then we examined the parts to see if they were to spec and they were. Then by closer visual inspection of the operator he saw they were not pushing directly in the center of one another and instead moving to one side. So to fix this a timer for each part would have to be introduced.
3	I first checked the operation of all of the sensors using the labview vs VB graphs in excel. Then I checked the logic to see what inputs could be missing to arrive at this problem. I then had the technician try a different part and it worked.
4	I started by watching the stage diagram to see where the problem was. I found that it was in the 2 process. I then looked at the excel spreadsheets and found that the gripper (actual) was not lining up with the gripper (logic). Therefore the problem was with the gripper. I then looked at the voltages and saw that the gripper never changed voltages. I knew that there was not a problem with the ladder diagram so I checked the connection to the control module. The wire for the gripper was unplugged.
5	Diagnosed if the gripper was receiving voltage. Then checked connection to terminal
6	I was looking at why the first two pegs were being underfed and the last two overfed. The biggest problem though was that the last two pegs that were overfed were not going into the hole. At least when the pegs were underfed, they still made it into the hole. So that's when I figured that lowering the time would actually make the last pegs be "underfed" and that way, all the pegs would go in the hole.
7	First, I determined what operation was failing first. Second, I went to the program to determine the trigger for that process. Once I had determined what input initiated the failed output, I observed the voltage reading of that input while the problem was reproduced. This confirmed that the "Part_Ready" optical sensor input was not receiving a signal. I had the part checked for defects and it was found that the part was not reflective like the others.
8	I looked at the data provided regarding voltages and PLC status and noticed that there was no data present for one of the sensors required to lower the arm over the part. The stage diagram also showed that the sensor did not trigger when the arm was extended. I was then able to communicate with the operator to determine that the sensor was not triggering and that it had moved out of position. He was able to relocate it so that it detected the position of the arm again.

Question 3: Is there anything that would have made the problem easier to solve?

Student	Response
1	I'm sure there is - but I can not think of it right now.
2	The problem was not something that could be spotted by sensors. It is just good communication with the operator and expressing details of the system.
3	Multiple computer monitors. There is too much information available to display on 1 monitor. I always want the web cam up, but I also want a full screen view of the program logic
4	I just started looking at the data that was given and narrowed down the problem areas to the gripper. The only thing that could have made this easier would be just familiarity with the programs.
5	
6	Knowing about the wiggling problem and how that could affect the straight line path of the part.
7	The array of tools available was overwhelming. Maintaining a consistent scale on the excel output graphs would have made them more useful.
8	Not for this failure.

Final survey

Question 1: Of the failures diagnosed which one do you think was easier to solve?

Student	Response
1	The first. The simplicity of a part loaded incorrectly.
2	The one where the arm failed to lower when inserting the part. This is because since it lowers to pick up the part you know it worked mechanically and is receiving power. So the only thing left is the sensors that activate the action.
3	The arm not extending was the easiest. The basic information was all that I needed, so I did not waste time looking at alternative data sources.
4	The easiest one to solve was the third one because I had the most resources to look at. I felt like I was given enough information to actually solve the failure.
5	The third failure was the easiest to solve because of all the diagnostics tools available to solve the problem.
6	The second architecture that involved the loose wire. It was easy because it was easy to identify and the solution was even quicker.
7	Failure to grip part. Checking the basic connections of the grabber to the solenoid to the PLC was sufficient.
8	3 - The availability of data allowed me to make informed decisions and troubleshoot the problem without needing multiple runs of the machine to show me what was happening.

Question 2: Which architecture is better and why?

Student	Response
1	Architecture 2. I feel architecture 3 had so many tools I didn't know what to use. Also - the tools presented in architecture 3 are not easy to "pick up" right away.
2	Architecture 2 is better because it gives you enough tools to diagnose the problem but not so many that they become redundant.
3	It depends on the problem, architecture 1 is the easiest to use remotely, but architecture 3 does present the most information that could be necessary for a difficult problem.
4	The third is the best because it offers everything an operator could possibly need to diagnose a failure. The 1st problem is different though because I felt that it didn't really matter how much information was given because of the type of failure.
5	The third architecture is the best because of all the diagnostic tools.
6	I liked the second one better because it allowed me to use both diagnostic information from the computer, but also relied on my information about the PLC machine and its components.
7	Architecture 3 is best because it provides nearly all the information required to diagnose most problems. With this architecture, all that is required of the operator is a knowledge of the terminology for the systems. It does intimidate first time troubleshooters, but would be the best assortment of tools for quick diagnosis with an experienced troubleshooter.
8	Architecture 3 is better because it allows access to all needed information for the technician to make informed decisions without the need for continual operator interaction and without requiring repeating the failure for monitoring.

Question 3: What additional features or diagnostic tools would you suggest for improving the remote troubleshooting performance?

Student	Response
1	Being unfamiliar with the matter, I do not know of any additional tools that would help.
2	No new tools come to mind.
3	group the video and program logic together. Switching between the two was frustrating when working in real time.
4	I was happy with the third architecture.
5	none
6	It was all available to me. The only thing that would have made the experiment better was to have a better web cam. The movie clips were all choppy.
7	The tools provided are quite sufficient. Additional camera views would be helpful but may not be practical.
8	Easy access to multiple camera wiews and live motion video would be helpful.

Question 4: Would a replica of the robot/ the production line be of use in the diagnosis of the failures considered?

Student	Response
1	I dont think so. If the replica with the expert is not behaving the same way - the expert may spend extra time trying to recreate the problem rather than diagnosing it. Also - the expert has a good knowledge of the production line and everything that he would want to do can easily be done by the operator.
2	If you could compare a line working correctly to one that is not it could potentially be useful. It would give you the options to compare programming line for line which is sometimes the problem.
3	I think a map would work just as well as a replica.
4	No, because you can see the videos and live feed of the robot.
5	Yes
6	.Yes. Often I would have to consult the first assembly process to see how the second one was supposed to work.
7	This would be useful for certain repairs, but is not necessary to diagnosis. A good set of drawings would provide similar benefit and require less space and allow troubleshooters to work with a wider variety of machines.
8	Yes - it would allow the technician to observe the problems and allow them to attempt to recreate problems such as scenario 1 where the remote sensors were not aware of the program. I could have seen that the parts were stacking up in the feeder.

Question 5: Are there some failures, the diagnosis for which, the replica would be useful? Are there any conditions that would warrant the use of the replica?

Student	Response
1	I can not think of any at this time.
2	Wiring failures could be recognized. So a replica would be useful in that sense. Although, the use of a replica is more of a luxury than a necessity.
3	If it is a mechanical problem that is based on some physical limit and not a logic or electronic problem, then yes, a replica would be very useful in fault diagnosis.
4	The first problem dealing with the insertion failure.
5	None
6	The replica would be useful for diagnosing the small problems that have no measured variable. For example the wiggling of the parts for the third architecture.
7	It would be most useful in directing unskilled workers to inspect the correct part or location, but it would still be very difficult to remotely trouble shoot with an unskilled worker inspecting the machine.
8	Replicas are wonderful for failures that may not be directly sensed by the system. If these types of failures are likely, or if a system failure is extremely costly, a replica may be justified.

Question regarding ranking of architectures: For the failures you diagnosed, give your overall ranking of the three architectures. In other words, your assessment of which architecture is more suitable to diagnose a type of failure. Ranking: 1-most suitable, 2-is in the middle, 3-least suitable.

Student	Failure diagnosed	Architecture-1	Architecture-2	Architecture-3
1	Failure to lower arm	3	2	1
	Failure to close gripper	2	1	3
	Insertion failure/ scratch	3	1	2
2	Failure to lower arm	3	1	2
	Failure to close gripper	3	1	2
	Insertion failure/ scratch	2	3	1
	Comment: Sometimes just being able to communicate with the operator efficiently is the best tool.			
3	Failure to lower arm	3	2	1
	Failure to pick part	3	2	1
	Insertion failure/ scratch	1	2	3
4	Failure to pick part	3	2	1
	Failure to close gripper	3	2	1
	Insertion failure/ scratch	1	3	2
5	Failure to lower arm	2	2	2
	Failure to pick part	2	1	1
	Failure to close gripper	3	2	1
6	Failure to lower arm	1	2	3
	Failure to close gripper	3	1	3
	Insertion failure/ scratch	3	2	1
7	Failure to pick part	3	2	1
	Failure to close gripper	2	1	3
	Insertion failure/ scratch	2	1	3
8	Failure to lower arm	3	2	1
	Failure to pick part	3	2	1
	Insertion failure/ scratch	2	1	3

Question regarding ranking of attributes: On a scale of 1 to 5, rank the following attributes in terms of their importance in evaluating troubleshooting performance. 1-most important, 5-least important.

Parameter	1	2	3	4	5	6	7	8
Time taken to diagnose the failure	5	2	1	3	5	3	1	1
Amount of searching	3	3	5	4	4	3	4	2
Number of diagnostic tests performed	1	3	3	2	4	2	5	3
Number of incorrect diagnoses	4	2	2	3	3	4	2	1
Quality of architecture	2	2	2	1	5	2	3	2

Question regarding ranking of sub-attributes: On a scale of 1 to 5, rank the following sub attributes under amount of searching in terms of their importance in evaluating troubleshooting performance. 1-most important, 5-least important.

Parameter	1	2	3	4	5	6	7	8
Number of information screens	4	5	3	2	3	2	1	1
Number of views of the system	1	2	5	3	4	2	5	1
Number of questions asked by the expert	2	3	4	1	4	2	2	3
Number of questions answered correctly by the operator	3	1	2	5	4	1	3	1
Number of treatments performed without operator involvement	5	4	1	4	3	3	4	2

Question regarding ranking of sub-attributes: On a scale of 1 to 7, rank the following sub attributes under quality of architecture in terms of their importance in evaluating troubleshooting performance. 1-most important, 7-least important.

Parameter	1	2	3	4	5	6	7	8
Reliability (how close to on site maintenance)	7	2	7	7	6	2	2	2
Quality of treatments performed by the operator	5	4	4	5	6	2	6	1
Accessibility (ease of remote access) of the information necessary for diagnosis, treatment	4	6	6	4	7	3	4	2
Objectiveness of the information (explainable and understandable)	2	1	5	1	5	3	1	3
Architecture facilitates cognitive reasoning without too many manipulations	1	3	2	3	5	2	3	2
Requirement of a skilled operator	6	5	3	6	6	2	7	1
Quality of guidance given to the operator (Graphic guidance and advice given to the operator on site)	3	7	1	2	5	3	5	1

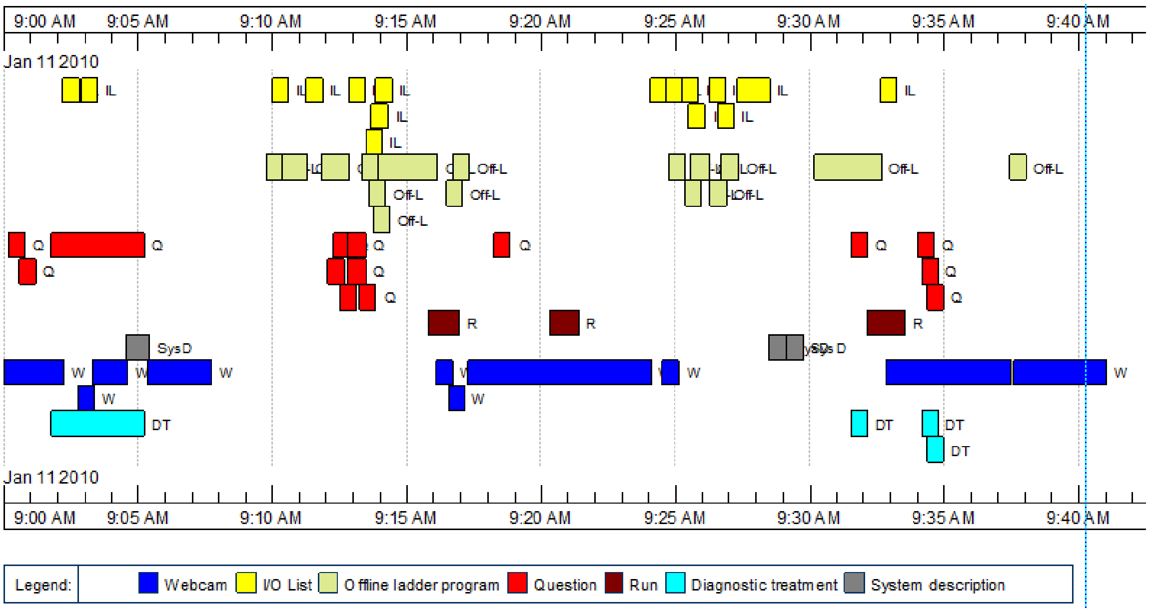
APPENDIX 3

TIMELINE PLOTS

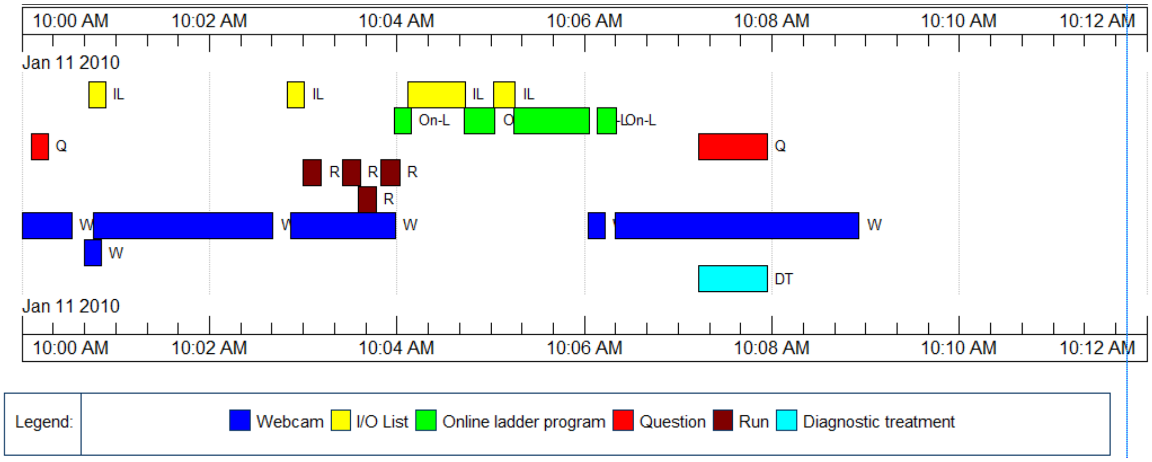
This section deals with the plots generated for the timeline analysis of the diagnosis activities of the expert and novice troubleshooters.

Expert 1:

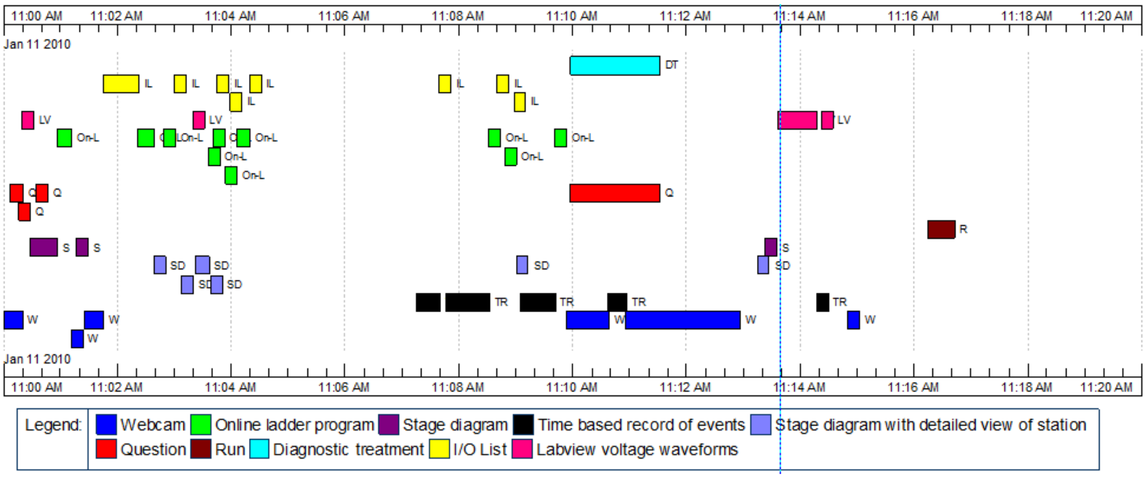
Architecture-1:



Architecture-2:

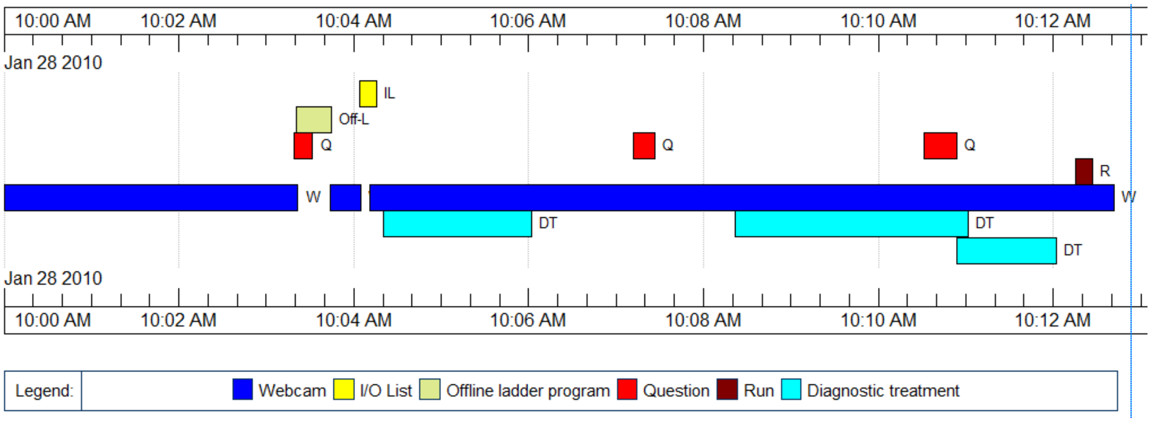


Architecture-3:

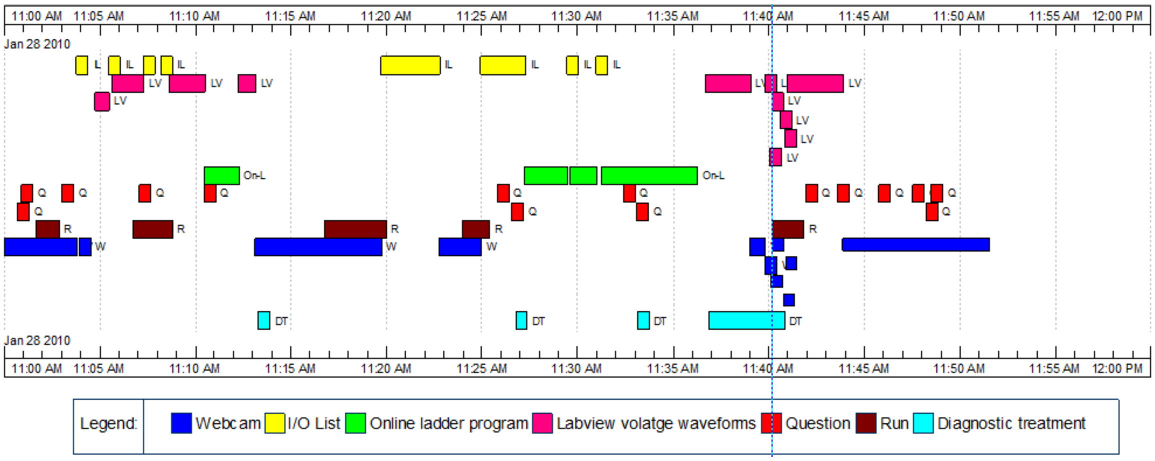


Expert 2:

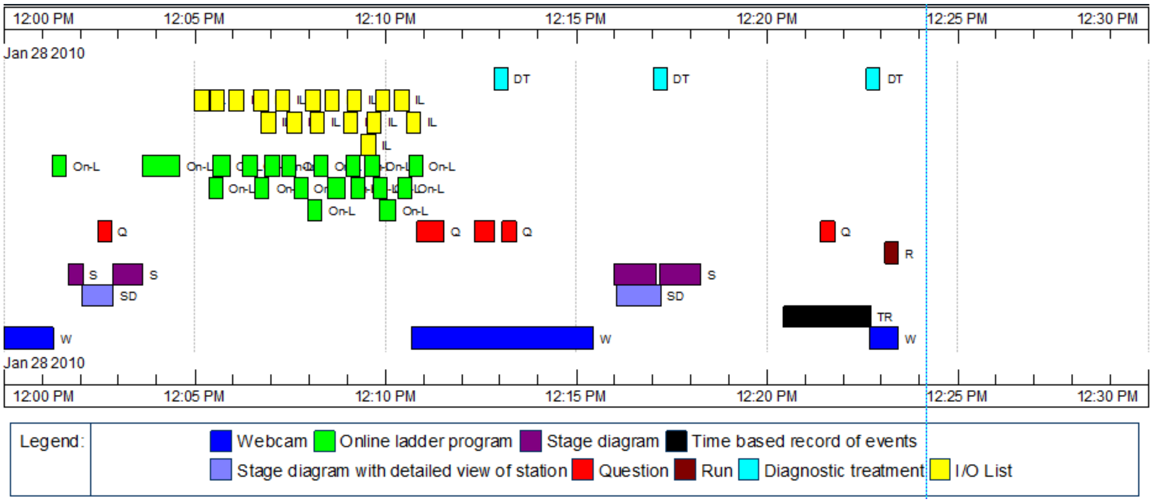
Architecture-1:



Architecture-2:

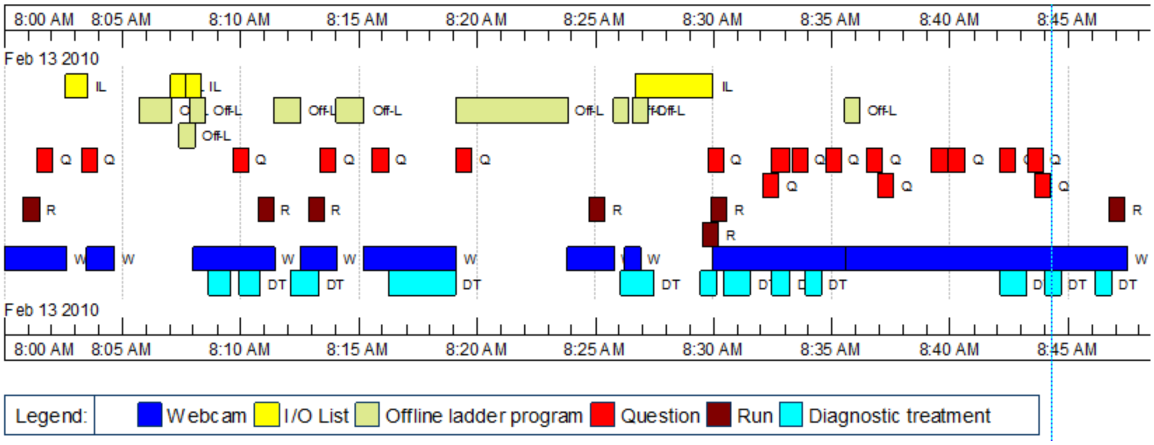


Architecture-3:

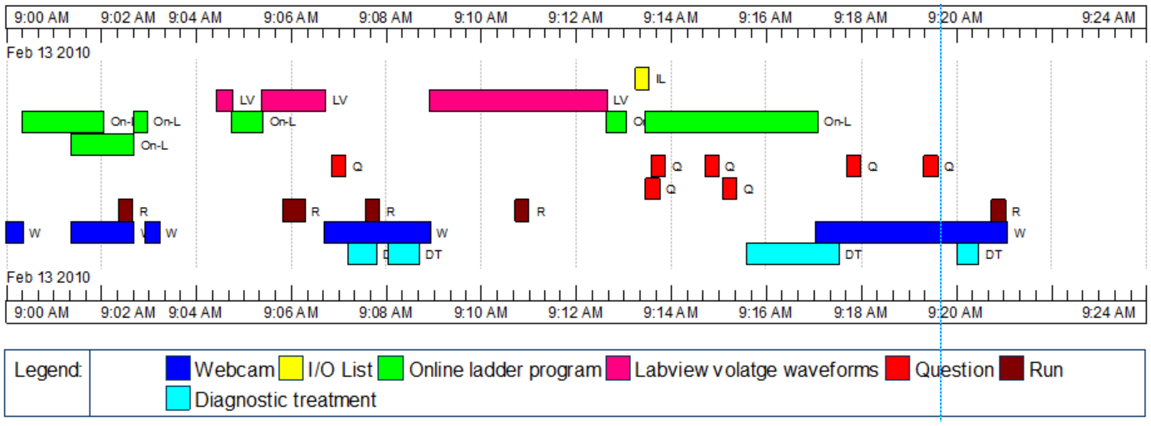


Expert 3:

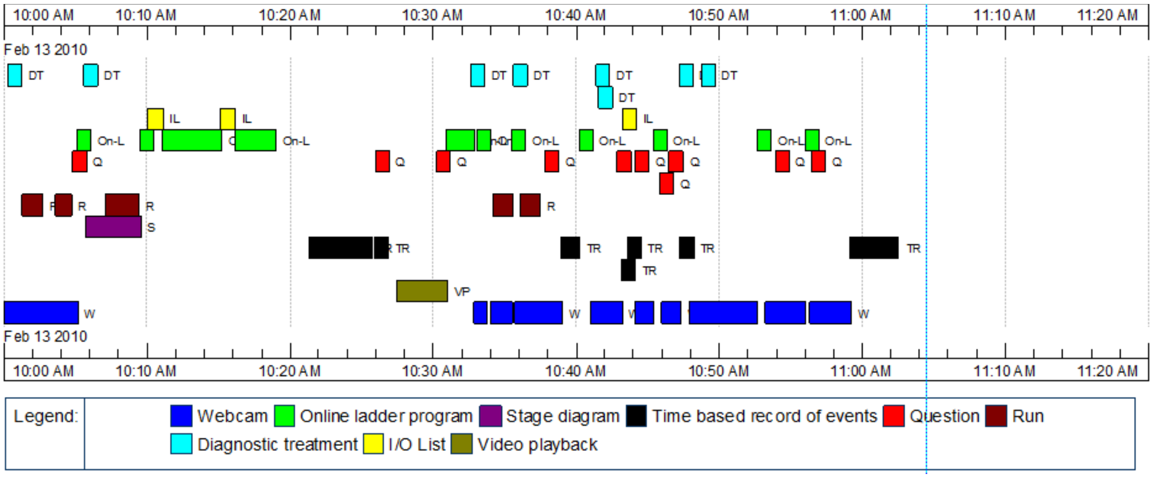
Architecture-1:



Architecture-2:

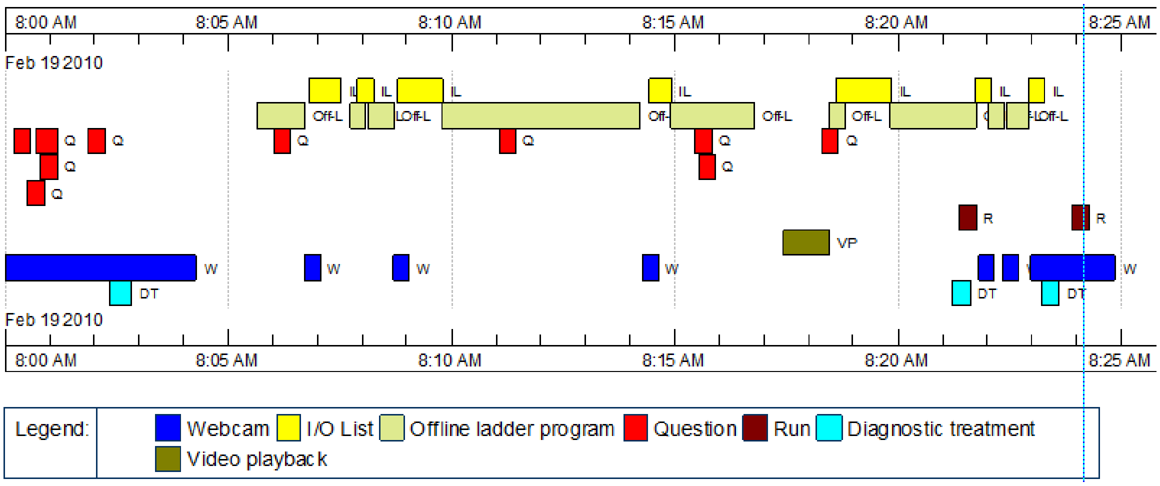


Architecture-3:

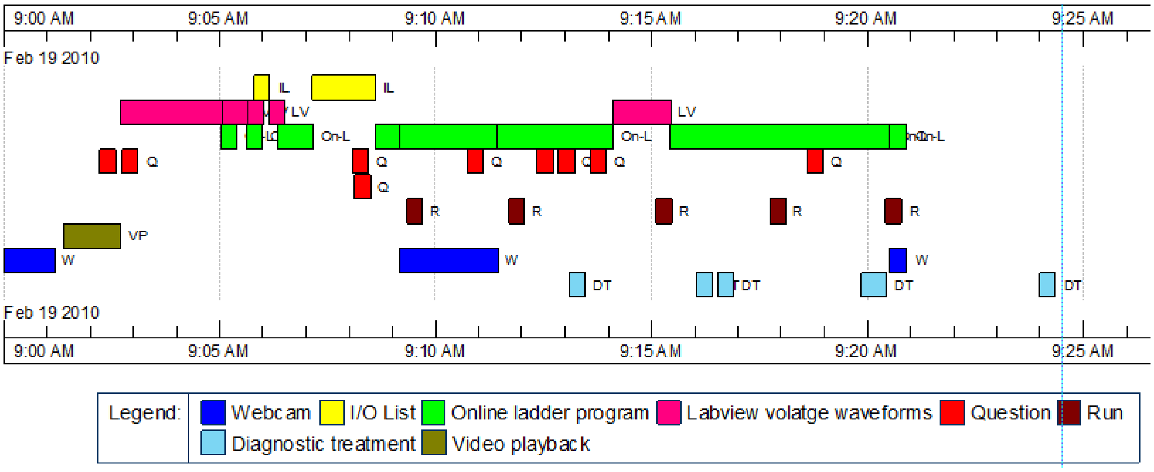


Expert 4:

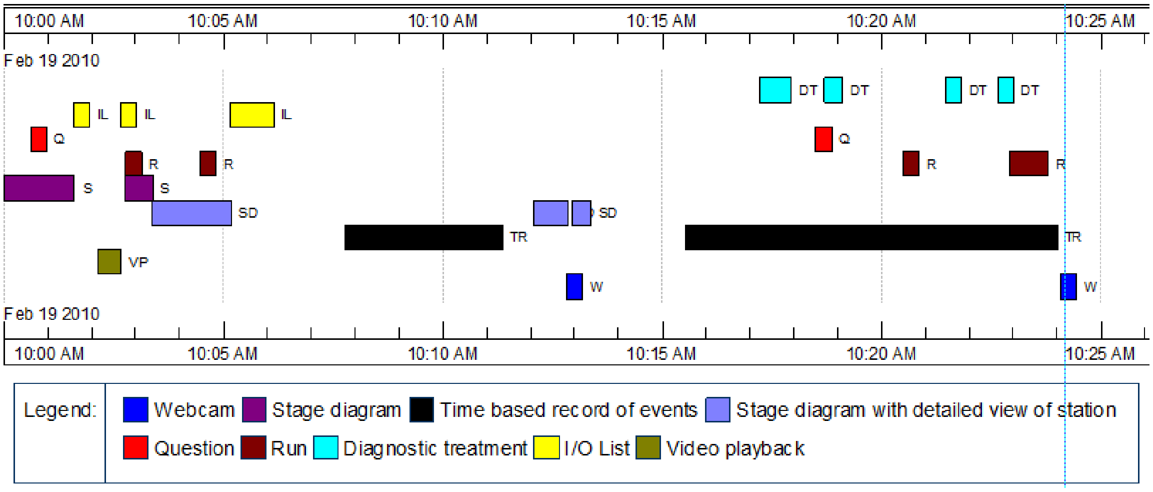
Architecture-1:



Architecture-2:

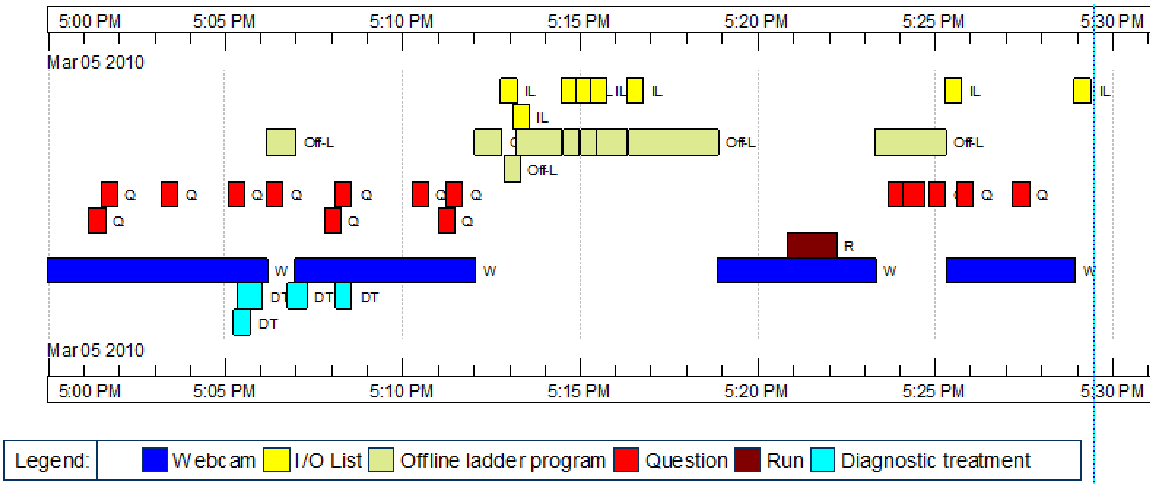


Architecture-3:

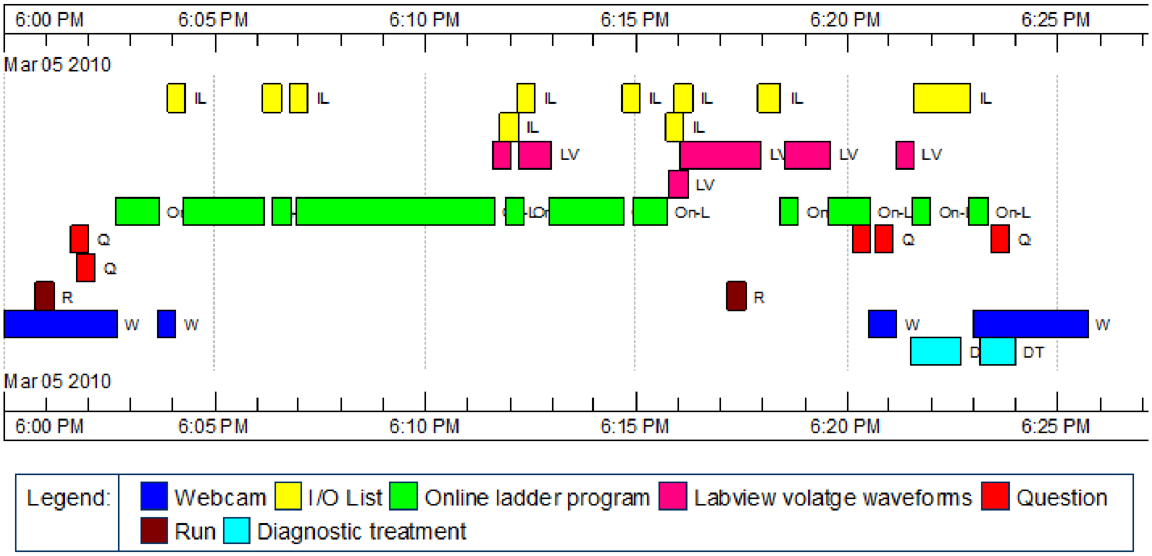


Expert 5:

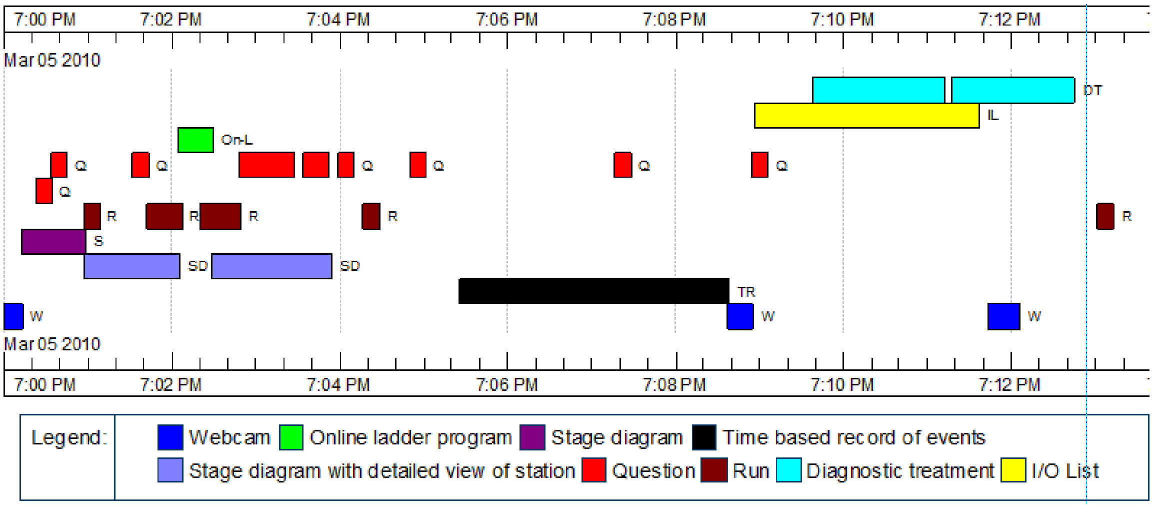
Architecture-1:



Architecture-2:

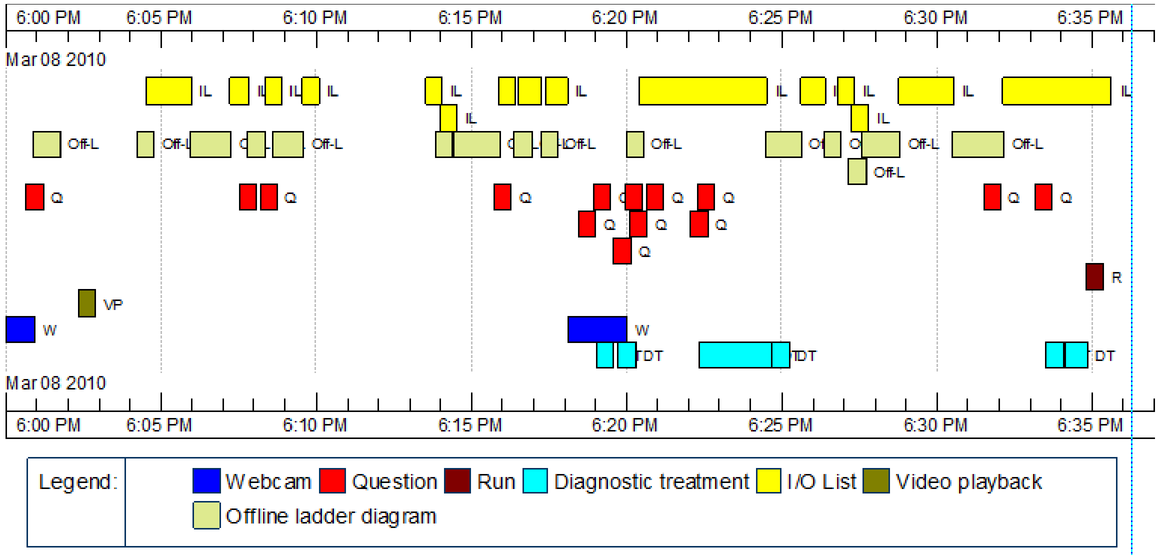


Architecture-3:

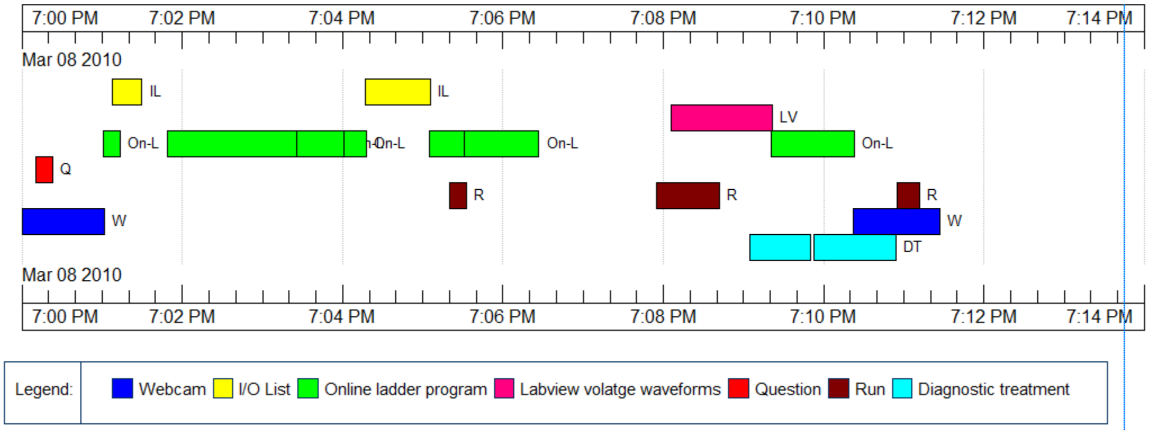


Expert 6:

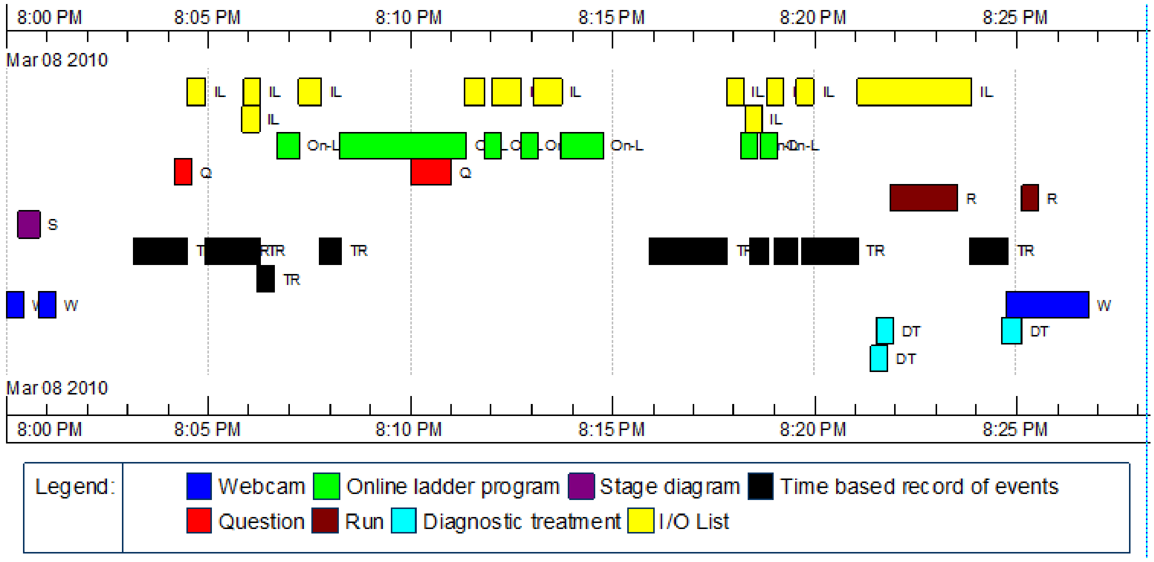
Architecture-1:



Architecture-2:

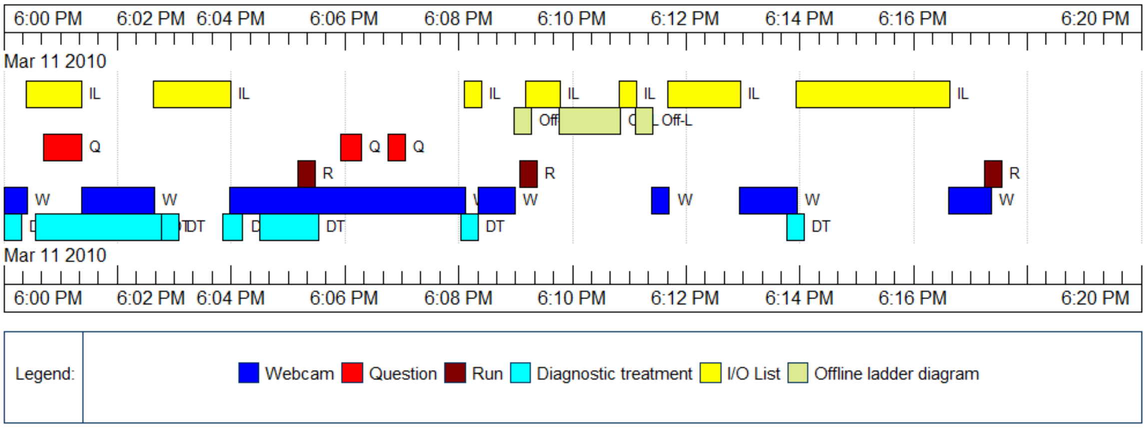


Architecture-3:

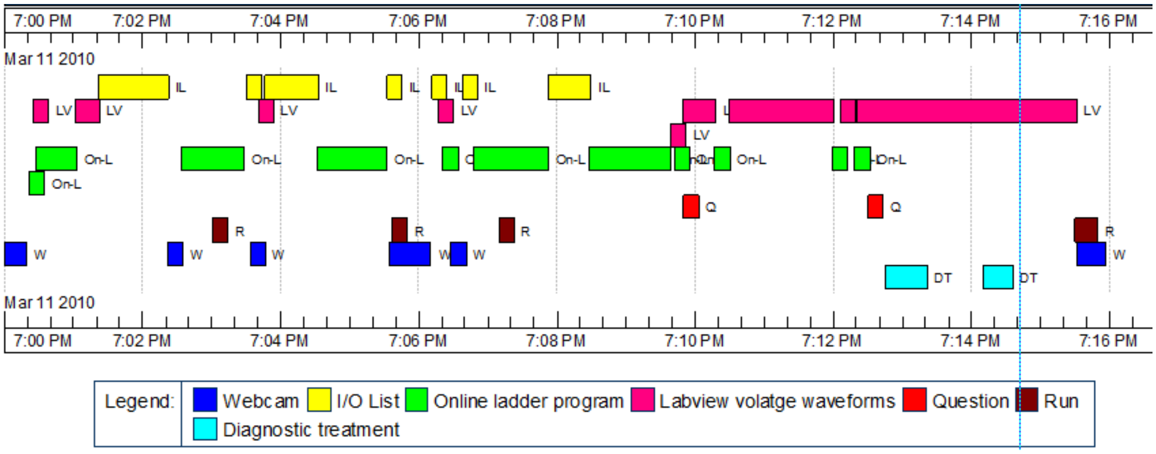


Expert 7:

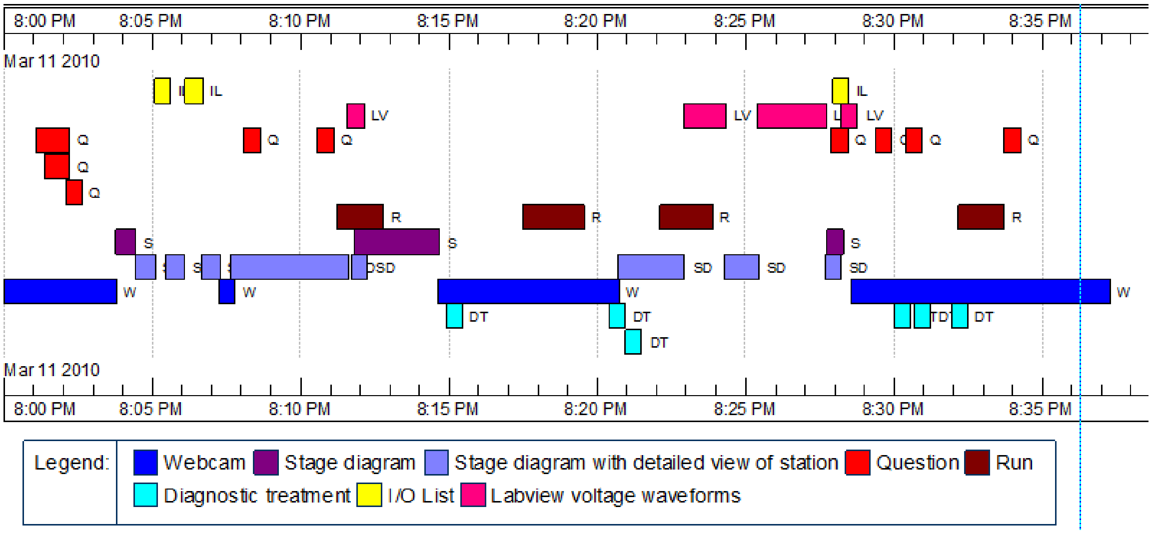
Architecture-1:



Architecture-2:

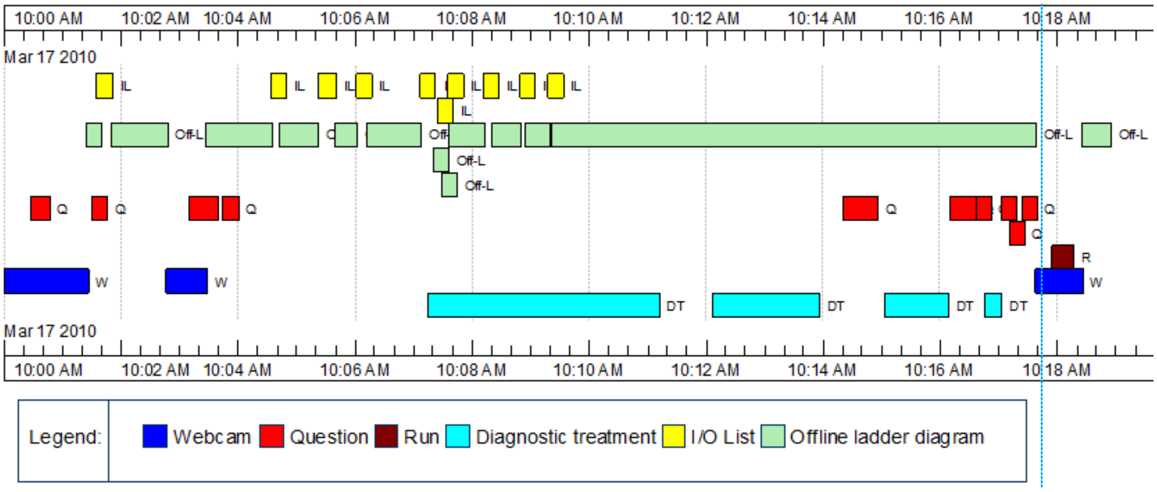


Architecture-3:

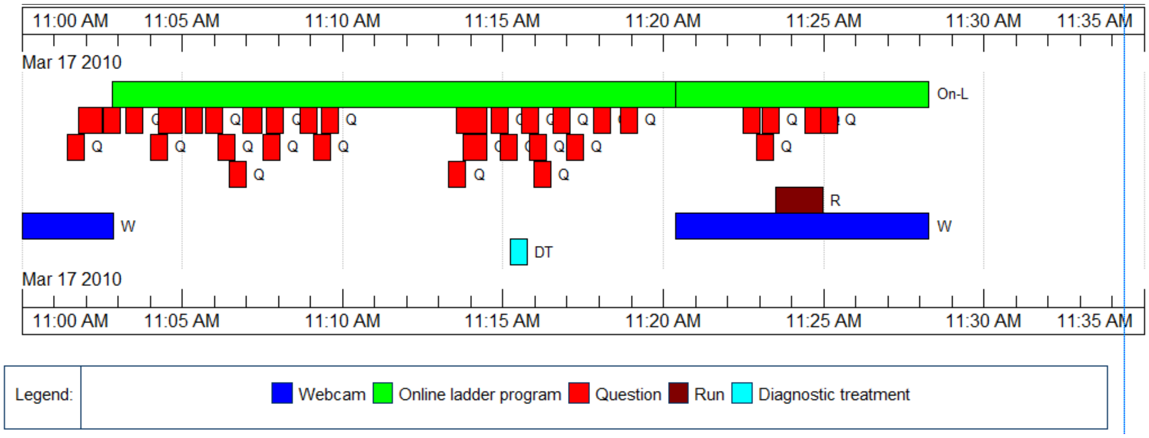


Expert 8:

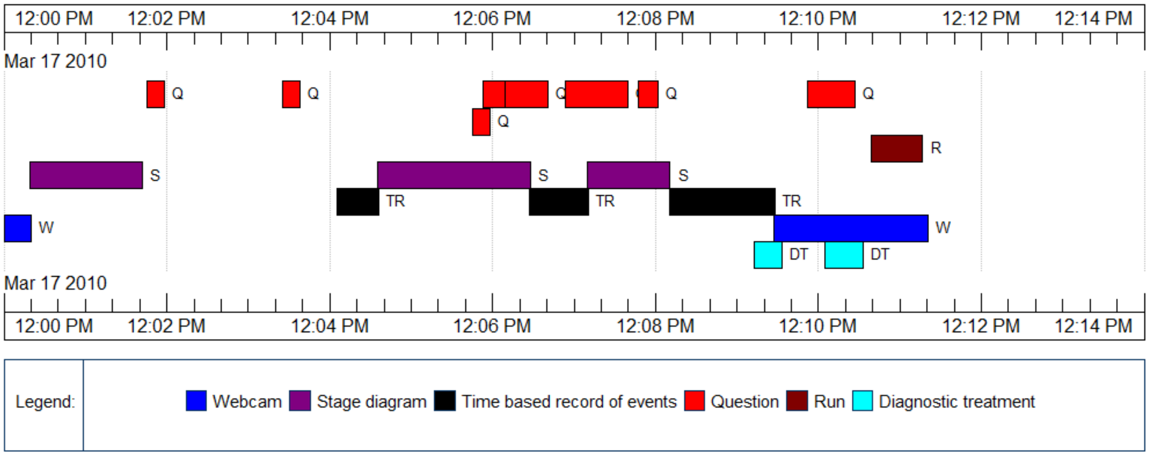
Architecture-1:



Architecture-2:

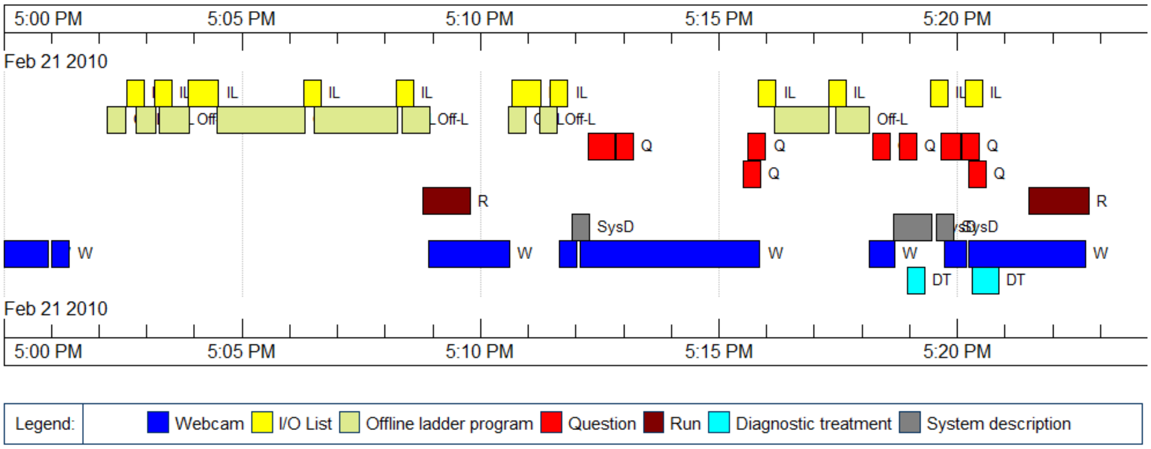


Architecture-3:

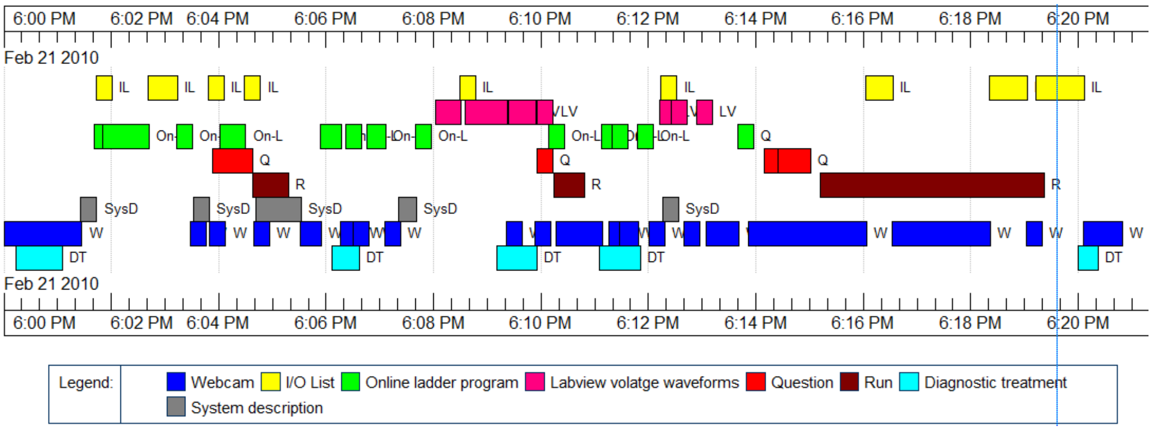


Student 1:

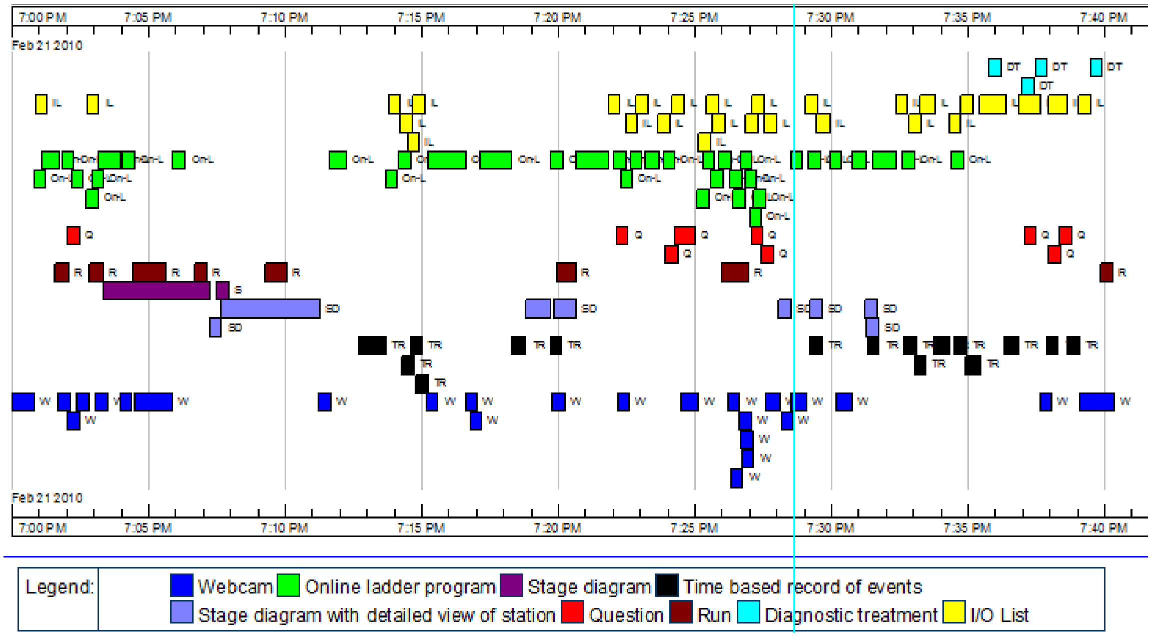
Architecture-1:



Architecture-2:

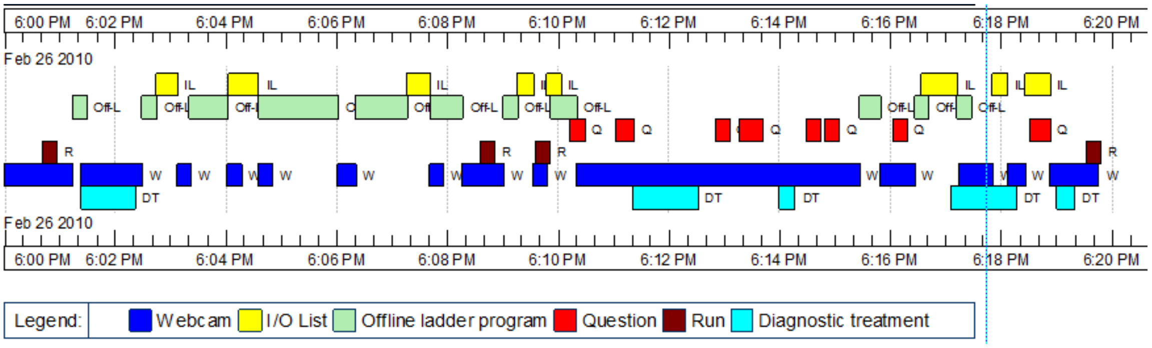


Architecture-3:

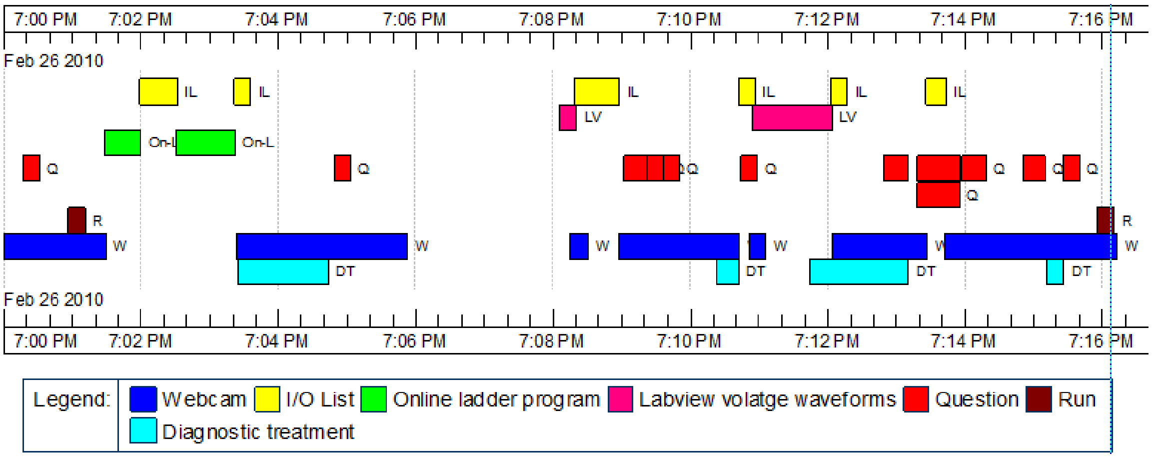


Student 2:

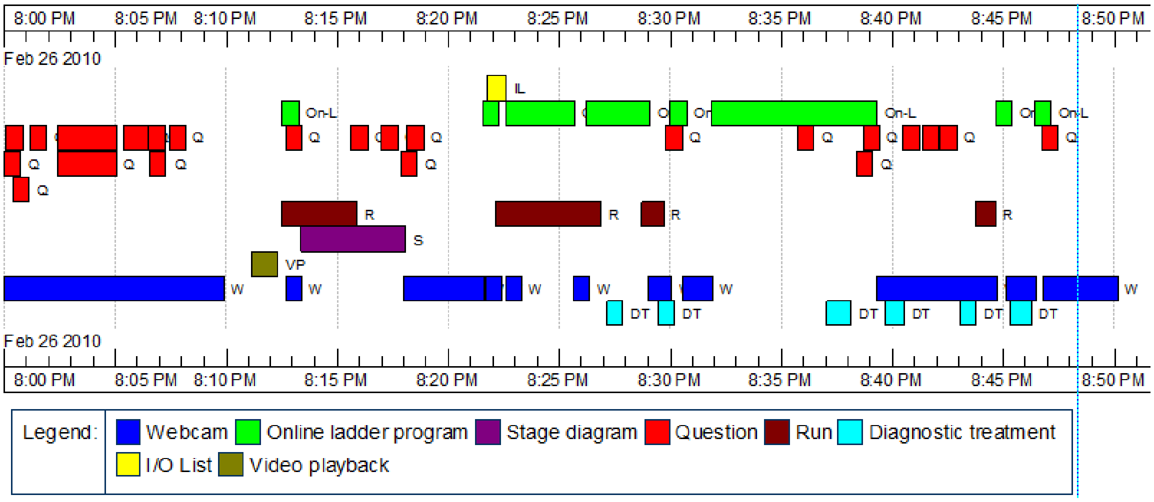
Architecture-1:



Architecture-2:

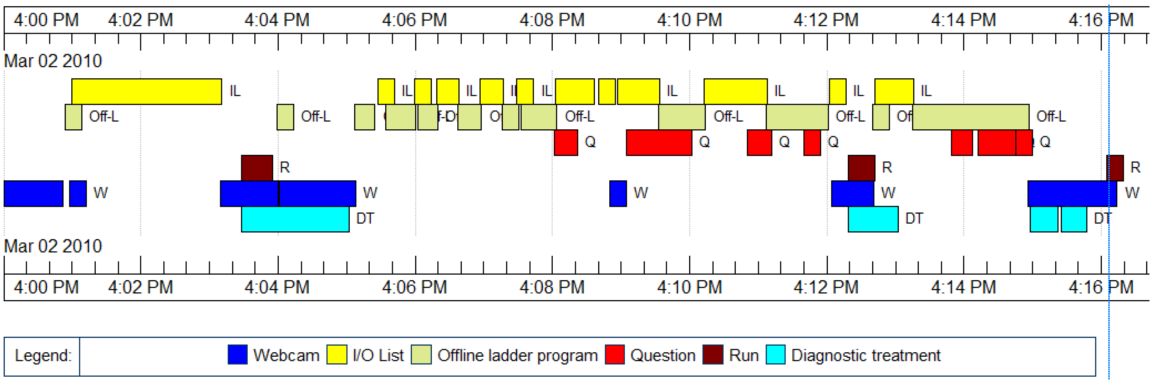


Architecture-3:

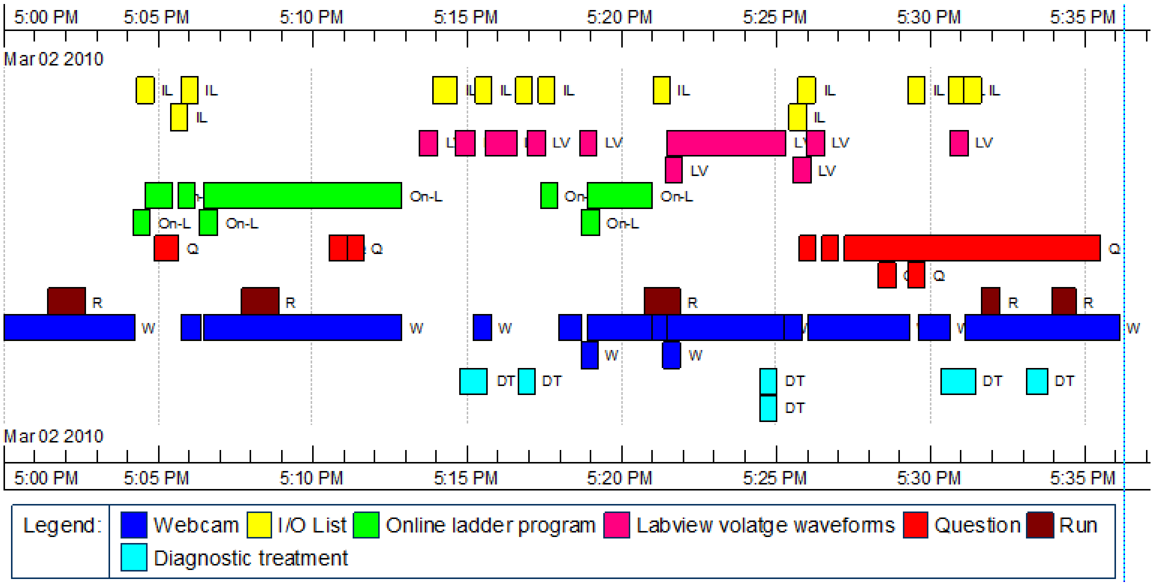


Student 3:

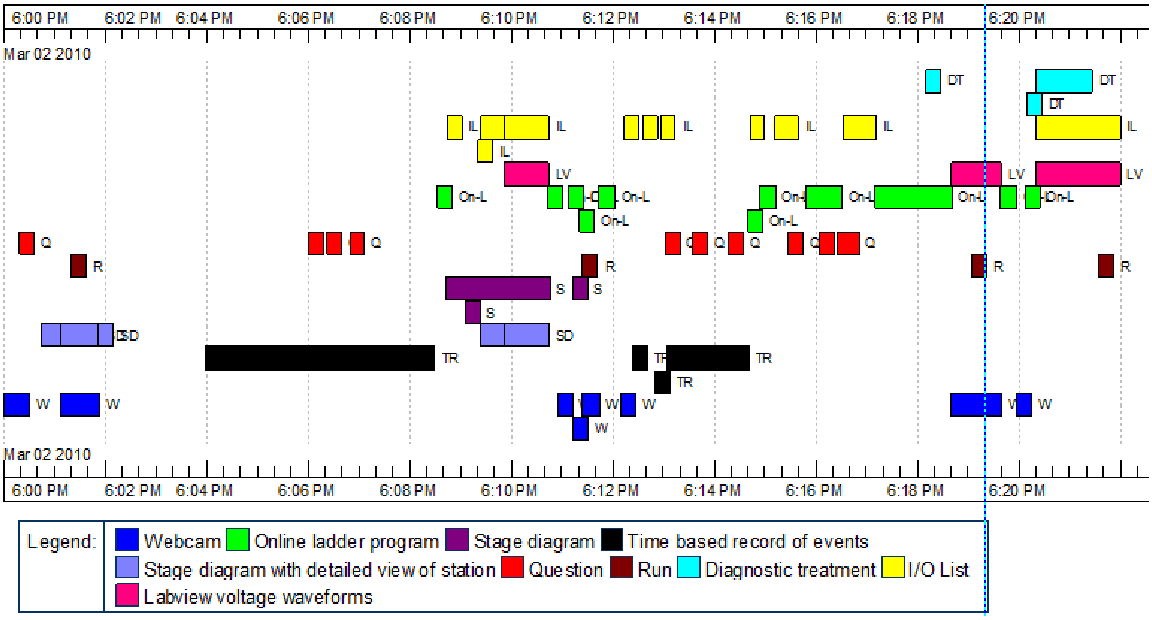
Architecture-1:



Architecture-2:

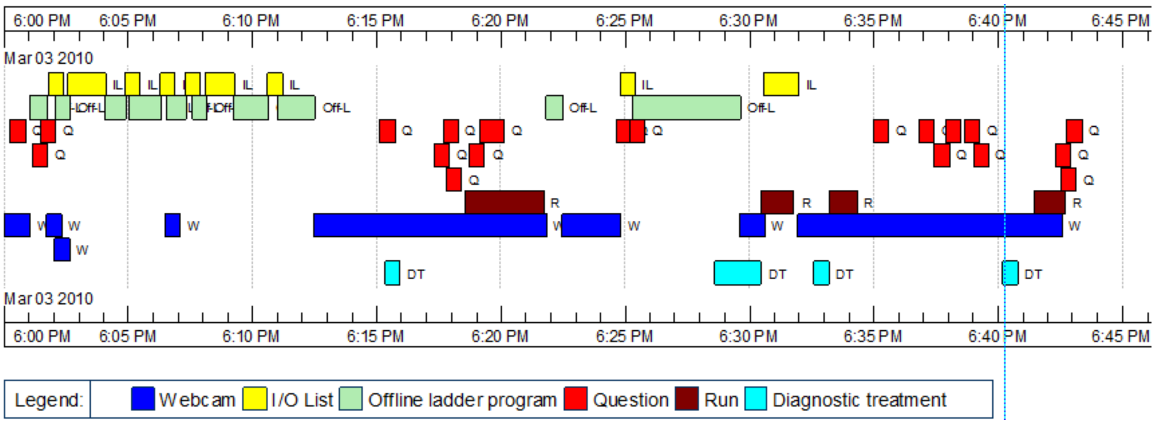


Architecture-3:

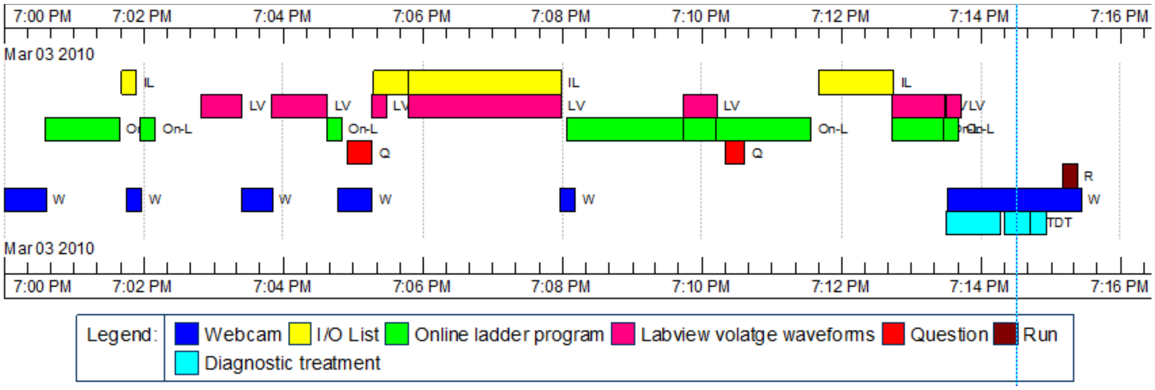


Student 4:

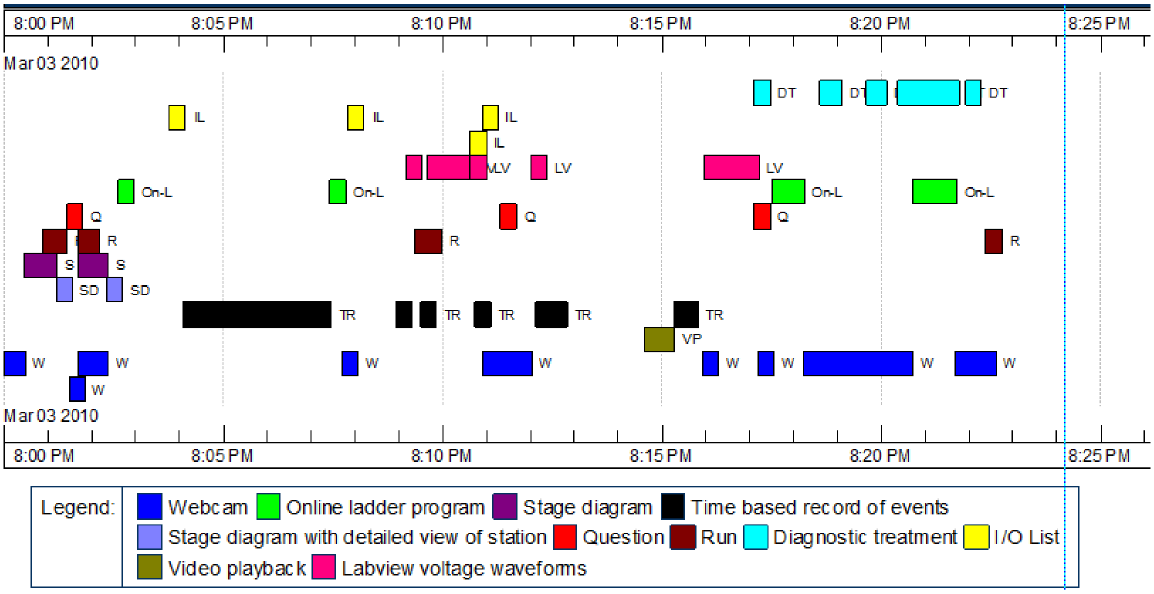
Architecture-1:



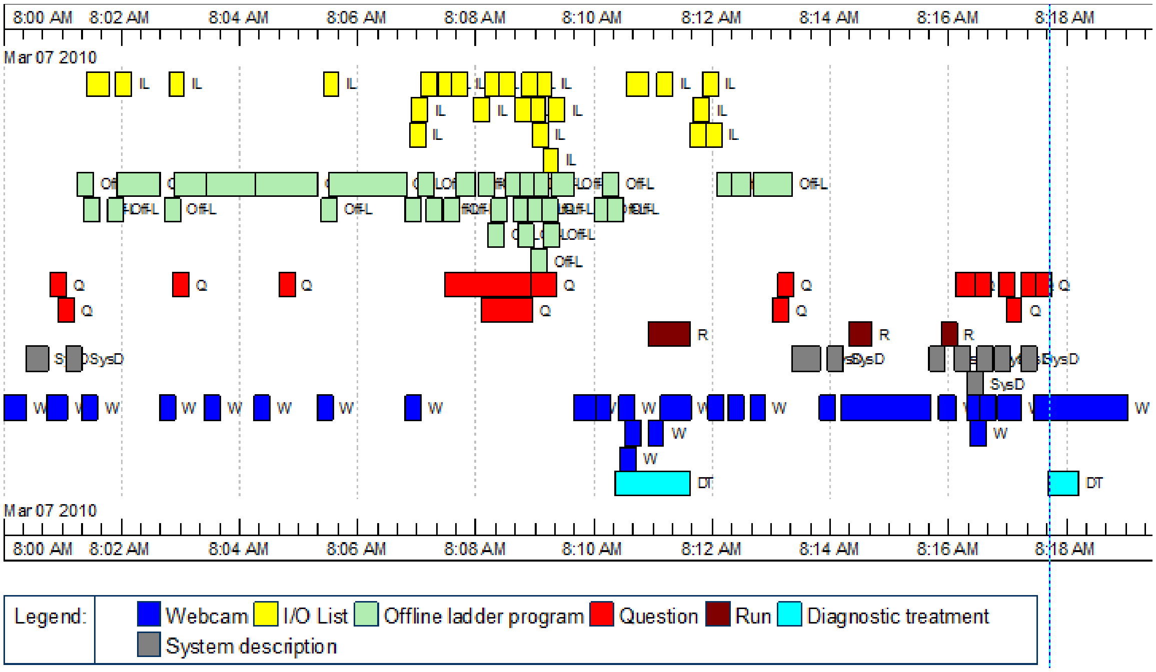
Architecture-2:



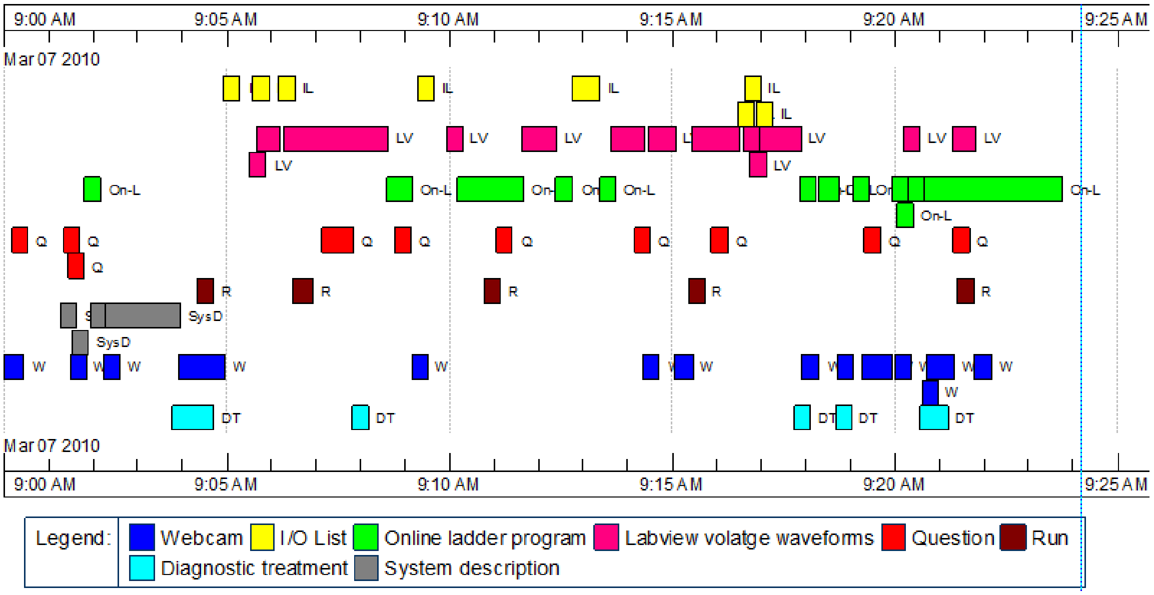
Architecture-3:



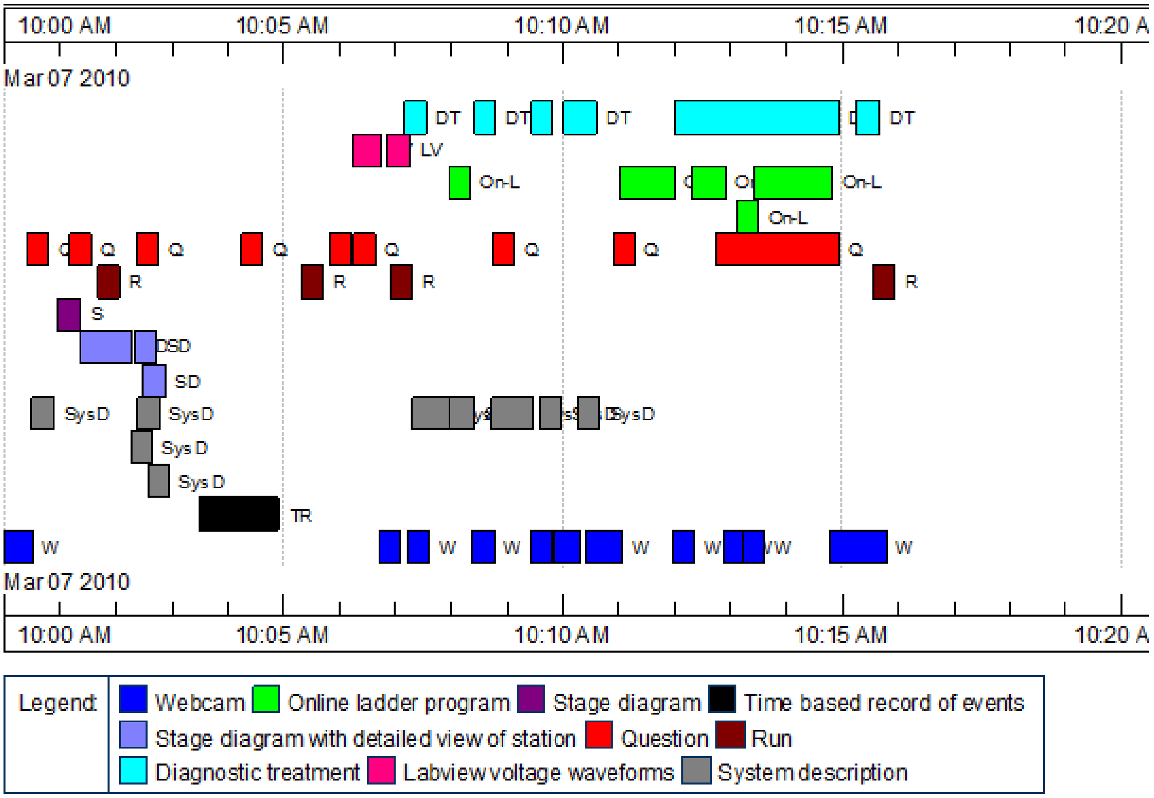
Student 5:
Architecture-1:



Architecture-2:

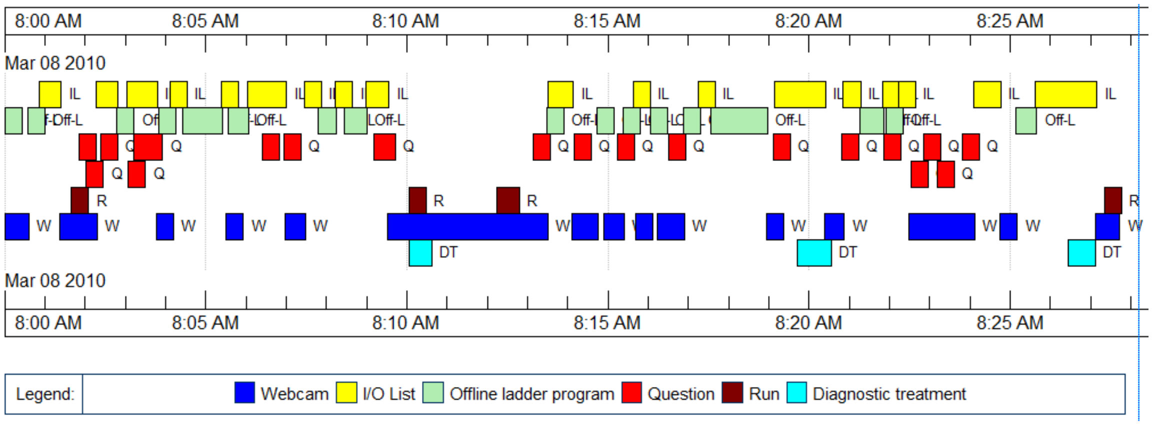


Architecture-3:

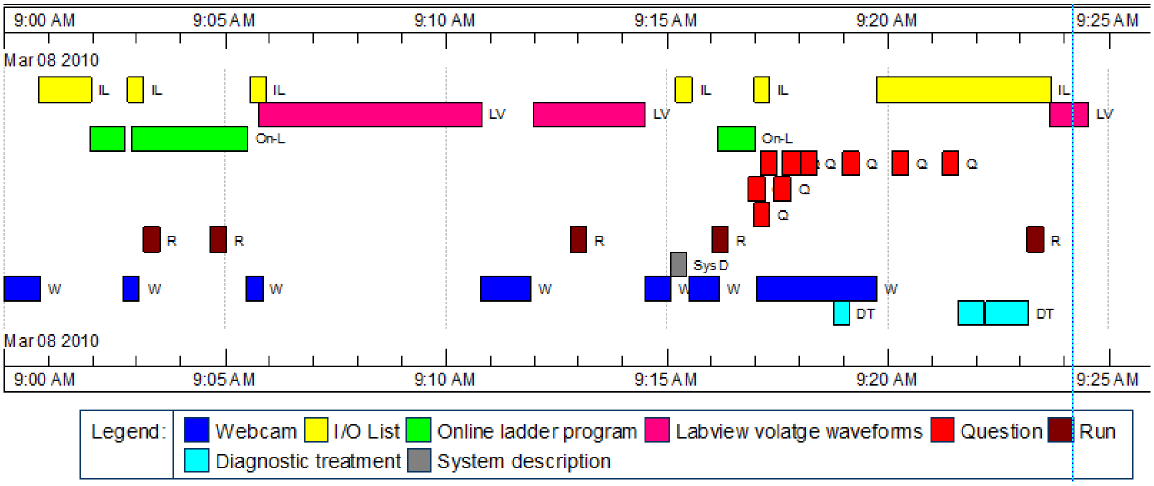


Student 6:

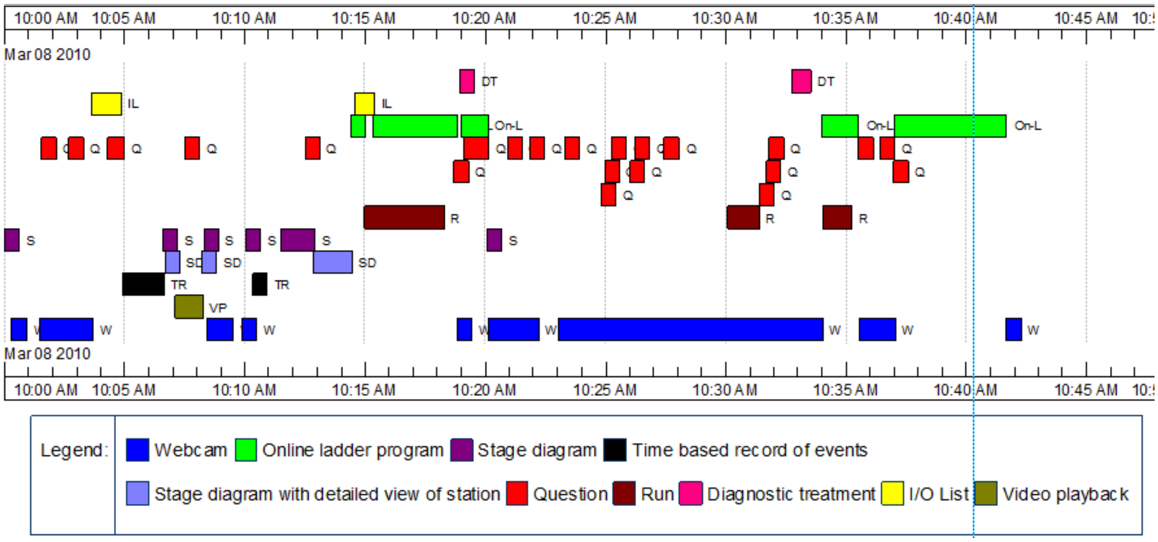
Architecture-1:



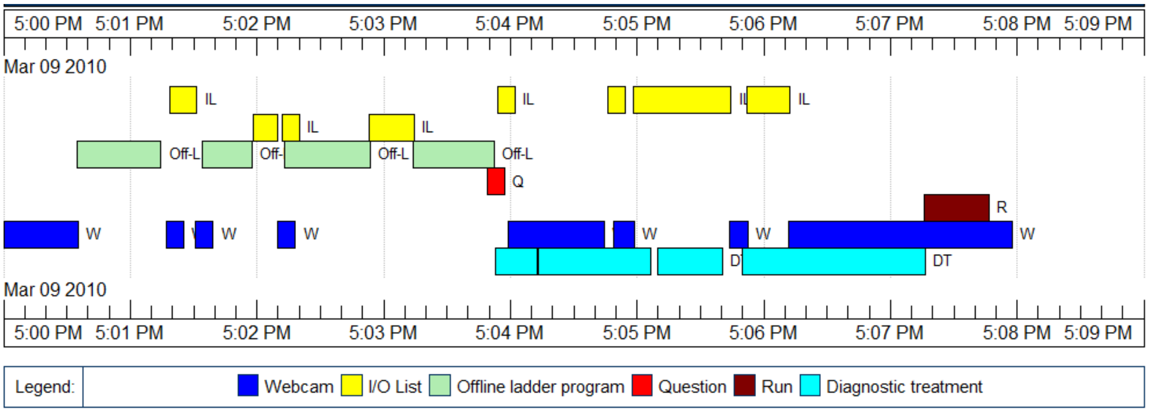
Architecture-2:



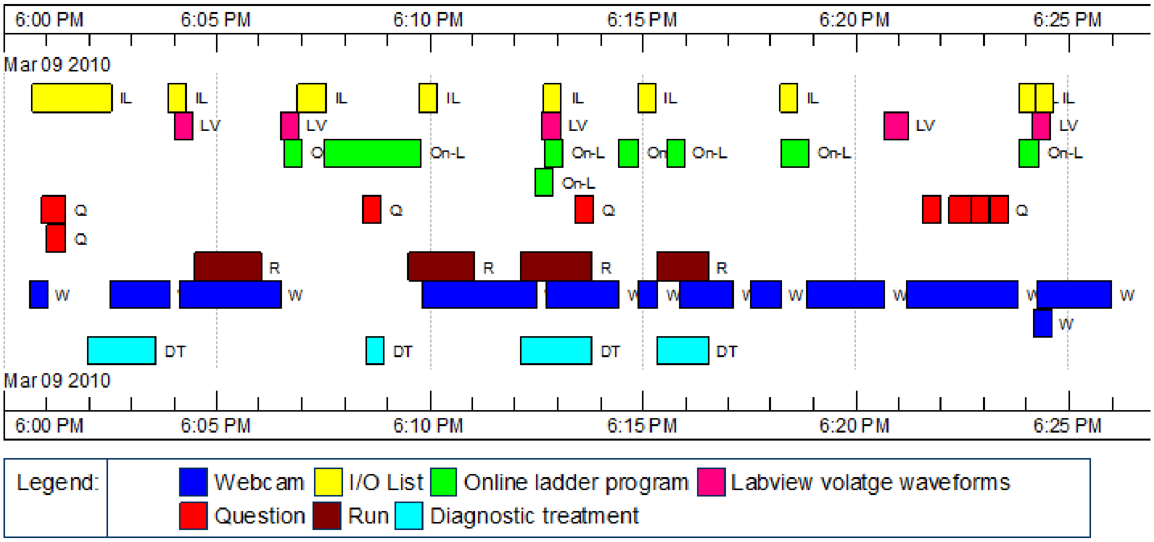
Architecture-3:



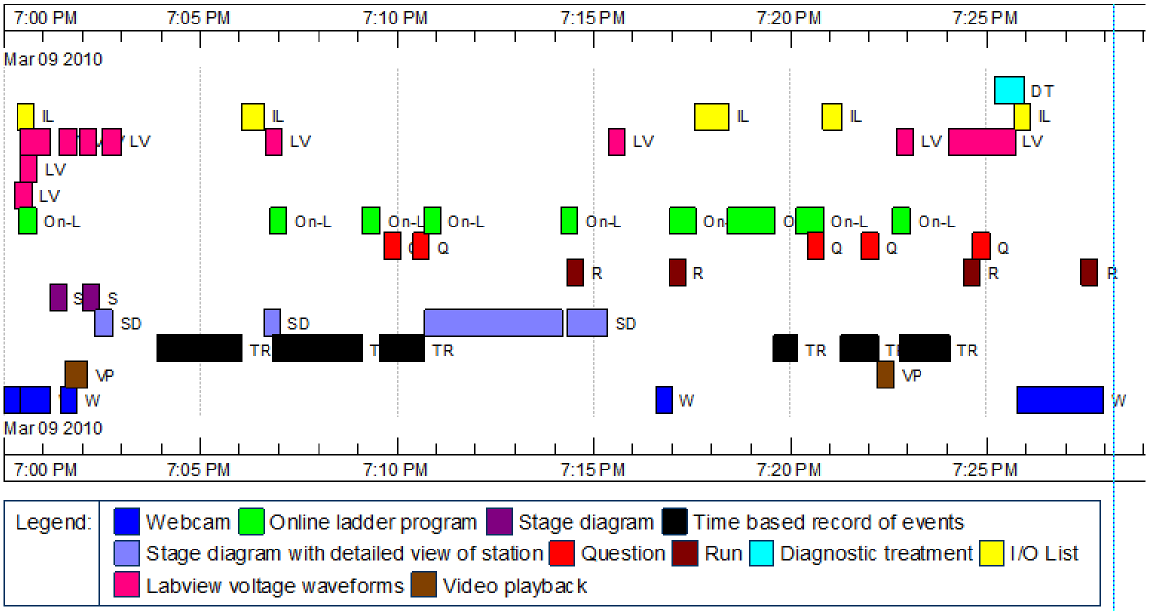
Student 7:
Architecture-1:



Architecture-2:

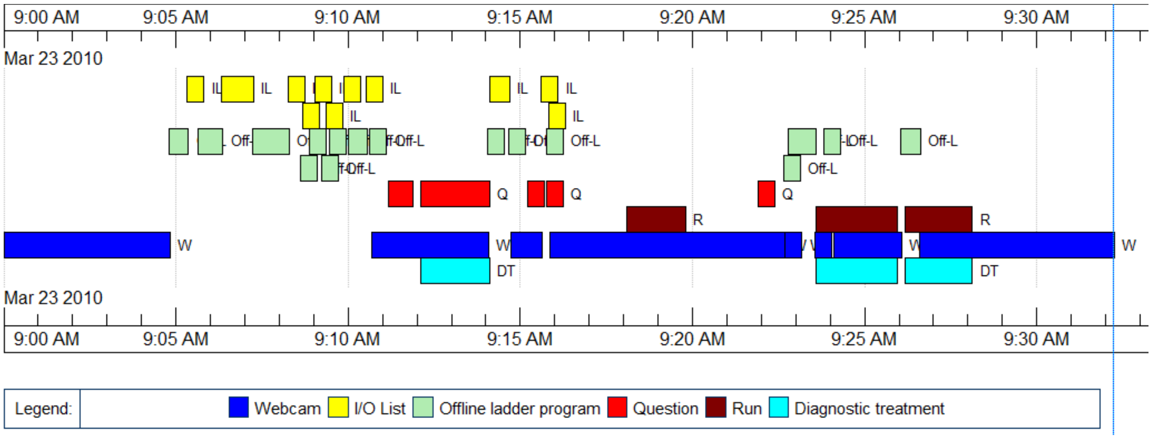


Architecture-3:

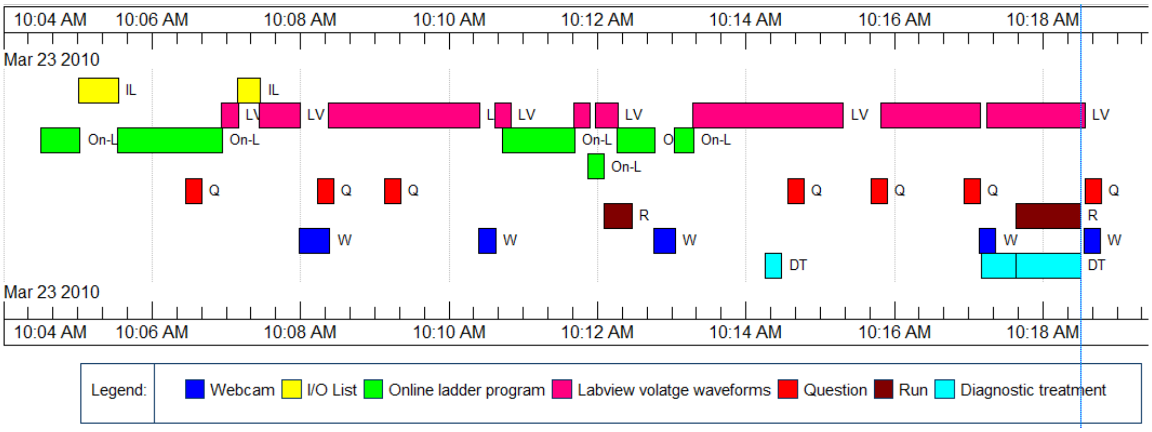


Student 8:

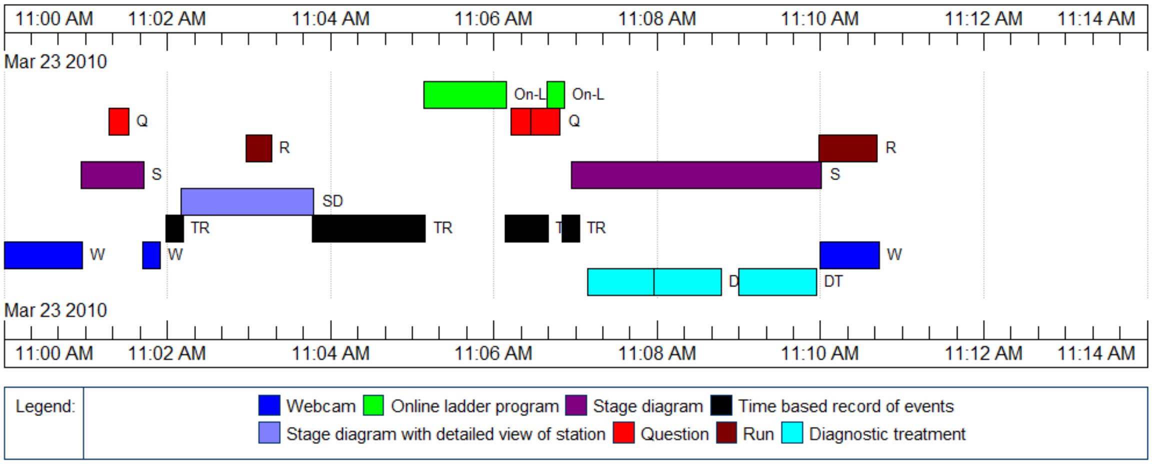
Architecture-1:



Architecture-2:



Architecture-3:



VITA

Name:	Ramnath Sekar
Permanent Address:	TAMU, Department of Mechanical Engineering
	Texas A&M University, College Station, Texas, 77843
Education:	B.Tech. Mechanical Engineering
	Jawaharlal Nehru Technological Institute, India 2006
	M.S. Mechanical Engineering
	Texas A&M University, College Station, Texas 2010
Email Address:	ramnath_skr@tamu.edu