COMPLEXITY MEASURES: THE CASE OF ENGINEERING UNDERGRADUATE EDUCATION

Zachary M. Soto Maldonado

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTERS OF SCIENCE IN

INDUSTRIAL ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
2017

Approved by:

______________________________ ____________
Lourdes A. Medina Aviles, PhD Date
Associate Professor of Industrial Engineering Department
President, Graduate Committee

______________________________ ____________
Saylisse Dávila Padilla, PhD Date
Associate Professor of Industrial Engineering Department
Member, Graduate Committee

______________________________ ____________
Betzabé Rodríguez Álamo, PhD Date
Associate Professor of Industrial Engineering Department
Member, Graduate Committee

______________________________ ____________
Jaquelina E Alvarez, M.A. Date
Representative of Graduate Studies

______________________________ ____________
Viviana I. Cesaní Vázquez, PhD Date
Chairperson of the Department

Abstract

This work investigated the relevance and impact of complexity in Project-based learning (PBL) for engineering undergraduate education. Interviews were conducted with engineering professors to evaluate weather complexity is and should be considered for two scenarios: when students are assigned the *same* project and when students work on *different* projects. The first scenario is examined in more depth with a case study on a particular course – Process Automation, were complexity metrics were *identified, adapted and implemented.* These metrics measure components interaction, process and station functionality, the number of linearly independent paths in the program and volume. To evaluate the relationship among complexity metrics and designer's characteristics and performance, a student survey was developed and implemented. Additionally, complexity prediction models are presented using randomForest, a statistical method for classification and regression problems. The intention of this work is to promote the assessment of complexity to identify and analyze PBL.

Resumen

Este trabajo investigó la relevancia y el impacto de la complejidad en PBL (por sus siglas en ingles) para la educación de ingeniería sub-graduada. Se realizaron entrevistas a profesores de ingeniería para evaluar si la complejidad es o debe ser evaluada para dos escenarios: cuando a los estudiantes se asignan al mismo proyecto y cuando el estudiante trabaja en diferentes proyectos. El primer escenario se examina más a fondo con un estudio de caso sobre un curso en particular - Automatización de Procesos, donde se identificaron, adaptaron e implementaron métricas de complejidad. Estas métricas miden la interacción de los componentes, la funcionalidad del proceso y de la estación, el número de rutas linealmente independientes en el programa y el volumen. Para evaluar la relación entre las métricas de complejidad y las características y desempeño del diseñador, se desarrolló e implemento una encuesta estudiantil. Adicional, modelos de predicción de complejidad se presentan utilizando randomForest, un método estadístico para la clasificación y los problemas de regresión. La intención de este trabajo es promover la evaluación de la complejidad para identificar, analizar y controlar la complejidad del aprendizaje basado en proyectos en PBL.

Acknowledgements

Special thanks to my graduate committee, in particular to my advisor Dr. Lourdes A. Medina for her unlimited support and for maintaining me motivated to accomplish my goal. To current and former Industrial Engineers and once members of the UPRM's IDDEAS Research Group who collaborated in this work: Melissa Rosado, Eliseo Martinez, Angel Collazo, Carlos Jusino, Ismael A. Garcia, Maricarmen Collet, Giovanni Cruz, Marisol Colón, David Del Pilar and Fernando Martínez. To my loving husband for supporting me throughout this journey. Also I want to thank Hector Espada and Yamil Odeh for all the flexibility provide to me in my current job to finish my degree. Last but not least important, Hewlett Packard Enterprise for the economic support.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

In education, a student-centered strategy known as Project-based Learning (PBL) is commonly used to provide students real-life experiences in a class environment. This strategy makes student competencies go beyond content knowledge (Sam Houston State University, 2016). PBL challenges students to research stimulating problems to create unique products (Intel, 2007) while, encouraging them to develop interpersonal skills in a flexible learning environment (Doppelt, 2003). It enables students to work in teams, communicate and be aware of time-management, applying their technical skills while exercising the technical aspects of their career (Medina et al., 2015). Students learn to make decisions in real-time in diverse environments that may include multiple stakeholders and decision makers. For example, all engineering students at University of Puerto Rico Mayagüez (UPRM) experience PBL in their Capstone projects. In particular, The Department of Industrial Engineering at UPRM has multiple courses that involve PBL such as: Process Automation (ININ 4057), Facility Layout and Design (ININ 4040), Work Measurement (ININ 4009) and Introduction to Medical Device Design Methods (ININ 5105).

Common teaching techniques combined with PBL promoted peer learning, group learning and self-motivation (Indiramma, 2014). Meanwhile, assessing collaborative work introduces a challenge not found when evaluating individual work (Webb, Nemer & Zuniga, 2002). Therefore, Researchers address the importance of *managing complexity* in real-life projects (Gottfredson & Rigby, 2009). Hence, a professor survey was submitted to UPRM engineering professors with experiences in PBL to answer the following questions:

(1) Is complexity considered for PBL in Engineering Education?

(2) Should complexity be assessed in PBL?

PBL Complexity was analyzed for two scenarios: when students are assigned the *same* project and when student work on *different* projects. Differentiating between these two scenarios enables the study of complexity for "solutions or project outcomes" when the same project is assigned, and the complexity of "project definitions or problems" when different projects are assigned. Considering the two scenarios, this study also answered the following questions:

(3) When students are assigned the *same project*, is the complexity of the *multiple solutions* generated significantly different?

(4) When students are assigned *different project,* is the complexity of the *project definitions* significantly different?

When students are assigned the same project, it is assumed they have the same lack of understanding (prior knowledge). Lack of understanding or deficiency in knowledge, increases the complexity of problem solving as a result of the increased effort that is required to overcome for the unknown information or skills (Crespo-Varela et al., 2012).

In design-related projects, engineering students generates multiple solutions while having the same requirements. To study this in more depth, a case study was performed on the process automation course, ININ 4057, which is a core course for all Industrial Engineering students at the UPRM. In this course Students learn and apply different skills to integrate electronic, mechanic and computer systems in the development of an automated process prototype (same project definition). A substantial contribution of this work involves adapting and developing complexity metrics and methods for their implementation in process automation.

> "Measurement is the key for controlling any process because it is difficult to manage what cannot be measured" (DeMarco, 1982).

Without metrics, comparisons and predictions are difficult to achieve. For the particular case of process automation in project-based learning, this research studies:

(5) **How can project complexity be assessed when students have the *same project* requirements?**

The case study contributions go beyond assessing complexity to also provide a deeper understanding on its relation with team characteristics and performance. It is hypothesized that each team member previous knowledge/characteristic influences the overall design complexity. "Research on project complexity has shown that complexity is relative not only to size and scope, but also the past experience of the project management team" (Owen et al., 2011).

Nowadays, in science and engineering fields, there are notable differences in gender, participation, performance and rewards (Sonnert and Fox, 2012). The data acquired with the case study allowed reviewing if for example, students' grade point average (GPA) and gender is related to project design complexity. According to the emerging state and national standard for assessment, it is recommended to incorporate small groups into large-scale assessment (Webb, Nemer & Zuniga, 2002). A study that took place within Ford Motor Company with over 270 employees, showed a clear relationship between team composition (diversity), complexity of task and team performance (Higgs et al., 2005). With these motivations, and for engineering project-based learning setting, this research addressed the following question:

The design team characteristics considered include: age, gender and knowledge. Also performance, individual contribution, abilities and difficulty was considered. Last, team dynamic was also evaluated.

To summarize, primary objective of this research is to investigate the relevance and impact of complexity in PBL for engineering undergraduate education. Accordingly, research objectives include the:

- Creation and implementation engineering professors survey to:
    - understand current considerations and assessments of complexity for PBL
    - explain project complexity when student are assigned different requirements
- Identification, adaptation, development and implementation of complexity metrics in order to explain project complexity when student are assigned common requirements
- Creation and implementation of student factor survey to analyze the relationship of project complexity with team characteristics and performance

With the results from this work, the aim is to promote a culture where complexity is considered, that includes to identify and analyze complexity in PBL.

The following sections provide research background along with the motivations.

Chapter 2: Literature Review

2.1 Overview

This chapter summarizes background information and relevant literature to this work. Background information includes a discussion about project-based learning (PBL) in Section 2.2 and project complexity in Section 2.3. Section 2.4 provides a review of literature addressing complexity metrics. Section 2.5 is focused on team characteristics and performance. Finally, a summary is provided along with research contributions in comparison to the literature reviewed.

2.2 Project-based Learning

With a considerable amount of literature dedicated to showing the PBL's benefits and keys for successful implementation, this concept is defined in multiple ways. From the analysis of different definitions, PBL can be explained as:

> A teaching strategy that enables students to develop competencies and gain deeper knowledge through active explorations of real-world problems.

Table 2.1 summarizes PBL definition from various sources and Figure 2.1 is visual representation of PBL definition word frequency.

Table 2.1: PBL Definition

Source	PBL Definition
Thomas, 2000	Model that organizes learning around projects.
Doppelt, 2003	Well-known method for imparting thinking competencies and creating flexible learning environments.
Balve and Albert, 2015	Course that display motivation and meaningful real-world task in the center of the students' attention.
Buck Institute for Education, 2016	Teaching method in which students gain knowledge and skills by working for an extended period of time to investigate and respond to an engaging and complex question, problem, or challenge.
Vega, 2015	Dynamic classroom approach in which students actively explore real-world problems and challenges and acquire a deeper knowledge.
Medina, 2015	Platform that enables student to work in teams, communicate and be aware of time-management while practicing technical aspect of their concentration.

Source	PBL Definition
Intel, 2007	Instructional model that involves students in investigations of compelling problems that culminate in authentic products.
Krajcik et al., 2006	Overall approach to the design of learning environments.

Figure 2.1: PBL Definition Word Cloud

PBL offers tremendous benefits to both, student and instructors. PBL drives students to the central and technical concepts of their concentration with a goal-directed process that involves the application of skills such as design, decision-making and problem solving (Thomas, 2000). A PBL environment allows student teams to examine questions, suggest hypotheses, discuss/challenge ideas, and attempt new things (Krajcik et al., 2006). Students are forced to use their own criteria to complete projects that do not take predetermined paths (Thomas, 2000).

PBL helps instructors succeed in their mission since it accommodates students with varying learning styles and differences. It makes education more engaging for students by improving learning and developing success skills for college, career and life. It enhances student's skills development for living in a knowledge-based, highly technological society while making teaching enjoyable and rewarding (Buck Institute for Education, 2016) Furthermore, PBL brings a new relevance to the learning at hand encouraging authentic assessment that promotes lifelong learning.

2.3 Complexity of Projects

Projects can be define as an individual or collaborative effort to accomplish a particular objective, for example a unique product, service or outcome (Project Management Institute, 2016). Lewis (1999) defines it as a one-time task that has a specific start and end date, as well as a particular scope, budget and performance to be attained. Complexity has various definition in literature. For example, some say is related to the difficulty or lack of understanding, a phenomenon in a given context or environment (Gul and Khan, 2011; Crespo-Varela et al., 2012). Ireland (2007) think is related to an item having one or more component or variables. Therefore, having a clear understanding of the operational definition of complexity within the project being managed its crucial, since it varies depending on the domain. Understanding the sources of complexity and its magnitude might help identify the abilities and competencies needed to deal with a problem (Remington, Zolin & Turner, 2009).

As summarized in Table 2.2 below, literature shows there are efforts to managing complexity by identifying sources of complexity among different domains. For instance, in the case of transportation projects, Gransberg et al., (2013) evaluated 18 projects from different countries – Canada, New Zealand, United States, and United Kingdom (see Table 2.2 below). As a result, they propose a framework from which the sources of complexity for transportation project can be conceptualized. They also developed a tool, the complexity footprint, to measure and visualize the various dimensions of project complexity. In general, the study searched to better understanding and prioritization of the available resources. They added financing and context as two new dimension to the traditional three-dimensional project management theory that involves: cost, schedule and technical. By doing so, they elevated the visibility of complex project context which represent both the controllable and uncontrollable factors that will be faced during the delivery of complex projects.

Table 2.2: Sources of Complexity by Category

Source	**Domain**	**Sources and types of project complexity**
Baccarini (1996)	Project Management emphasis	Differentiation and interdependency
Hussein et al. (2014)	New Products and Process Development Project	Product development projects: interdependency between tasks and the novelty of the project Process improvement projects: diversity and multiplicity of end-users and uncertainty
Bosch-Rekveldt et al., (2010)	Process Engineering Industry	Technical, Organizational and Environmental complexity
Gransberg et al., (2013)	Transportation Project	Five-dimensional sources of complexity: cost, schedule, technical, financing and context
Ireland (2007)	Planning Standpoint	Two dimension source of complexity: Technical Complexity and Management Complexity

In summary, project complexity is addressed in the context of project management in which experienced project managers are the SMEs helping define the *type of complexities* that they encounter. As shown in Table 2.2, there are sufficient efforts to identify the *sources of complexity* for the management and implementation of real projects. However it is also desired to understand the quantitative approach used to measure and manage complexity. Hence a review of complexity metrics was performed on the Section 2.4.

2.4 Complexity Metrics

Relevant literature on complexity metrics in general was addressed (summarized in Table 2.3 below). The objective was to identify and adapt, when possible, existing metrics to measure process automation project complexity. Accordingly, diverse complexity metrics were found in the literature, and are divided in the following categories: design complexity, software complexity, and choice complexity.

2.4.1 Design Complexity

Most relevant design complexity metrics involve the study of modules and interactions (Keating 2000), product functionality (Bashir & Thomson, 1999), and product variations (Roy Evans, Low and Williams, 2011).

For the assessment of hardware design quality at early stages of the design cycle, Keating (2000) proposes the study of modules at each level of hierarchy and interactions with the belief that quality and functional correctness are not tested in, rather design in. A block diagram is performed to have a reasonable explanation of product functionality. Blocks are decomposed into a hierarchy of what the study calls "too many levels" such that the design is divided into independent units. After the block diagram is completed, the metric is implemented with the sum of squares of the number of modules (M) and the interfaces (I).

Meanwhile, the Product Complexity (PC) metric makes an assessment based on product *functionality* using a deductive approach (Bashir and Thomson, 1999). This metric uses a hierarchical approach to decompose product functions into different levels. The more sub-function at any level and depth of the functional tree, the greater the complexity (lower functions in the tree, imply more complexity). The metric count the number of functions at each level and weight them by the number of levels. According to Faulconbridge and Ryan (2003), complex technical projects, can only be manage effectively when functional requirements are analyzed.

Roy, Evans, Low and Williams, (2011) measures product complexity from the perspective of product variations. The metric involves calculating a design ratio (DR) which is based on the commonality of components for the end-product. For example, a low DR indicates less commonality of the component in the design and therefore higher complexity.

Similar, focused on assembly, Mathieson, Wallace and Summers (2010) develop a model to predict assembly time of a system based on complexity metrics of the system architecture using that of a power regression.

2.4.2 Software Metrics

Software metrics are the oldest and most proven complexity metrics. A well-known metrics is the Cyclomatic Complexity metric (V(G)). V(G) measures the number of linearly independent paths in a program control graph. McCabe (1976) worked on the mathematical technique that allowed identifying software modules, and testing difficulties. The approach was to measure and control the number of path in a program. Research findings included that complexity is independent of size, but it depends on the decision structure of a program.

On the contrary, Halstead (1977) defined the Software Science Metric with the belief that the effort required to implement a computer program is proportional to the program size. The metric measures complexity as related to the *length and volume* of a program. For its implementation, any symbol or keyword in a program that specifies an algorithmic action is considered an operator, and any symbol used to represent data is considered an operand. As a result, the length of the program becomes a function of the unique operators and operands.

Finally Basili and Perricone (1984), studied 90,000 lines of code (LOC) of a software project which general purpose was satellite planning and concluded that the larger the module, the less error prone it was. LOC metric measures the number of lines (statements) in a program. However, Yu (2010) mentioned that even though this metrics is easy to understand, LOC ignores jumps in the software as well as complexity on each code line.

2.4.3 Operator Choice Complexity

In the context of decision-making, the Operator Choice Complexity (OCC) deals with the decision operators can make regarding assembly and the risk associated with their choice (Fast-Berglund, Fässberg, Hellman, Davidsson, & Stahre, 2013). It is stated that decision making is needed more when there are additional variants and parts to be handled. In general terms, the study focuses on determining if there are any correlations between the areas of complexity, cognitive automation and quality. The areas of complexity are defined as the nature of product, processes, and strength of interactions, among others; cognitive automation refers to the decision making in production that enables error-free products (Fast-Berglund, Fässberg, Hellman, Davidsson, & Stahre, 2013).

Fast-Berglund, Fässberg, Hellman, Davidsson, and Stahre (2013) formulation is based on the average uncertainty or randomness in a choice process and occurrence probability to get a complexity measure for the stations. Formulation independent variables include the number of variants that occurs at each station and the demand of each variant. In their experiment, operator's performance depended on assembly errors extracted from seven station for a 16 week time frame and retrieved from an internal quality system named ATACQ. The study concluded that the main cause of complexity is due to assembly workers' restricted timeframe and workspace with positive correlation between OCC and assembly error and more than 60% of the assembly task lacking cognitive support.

2.4.4 Summary

Table 2.3 summarizes all the metrics found relevant to this research with possible application to the study of process automation projects. Some of these metrics will be studied in more detail for their adaptation and the development of implementation methods.

Table 2.3: Review of Metrics

Source	Metric Name	Purpose/Definition	Notation	Equation
Keating, 2000	Complexity of the partition (C)	Measures based on component *interaction*	M - number of modules I - number of interfaces	$C = M^2 + I^2$
Bashir and Thomson, 1999	Product Complexity (PC)	Measures component *functionality* hierarchically	Fj - number of functions at level j i - number of levels	$PC = \sum_{j=1}^{l} F_j j$
Roy, Evans, Low and Williams, 2011	Design ratio (DR)	Measures from the perspective of product variation	ni -number of product variants that use part variant i n -total number of product variants	$DR(i) = \frac{n_i}{n}$
Basili and Perricone, 1984	Lines of Code	Metric to determine the *size* of the program	li-lines of code i	$\sum l\, i$
McCabe, 1976	Cyclomatic Complexity Metric (V(G))	Measures the number of linearly independent paths in a program	n –vertices e –edges p -connected components	$v(G) = e - n + p$
Halstead, 1977	Halstead Software Science Metrics	Determine a quantitative measure of complexity directly from the operators and operands in the program, related to the *length* and *volume* of a program.	n1 -number of unique operators, n2-number of unique operands, N1-total number of operators, N2-total number of operands.	$V = (N_1 + N_2)\log_2(\eta_1 + \eta_2)$
Fast-Berglund, Fässberg, Hellman, Davidsson, & Stahre, 2013	Operator Complexity Metric	Quantify human performance on making choices	Pij- occurrence probability of a state j in the random process i, C- constant (depending on the base of the logarithm function chosen)	$Hi(\rho i1, \rho i2, \ldots, \rho iMi) = -C \cdot \sum_{j=1}^{Mi} \rho ij \cdot \log \rho ij$
Mathieson, Wallace and Summers, 2010	Assembly Time Metric (t_a)	Predict the assembly time of a system based on the architecture of that system	APL-average path length n- number of elements PLD- path length density	$t_a = APL\ x\ n^{(1.185+PDL)}$

2.5 Team Characteristics and Performance

Real-life project management and implementation requires continuous teamwork and collaboration. Accordingly, effective undergraduate engineering education must include exposing students to similar experiences throughout their course curriculum. Student attributes and team composition are factors that influence group project outcomes where the same student may perform differently depending on the group (Webb, Nemer & Zuniga, 2002). Therefore, the benefit from collaborative assessment work is not necessarily found in individual assessment.

The comparison between group and individual performance is affected by ability, gender, and affiliation preferences (Hills, 1982). While it can be generally stated that group performance is superior to individual performance (Hills, 1982), an exceptional individual can be superior to that of a committee, especially, when solving a complex problem (Davis, 1969). High-ability students perform well in homogenous group and group interaction is a strong predictor for student performance (Webb, Nemer & Zuniga, 2002). In particular, group interaction examples that impact project outcomes include: leadership efforts, approval or disapproval of fellow group members, and influence attempts, among others (Guzzo & Shea, 1990).

In terms of student attributes, the literature emphasizes the role of student's GPA as an indicator of performance (Sonnert & Fox, 2012). GPA was found to explain retention of student across fields (Aitken, 1982). Meanwhile, the study of a sample of 5,223 senior students from a midsized Midwestern public university (between 2001 and 2009) showed that student's gender does have an effect on GPA (Tessema, Ready & Malone, 2012). Results showed that female perform better with an average a GPA of 3.37 in comparison to males that had an average GPA of 3.13.

2.6 Summary and Contributions

In comparison with the existing literature, this work opens up the paradigm of complexity for PBL, particularly for process automation. While the importance of project complexity is clearly stated for the management and implementation of real-problems in the context of project management for different domains, there is a gap in the consideration of this concept for project-based learning. This work developed the project complexity concept further. As part of the methodology, a case study on process automation was performed. Complexity metrics reviewed from the literature were adapted and implemented to fourteen projects. At the same time, team characteristics and performance were collected from a student factor survey. Last, the relationship of project complexity with team characteristics and performance was assessed.

To conclude, the major research contribution is for this method (the use of complexity metrics) to be used in school systems and higher education on a large scale to provide students the venue to identify, analyze and control complexity. Faculty will be likewise, to use complexity measures as part of project evaluations.

Chapter 3: Methodology

3.1 Overview

Research question and objectives were addressed in four major stages as shown in Figure 3.1. The first stage (Section 3.2) involved creating and launching a survey to engineering faculty with experience in PBL to answer the first four research questions. In contrast, the second stage (Section 3.3.1) was focused on identifying and adapting complexity metrics previously identified in literature, followed by the development of implementation methods for a specific domain, process automation project, as given in Process automation (ININ 4057) course. This section answers research question five and also research question 3. Data concerning team characteristics and performance were collected as part of the third stage (Section 3.3.2) in which a web-based survey was launch for students who designed process automation as required in the Process Automation (ININ 4057) course. After the three stages were implemented, the fourth and last stage integrates all the information obtained to answer the last research question and to suggest a complexity prediction model (Section 3.3).

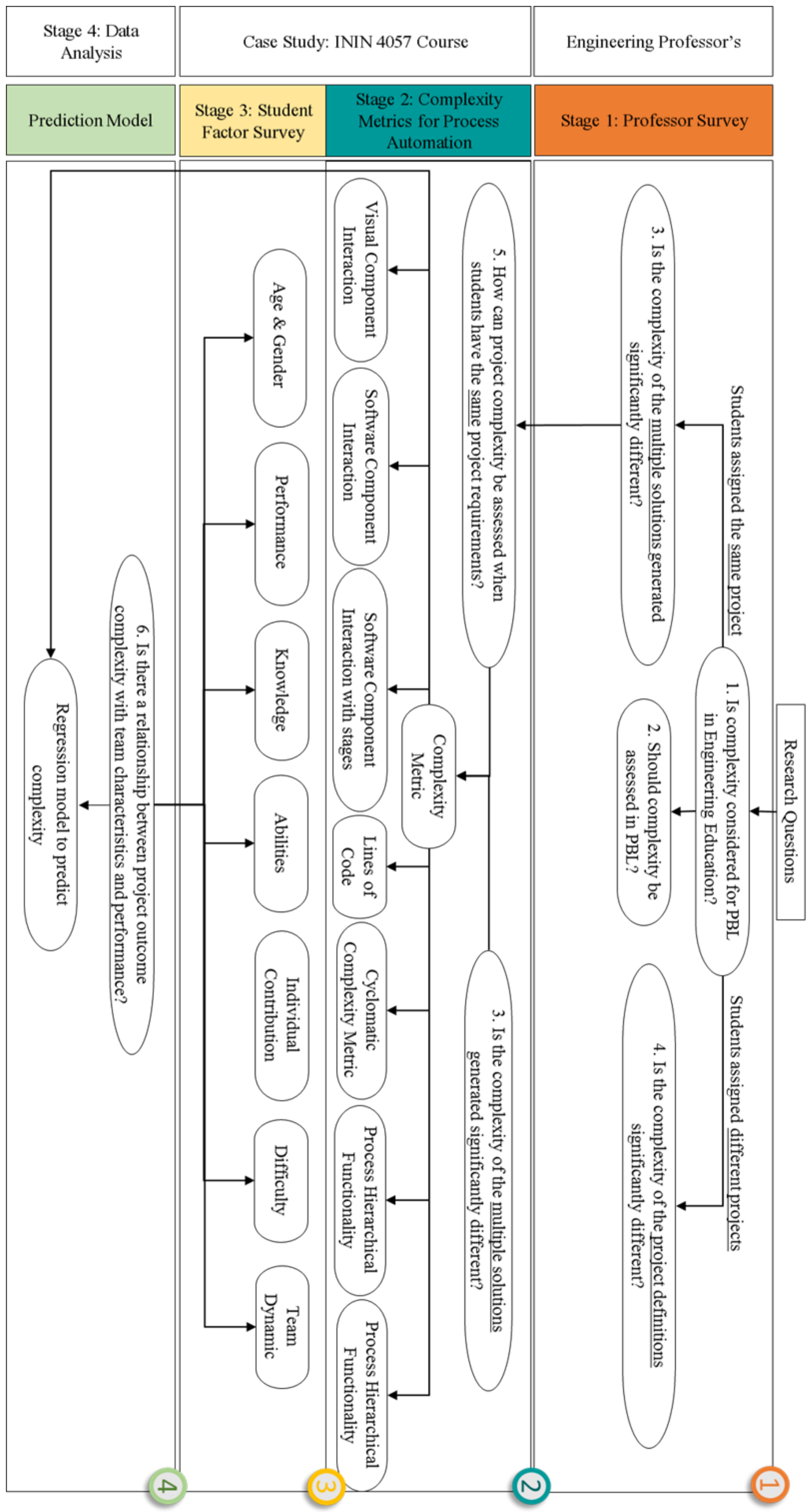


Figure 3.1: Research Overview

3.2 Stage 1: Development of Professors Factor Survey

In order to assess complexity in engineering undergraduate education, a 14-question survey was created (see Table 3.1 below). Subject matter experts (SME's) were identified to be engineering faculty with experience implementing PBL. The purpose of the survey is to gain knowledge on how complexity is currently managed for two scenarios: (1) when students are assigned the same project requirements (the focused is on projects *solution*) and (2) when students are assigned different projects (the focus is on project *definitions*).

The questionnaire, illustrated in Table 3.1, is divided in four sections: (1) screening or profile; (2) define complexity; (3) explain current consideration of complexity in engineering education; and (4) provide opinion regarding complexity for the two scenarios specified above. For instance, part of SMEs' opinion includes explaining if they have experienced significant differences in complexity of the two scenarios - project definitions and solutions.

Table 3.1: Professor Factor Survey Questions

Research Question	**ID**	**Professor Factor Survey Questions (*Spanish* / English)**
Is complexity considered for PBL in engineering education?[2,3]	5	*En el contexto de cursos con proyecto (donde se aplica PBL), defina, qué es complejidad para usted. Explique.* / In the context of courses with project (PBL), define, what it is complexity for you. Explain
	9 10	*Actualmente, ¿se considera la complejidad en la enseñanza mediante proyectos?*/ Is complexity currently considered in Project-Based Learning? Scale 1(Substantially not considered) to 7 (Substantially considered) *De ser así, ¿cómo se considera?* / If so, how do you consider it?
	11 12	*Actualmente, ¿se emplean* ***métricas*** *objetivas para medir la complejidad de la definición o evaluación de proyectos*/ Is there objective metric currently used to measure project definition or evaluation complexity? Scale 1(Substantially not used) to 7 (Substantially used) *De utilizarse métricas, favor especificarlas y evaluarlas indicando con qué frecuencia integra esa métrica de complejidad en la evaluación de proyectos.* / If complexity metrics are used, please specify which and evaluate with what frequency you integrate each of those metrics in project evaluation. Scale 1(Never) to 7 (Always)
Should complexity be assessed in PBL?[4]	6	*Califique el impacto de las siguientes variables en la complejidad de proyecto en cursos de ingeniería.* / Rate the impact of the following variables in project complexity in engineering courses.

Research Question	**ID**	**Professor Factor Survey Questions (*Spanish* / English)**
		Scale 1(Not related significantly to 7 (Related significantly)
	7	*Califique cuán importante es medir la complejidad en la enseñanza con proyectos /* Rate how important is to measure complexity in Project-Based Learning. Scale 1(Extremely irrelevant) to 7 (Extremely relevant)
	8	*Califique cuán importante es tener* ***métricas objetivas*** *para* ***evaluar*** *la complejidad de los proyectos de ingeniería.* / Rate how important is to have objective **metrics** to evaluate the complexity of project in engineering. Scale 1(Extremely irrelevant) to 7 (Extremely relevant)
Is the complexity of the multiple solutions generated significantly different?[4]	14	*Cuando a los estudiantes se les asigna el* ***mismo*** *proyecto, califique cuanto difiere* ***la complejidad*** *de las* ***soluciones*** *provista por los estudiantes.* / When students are assigned the same project, rate how different the complexity of the solutions provided by students are. Scale 1(Substantially the same) to 7 (Substantially different)
Is the complexity of the project definitions significantly different?[4]	13	*Cuando a los estudiantes se les asigna* ***diferentes proyectos****, califique cuan diferente es la* ***definición*** *(especificaciones) del proyecto.* / When students are assigned different project, rate how different is the project definition (specifications). Scale 1(Substantially the same) to 7 (Substantially different)
Professor Profile[1]	1	*Seleccione el departamento de ingeniería al que pertenece.* / Select the engineering department you are part of.
	2	*¿Cuantos años de experiencia tiene como profesor(a)?* / How many years of experience you have as a professor?
	3	*¿Cuantos años de experiencia tiene enseñando cursos basados en proyectos?* / How many years of experience you have teaching Project-based learning course?
	4	*¿Qué cursos ha ofrecido donde se implementa el aprendizaje a través de proyectos (“Project-based Learning”) ó PBL por su siglas en inglés?* / What courses have you offered where learning is implemented through projects (Project-Based Learning)?

Legend: 1-profesor profile, 2- define complexity, 3- consideration of complexity in engineering education, 4-opinion regarding complexity

3.2.1 Analysis of Professors Factor Survey

The first set of data that needed to be analyzed, was the result from the professor’s survey. This results were analyzed using 1-Sample Wilcoxon test, a nonparametric hypothesis test for the median of a single population. Hypothesis tests prove if there is enough evidence to support claims related to research questions. Hypothesis tests statements are provided below:

H1o: *Complexity is considered for PBL in Engineering Education.*
H1a: *Complexity is not considered for PBL in Engineering Education.*

H2o: *Complexity should be assessed for PBL in Engineering Education.*
H2a*: Complexity should not be assessed for PBL in Engineering Education.*

H3o: *When students are assigned the same project, the complexity of the multiple solutions generated is different.*
H3a: *When students are assigned the same project, the complexity of the multiple solutions generated is not different.*

H4o: *When students are assigned different projects, the complexity of the project definitions is different.*
H4a: *When students are assigned different projects, the complexity of the project definitions is not different.*

Next sections describe remaining methodology stages.

3.3 Case Study: Process Automation (ININ 4057) Course

Automation has become a key factor for many manufacturing processes who are impacted by workforce reduction along with workload increase (McQuilken, 2014). Cost reduction, higher production rates, better product quality and reduced factory lead times are some examples among the many advantages of automation.

In unison with industry trends, universities include in their curriculum, introductory engineering elective courses, process automation and robotics. For instance, the Department of Industrial Engineering at the University of Puerto Rico at Mayagüez requires all students' in the program to take Process Automation (ININ 4057), Fundamentals of Electrical Engineering (INEL 4075), Fundamentals of Electronics (ININ 4076), and Basic Electronic Laboratory (INEL 4077), Manufacturing Process (INME 4055), Manufacturing Process Laboratory (INME 4056), among other courses.

In the Process Automation course (ININ 4057), students learn and apply their skills in electronics, computer science, and programming. Specifically, the course syllabus states that students should be able to: (1) identify and use industrial sensors and actuators as main components of a process (2) creatively integrate electric, pneumatic and mechanical systems to automated process (3) formulate and code the control logic to run a process in real time and (4) use software to build a Human Machine Interface (Medina, 2013).

When project starts, students are divided in groups of two or three (the majority) and receive the description of a manual process that they have to automate during the semester, a five-month period. The design of the automated process is divided in four phases with detailed rubrics provided at each phase. In the first phase, the design process is executed– where student work on the concept and come up with a design proposal. Students are given flexibility in terms of the use of software. They are allowed to make designs with free hand or use software they know such as Sketch up. Special emphasis is given to concept generation and ideation with methods such as radial thinking and morphological chart. This design is evaluated by the instructor and influenced by the group.

The second phase involves the construction of the structure, with Fishertechnik components with all the electric connections, inputs-X and outputs-Y. The third phase is the most challenging part of the project that requires programming in the Programmable Logic Controller (PLC), using Ladder Logic, and troubleshooting the automated process model to make sure it works in compliance with the requirements. This troubleshooting often requires student to re-design and re-build some workstations until the model is functioning as desired. The fourth and last phase involves the project report.

In particular, the Process Automation course motivated this research because student design, develop, evaluate, integrate and manage projects that are used in real-life applications. However the attention is in students' preference and choices as they develop a diverse range of solution with different complexity that can now be quantified. Section 3.3.1 describe complexity measures.

3.3.1 Stage 2: Identify, Adapt and Develop Complexity Metrics for Process Automation

From an in depth review of complexity metrics, four metrics: (1) Complexity of Partition, (2) Lines of code, (3) Cyclomatic Complexity Metric and (4) Product Complexity, were identified to become the baseline and inspiration to create and adapt existing metrics (see Table 3.2 below). These metrics were selected due to the relevance of *interactions*, *size* and *functionalit*y in the proper assessment of complexity as a well as their feasible application to process automation. Literature shows that interactions have structural and behavioral impact (Blay-Fornarino, Charfi, Emsellem, Pinna-Dery, & Riveill, 2004), size is a basic attribute of software products (Bajwa, Gencel & Abrahamsson, 2014), and according to Faulconbridge and Ryan (2003), complex technical projects, can only be manage effectively when functional requirements are analyzed.

Table 3.2: Adaptation of Complexity Metrics

Original Metric	**Proposed Metric**	**Emphasis**	**Notation**	**Formulation**	**Data**
Complexity of Partition (C) (Keating, 2000)	Visual Component Interaction (VCI)	Components and their physical interactions	M- unique components I-interactions	$VCI = M^2 + I^2$	Ladder Logic Visual Component interaction diagram
	Software Component Interaction (SCI)	Components and their interactions through the program.	M- unique components I-interactions	$C = M^2 + I^2$	Ladder Logic Network Diagram
	Software Component Interaction with stages (SCIS)	Components and their interactions through the program with stages considered	M- unique components I-interactions	$C = M^2 + I^2$	Drawing with identified components Pictures and Videos
Lines of Code	Lines of Code (LOC)	Size of the program based	L - last line identification	$LOC = L - n$	Ladder Logic

Original Metric	Proposed Metric	Emphasis	Notation	Formulation	Data
(Basili and Perricone, 1984)		on the number of lines	number for the ladder logic n – number of blank lines (output NOP)		
Cyclomatic Complexity Metric (V (G)) (McCabe, 1976)	Cyclomatic Complexity Metric (V(G))	Number of linearly independent paths in a program	n – number of stages in the program e –number of lines (interactions) joining each stage p –number of initial stages	$V(G) = e - n + p$	Grafset of the stages Ladder Logic
Product Complexity (PC) (Bashir & Thomson, 1999)	Process Hierarchical Functionality (PHF)	Process functions decomposed in multiple levels	F- number of functions at each level l- number of levels {1,2,…n} k_l- weight for level l, where k_1 = 1, k_2 = 2, k_n=n i- total process functions	$PHF = (\sum_{j=1}^{i} F_l) * k_l$	Pictures and Videos Project Description
	Station Hierarchical Functionality (SHF)	Stations functions decomposed in multiple levels	F- number of functions at each level l- number of levels {1,2,…n} k_l- weight for level l, where k_1 = 1, k_2 = 2, k_n=n i- total station functions	$SHF_i = (\sum_{j=1}^{i} F_l) * k_l$ $SHF = \sum_{i=0}^{n} SHF_i$ $SHF = \prod_{i=0}^{n} (SHF_i)$ $SHF = Max\{SHF_i\}$	Pictures and Videos Project Description

Table 3.2 summarizes these metrics from literature along with their adaptation that in some cases resulted in the development of more than one metric. A brief description of the emphasis of each metric is provided along with its notation and formulation. The last column specifies data available from the process automation course to implement the metric. Overall, a total of seven metrics were generated to assess complexity of process automation projects. Each one of these metrics is discussed in the following sections with the development of implementation methods (Colón et al., 2013; Soto et al., 2015; Martínez et al., 2015; Martínez et al., 2016; Collado et al., 2016; Jusino et al., 2016).

Student project reports, from the process automation course, provided the necessary data to implement metrics. Besides the general documentation (pictures, videos, drawings and descriptions) that explains the project design, the developed program or software must be considered. The program is assessed in two forms, through Ladder Logic and Grafcets.

Four of the seven metrics, required analyzing the ladder logic in order to come up with a result. Ladder Logic is the most popular programming language used to program process automation that is mostly implemented with programmable logic controllers (PLCs). It is a very visual graphical language which structure was designed to mimic the electrical schematic of relays. Some basic functions as shown in Figure 3.2 below are: Examine On (X0) and Examine Off (X1), located at the left side of the line of code, and the Output(s) (Y0, C5), at the right side of the line of code. Examine On is when the input element allows the flow of current. At the contrary, Examine Off is when the input element does not allow the flow of current. Output turn on or off an output element. The performance can be seen as responding to messages (probably events) sent by some component instances to other component instances (Blay-Fornarino, Charfi, Emsellem, Pinna-Dery, & Riveill, 2004). Being the basic structure of a ladder logic, there is the option of organizing groups of lines of codes into Stages (See example in the Appendix A) where only the code on active stages will be executed.

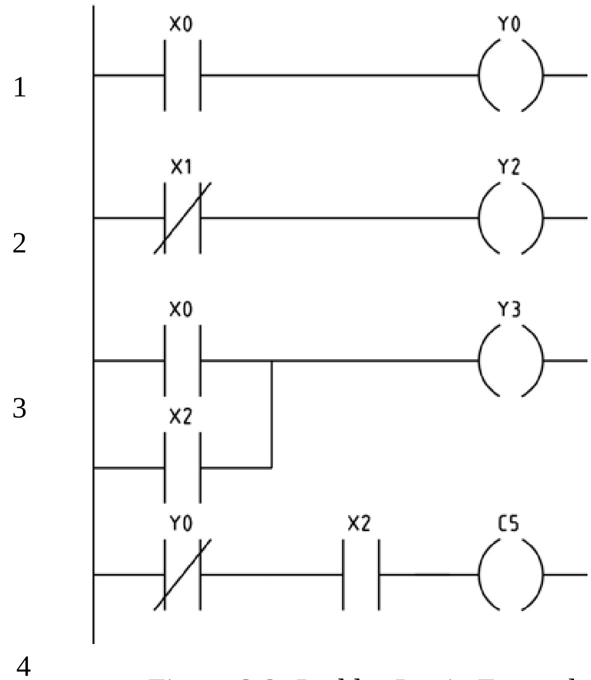


Figure 3.2: Ladder Logic Example

Grafcets (Figure 3.3) are used as a summarized and visual representation of Ladder Logic codes with stages. It is used for the Cyclomatic Complexity metric in order to determine the number of possible routes or roads the program has to complete the process.

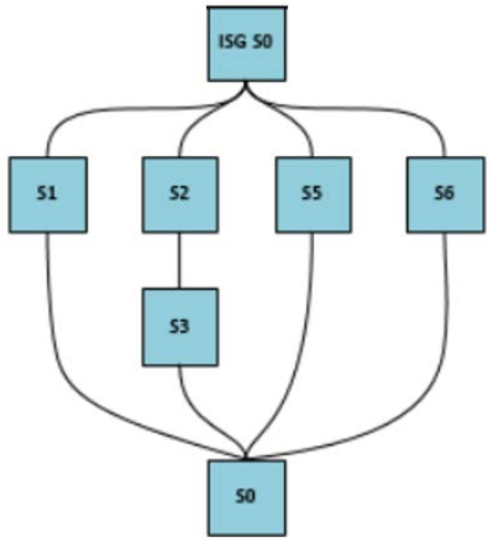


Figure 3.3: Grafcet Example

3.3.1.1 Visual Component Interaction (VCI)

Inspired by Keating's (2000) Complexity of Partition (C) metric, the Visual Component Interaction (VCI) metric is proposed to study physical interactions (visual) between components in process automation projects. The original metric was developed to predict design quality early in the design cycle by assessing the complexity of a design partitioning. However, VCI is developed to measure overall design complexity based on the component interaction that can be observed physically in the process, independently of the program or software (Colón, Collet, Cruz, Del Pilar, & Martinez, 2013; Soto, Rosado & Medina, 2015; Collado, Medina & Soto, 2016).

Besides Keating (2000) formulation to relate components and interactions mathematically, this research contribution includes the development of an implementation method to comply with VCI's intended objective. A detailed description of the implementation method proposed for VCI is provided below:

1. Following the notation proposed in Figure 3.4, a VCI component-interaction diagram (see example in Figure 3.5) is necessary to determine the number of components and interactions.

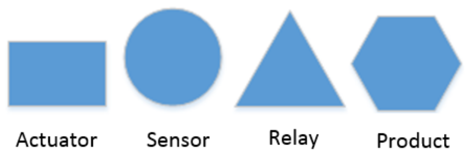


Figure 3.4: VCI Figures Legend

i. Identify relays with a triangle and place them in a column to the far left.
ii. Identify actuators with a square and place them in a column right to the triangles. Note: Actuators include motors, pistons and valves. Lights will not be considered in this interaction, they will be placed in another column to the far right.
iii. Identify raw materials, finished products and/or packaging products with a hexagon and place them in a column to the right of the actuators' column.
iv. Identify sensors with a circle and place them in a column to the right of the hexagons. Note: If the sensor function is to reset the complete process, then place its circle in the far left before the relays column (See X0 in Figure 3.5).
v. Identify actuators that interact with relays, by tracing a line between them.
vi. Identify actuators that interact with other actuators by tracing a line between them.
vii. Identify actuators that interact with products by tracing a line between them.
viii. Identify sensors that interact with products by tracing a line between them.
ix. Identify sensors that interact with actuator by tracing a line between them.

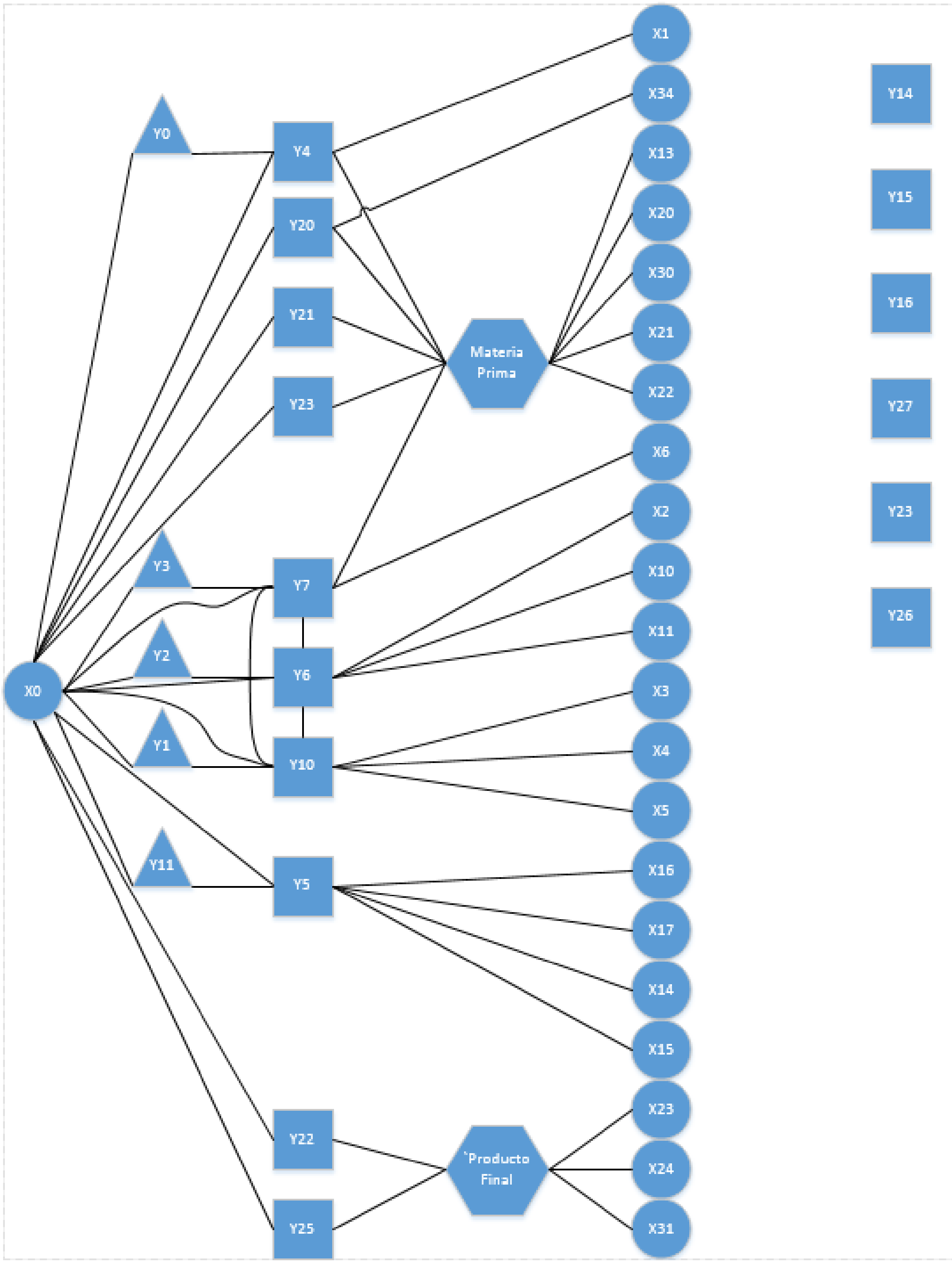


Figure 3.5: VCI Component-Interaction Diagram Example

2. After the VCI diagram is done, count all the figures. The result will provide the number of unique components, M.

3. Count the amount of interactions represented as lines between components in the VCI diagram. The result will provide the number of interactions, I.

Finally, calculate the VCI metric using the C and I values obtained from steps 2 and 3 using the following equation:

$$VCI = M^2 + I^2 \qquad (3.1)$$

The value for the example presented in the figure is VCI= $45^2+49^2= 4426$.

3.3.1.2 Software Component Interaction (SCI)

Similar to VCI, the Software Component Interaction (SCI) metric uses Keating's (2000) formulation to relate components and interactions. SCI is intended to measure the complexity of process automation components and interactions through the software, particularly, the ladder logic (Colón et al., 2013; Soto et al., 2015; Martínez et al., 2015 and Martínez et al., 2016). In this particular context, components are obtained from condition and output statement in the ladder logic. In the program, Xs and Ys represent in the software sensors and actuators (including relays), respectively. Contrary to VCI, raw materials, finished products and/or packaging products are not considered since they are not represented in the code. Meanwhile, other components are added. These include internal variables (Cs) used to facilitate the programming when needed, counters (CTs), and timers (Ts).

Interactions are determined as a result of various components (Xs, Ys, Cs, CTs, and Ts) joining together as conditions for the output statements (set or reset Ys, Cs, CTs and/or Ts). Contrary to VCI where a diagram of components and interactions could be developed right away from observing the process automation project, implementing SCI is challenging. While components can be easily determined by counting the number of unique variables (Xs, Ys, Cs, CTs, and Ts), interactions require evaluating the code in detail. As part of this research contribution a procedure is developed to obtain the number of interactions. The procedure involves: (1) making components-interaction diagrams for each line of code, (2) eliminating redundant diagrams and (3) overlapping diagrams when interactions are reduced. The whole procedure to implement the metric is provided as follows:

1. Create a table, tabulating unique components within the ladder logic. The amount of all unique components is M.
2. Follow the following steps to obtain the number the interactions, I:
   I. Make components-interaction diagrams for each line of code:
      i. Identify components in a line of code with a circle.
      ii. Connect components in the conditions statement with a line between all pair-wise comparisons and draw a circle/oval to surround them (after all the lines are made).

      Example: To represent the condition statement shown in Figure 3.6 of the line of code 4, the component-interaction diagram in Figure 3.7 was drawn.

Figure 3.6: Line of Code Example

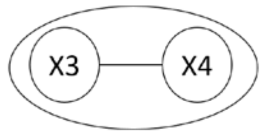


Figure 3.7: Network Diagram Example

iii. Connect output variables to the condition statement component-interaction diagram previously done. Note: JMP (e.g. JMP S1 in the example) commands are considered as a connection the specified stage, therefore, the actions in that specified stage are related to the conditions before the JMP command.

Example: Figure 3.8 shows how outputs should be included.

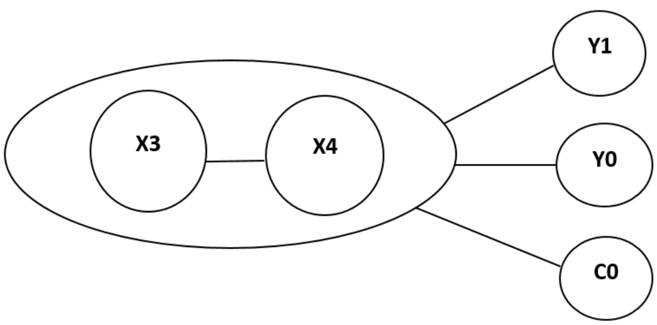


Figure 3.8: Line of Code Network Diagram Example

iv. Repeat steps i though iii until component-interaction diagrams are performed for all the lines of code.

II. Analyze and compare diagrams completed in order to identify and eliminate redundant diagrams.

III. Analyze and compare remaining diagrams after eliminating redundancy in II to identify opportunities for overlap. Overlapping is necessity only if interactions are reduced. For example, overlapping is necessary when component-interaction diagrams coincide in the conditions part while happing different output statement.

IV. Once elimination and overlapping is completed, count the number of lines that result from the component-interaction diagrams. The result is the number of interactions, I.

3. With the number of components (M) and interactions (I) obtained in the previous steps calculate the metric:

$$SCI = M^2 + I^2 \quad (3.2)$$

This procedure has been proven to be an equivalent simplification to the challenge of completing the all components and all interactions diagram (Appendix B shows the procedure completed for a particular project by Martínez et al., 2015).

3.3.1.3 Software Component Interaction with Stages (SCIS)

The Software Component Interaction with Stages (SCIS) metric – Colón et al., (2013), Soto et al., (2015), Martínez et al., (2015), and Martínez et al., (2016) is proposed as a modification of the SCI metric. As shown before, the SCI do not consider stages (S) as a component. For the SCIS, stages will be considered and included as a component. This change impacts both, the number of components and the number of interactions. The same procedure as SCI is followed; an example of how the stages are considered in the component-interactions diagrams is provided. Figure 3.9 shows the component-interaction diagram for line of code 4 in Figure 3.6. In comparison to Figure 3.8 where stages were not considered for SCI, for SCIS S0 becomes a condition since stage 0 (S0) must be active for the program to consider line 4.

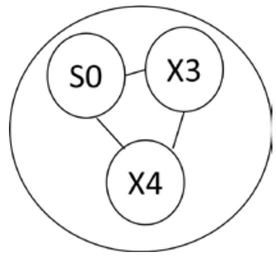


Figure 3.9: Network Diagram Example

Following, the output (S1) is added as shown in Figure 3.10. In comparison with Figure 3.7, for SCIS the stage becomes the output and Y1, Y0 and C0 will be considered in a separate diagram for line 6 where S1 is the condition.

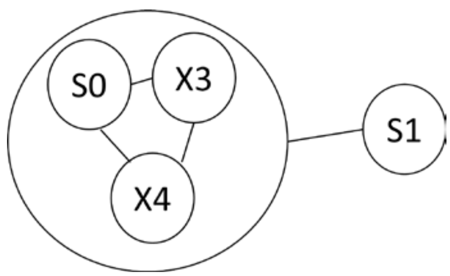


Figure 3.10: Line of Code Network Diagram Example

3.3.1.4 Lines of Code (LOC)

Basili and Perricone (1984) discuss one of the simplest software metrics, Lines of Code (LOC) LOC, which measures the number of lines (statements) in a program, was originally used in software projects coded in FORTRAN. In the proposed work LOC is used to measure the lines of code of ladder logic (Colón et al., 2013). The steps to implement this metric are provided.

1. Identify the line identification number of the "End" statement of the ladder logic. In Figure 3.11 this corresponds to 111.

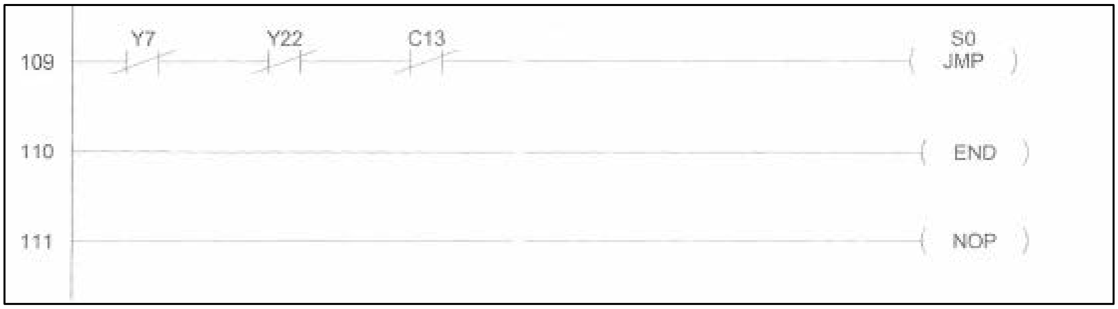


Figure 3.11: Ladder Logic Identification Number

2. Count number of blank lines (output NOP) in the ladder logic.
3. Calculate LOC metric with the following equation:

$$LOC = L - n \tag{3.3}$$

Where,

L - last line identification number for the ladder logic

n – number of blank lines (output NOP)

3.3.1.5 Cyclomatic Complexity (V(G))

McCabe (1976) proposes the Cyclomatic Complexity (V(G)) metric as a more robust assessment of software complexity. V(G) measures the number of linearly independent paths in a program by identifying the number of vertices (n), edges (e) and connected components (p). To implement V(G) in the context of process automation, the use of Grafcets (Figure 3.12) is proposed since it provides a visual representation of the ladder logic code (Colón, Collet, Cruz, Del Pilar, & Martinez, 2013; Soto, Rosado & Medina, 2015; Collado, Medina & Soto, 2016). Accordingly, the variables are considered to become the stages (n), initial stages (p) and jumps (e). The steps to implement Cyclomatic Complexity metric are provided as follows.

1. Draw a Grafcet of the ladder logic. Note that the boxes represent the different stages that are interconnected (one box per stage). If a stage is specified in a command of JMP

(e.g. JMP S0) that means it should follow the particular stage where the command is provided (like S2 and S3 in Figure 3.12). If a stage is specified in a command of Set (e.g. Set S0, Rst S0), then it should be connected to the stage where the command is provided while it does not represent the end that stage (like S2 and S5 in the example).

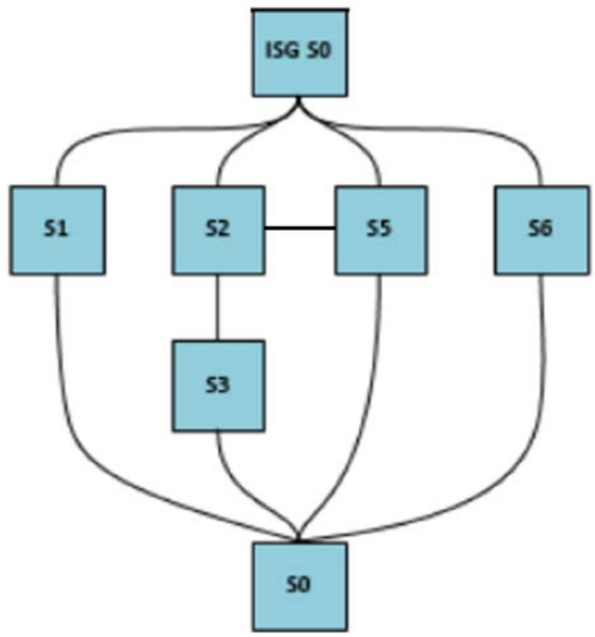


Figure 3.12: Ladder Logic

2. Identify the number of stages.
3. Identify the number of lines joining each stage.
4. Identify number of initial stages identified (e.g. ISG S0).
5. Calculate the V(G) metric with the following equation:

$$V(G) = e - n + p \tag{3.4}$$

Where,

n – number of stages in the program

e –number of lines (interactions) joining each stage

p –number of initial stages

From the example in Figure 3.12, the Cyclomatic Complexity is provided below.

$$V(G) = e - n + p = 10 - 6 + 1 = 5$$

3.3.1.6 Process Hierarchical Functionality (PHF)

Inspired with Bashir and Thomson's (1999) product complexity (PC) metric, the Process Hierarchical Functionality (PHF) metric is intended to assess specific functions of completed process automation projects (Colón, Collet, Cruz, Del Pilar, & Martinez, 2013; Soto, Rosado & Medina, 2015; Jusino, Medina, and Soto, 2016). In contrast, PC was developed for products to be designed in order to estimate design effort. Still, both metrics coincide in the need to define a hierarchical decomposition of functions into different levels. The steps to implement PHF are provided below.

1. Learn about the process automation project through the project documentation, photos and videos.
2. Identify functions from the generic functional decomposition provided in Figure 3.13.

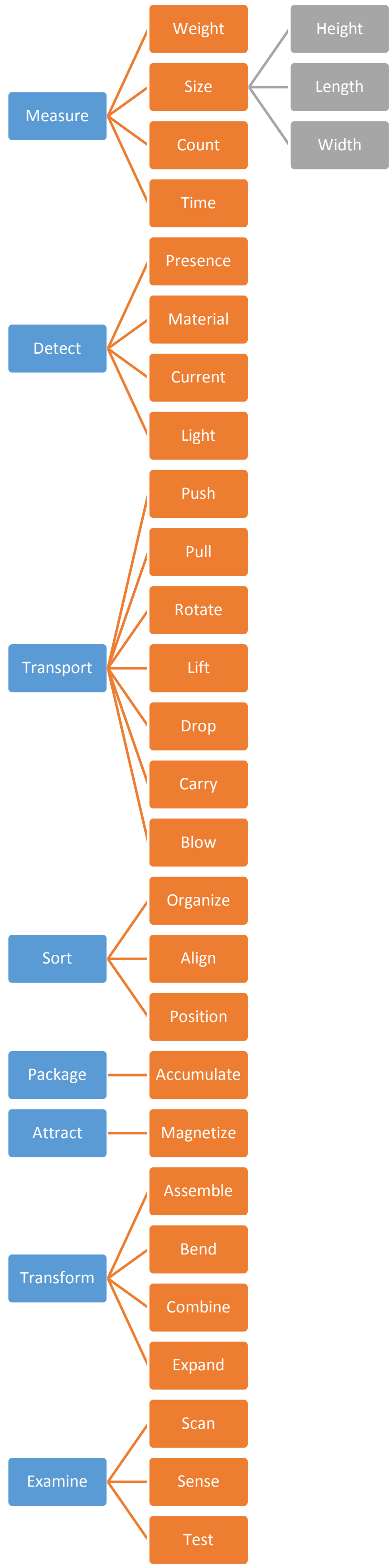


Figure 3.13: Generic Functional Decomposition for Process Automation (Jusino, Medina

and Soto, 2016)

3. Make a customized functional decomposition diagram with the identified functions as shown in Figure 3.14.

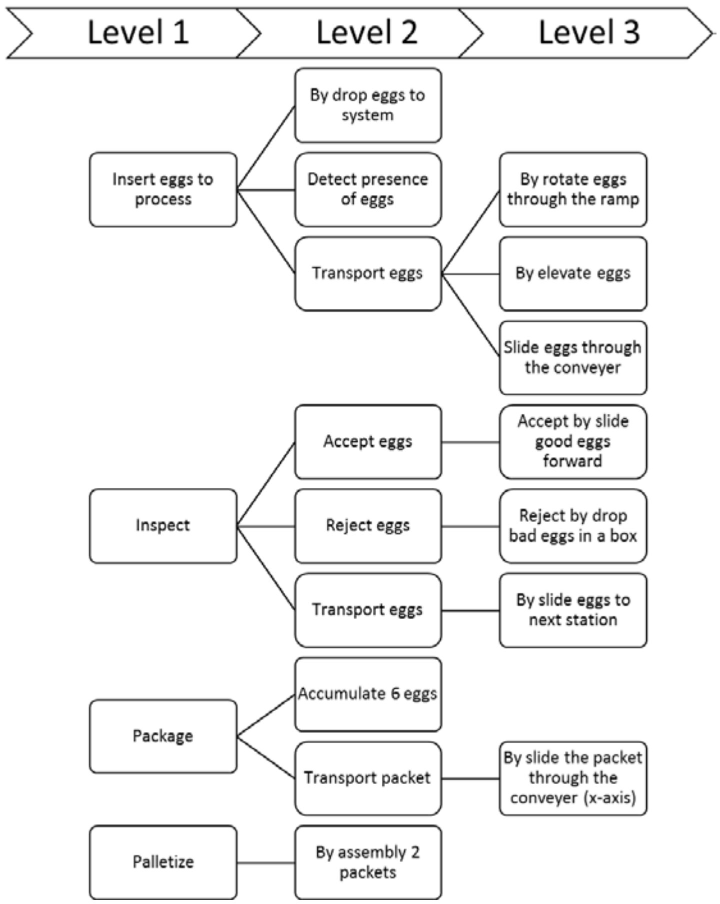


Figure 3.14: Example of a Functional Decomposition Diagram (Jusino, Medina and Soto, 2016)

4. Calculate the PHF metric with the following equation:

$$PHF = \left(\sum_{j=1}^{i} F_l\right) * k_l \tag{3.5}$$

Where,

F_l is the number of functions at level l,

l is the number of levels {1,2,…n}

k_l is the weight for level l, where $k_1 = 1$, $k_2 = 2$, $k_l = l$

i= number of functions

Below is an example of the functional decomposition diagram. The result for example above is: PHF = 4*1+ 9*2+ 7*3 = 43

3.3.1.7 Station Hierarchical Functionality (SHF)

Similar to PHF, the Station Hierarchical Functionality (SHF) measures the complexity in relation to station functions instead of the whole process (Colón, Collet, Cruz, Del Pilar, & Martinez, 2013; Soto, Rosado & Medina, 2015; Jusino, Medina & Soto, 2016). The same procedure is followed but for each station independently. The generic functional decomposition in Figure 3.10 is also used to determine station functions. SHF equation is defined by:

$$SHF_i = \left(\sum_{j=1}^{i} F_l\right) * k_l \quad (3.6)$$

Where,

F_l is the number of functions at level l,

l is the number of levels {1,2,…n}

k_l is the weight for level l, where $k_1 = 1$, $k_2 = 2$, $k_l=l$

i= number of stations

After the complexity is assessed for each station, an overall complexity is obtained from (1) identifying the station with maximum complexity, (2) obtaining the summation of stations complexities and (3) calculating the product (multiplication) among stations complexities. Equations are provided below.

$$\text{Max\_}SHF = Max\{SHF_i\} \quad (3.7)$$

$$Sum\_SHF = \sum_{i=0}^{n} SHF_i \quad (3.8)$$

$$Product\_SHF = \prod_{i=0}^{n}(S\text{HF}) \quad (3.9)$$

Following an example is provided with Figure 3.15 providing the result for the functional decomposition and the different complexity calculations provided.

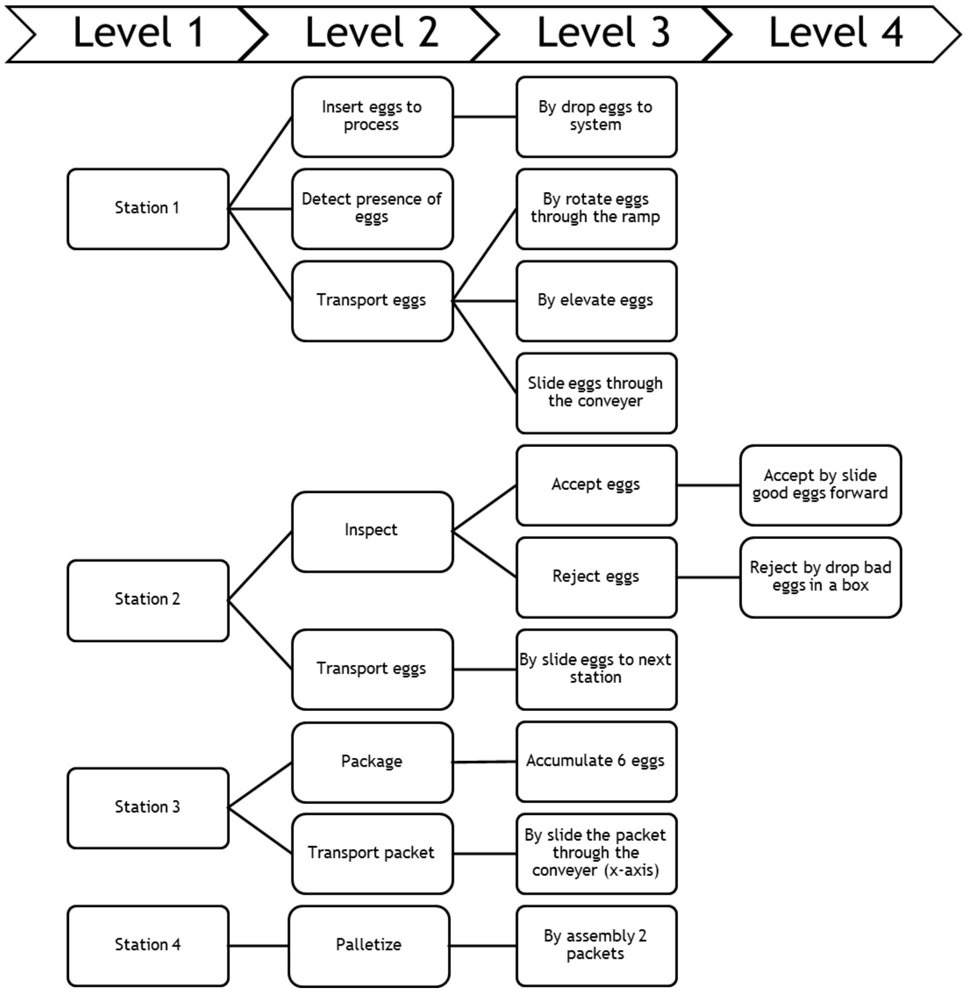


Figure 3.15: SC Functional Decomposition Diagram (Jusino, Medina and Soto, 2016)

$$SHF_i = \sum_{j=1}^{i} F_l * k_l$$

SHF_1 = 1*1 + 3*2 +3*4 = 19

SHF_2 = 1*1 + 2*2+ 3*3+ 2*4 = 22

SHF_3 = 1*1 + 2*2+ 2*3 = 11

SHF_4 = 1 *1 + 1*2 +1*3 = 6

$$Max\_SHF = Max\{SHF_i\} = Max\{19,\ 22,\ 11,\ 6\} = 22$$

$$Sum\_SHF = \sum_{i=0}^{n} \text{SHF} = SHF_1 + SHF_2 + SHF_3 + SHF_4 = 19 + 22 + 11 + 6 = 58$$

$$Product\_SHF = \prod_{i=0}^{n}(SHF_i) = SHF_1 * SHF_2 * SHF_3 * SHF_4 =$$

$$19 * 22 * 11 * 6 = 27{,}588$$

3.3.2 Analysis of Complexity Metrics for Process Automation

Once the complexity metrics were implemented, each complexity metric result was analyzed using descriptive statistics, correlation analysis Kruskal-Wallis rank sum test and Pairwise comparison using Dunn's test for multiple comparison of independent samples. Also, these test, complexity metrics result were normalized by dividing $\frac{y}{\max(y)}$ in order to build a *radar char*, commonly known as a *spider graph*, used to display and analyze the metrics altogether.

3.3.3 Stage 3: Development of Student Factor Survey

A student factor survey was developed with the objective to collect information from continuous and categorical variables related to students and design teams (age, gender, knowledge, abilities, individual contribution, difficulty, team dynamic and individual and team performance) to correlate these factors with the complexity. The survey is a 24 closed-ended questions (see Table 3.3 below for reference). The survey presented here is a modified version to Colón, Collet, Cruz, Del Pilar, and Martinez (2013).

Table 3.3: Student Factor Survey Questions

Category	Student Factor Survey Questions
Profile	1.*Género*/ Gender
	2. *Edad al tomar el curso ININ 4057* / Age at the time enrolled at the course ININ4057
	3. *Aproximadamente, ¿Cuántos créditos tenía usted matriculado cuando tomo ININ 4057?* / Approximately, how many credit did you had enrolled when you took ININ 4057?

Category	Student Factor Survey Questions
Performance	23. *¿Actualmente, cuál es su promedio de concentración (Ing. Industrial)?* / Currently, what is your major (Industrial Engineering) GPA?
	24. *¿Actualmente, cuál es su promedio general?* / Currently, what is your general GPA?
	25. *Favor de indicar que nota (A,B,C,D) obtuvo en cada una de las siguientes clases*/ Please indicate what grade (A,B,C,D) you obtained in the following courses: *Algoritmos y Programación de Comp*./ Computer programming and Algorithm (INGE 3016) *Circuito* / Circuit (INEL 4075) *Electrónica* / Electronic (INEL 4076) *Laboratorio de Electrónica* / Electronic Laboratory (INEL 4077) *Proceso de Manufactura* / Manufacturing Process (INME 4055) *Laboratorio de Proceso de Manufactura* / Manufacturing Process Laboratory (INME 4056) *Proceso Automatizado* / Process Automation (ININ 4057)
Knowledge	4. *Seleccione todas las prácticas obtenidas antes del curso, relacionadas a la Ingeniería Industrial que apliquen.* / Select all the experience obtained before the course related to Industrial Engineering
	5. *¿Antes de tomar el curso ININ 5057 estuvo expuesto a procesos automatizados relevantes al curso mencionado?* / Before taking the course ININ 4057, were you exposed to automated process relevant to the course? 6. *Si contestó sí en la pregunta anterior, explique*/ If you answered yes to the previous question, explain.
	7. *Califique su conocimiento sobre programación antes de comenzar el curso ININ 4057.* / Rate your knowledge regarding programin before you took the course. Scale 1 (Poor) to 7 (Excellent)
Perceived Abilities/ Skills	*9. Califique su destreza al programar su maqueta.* / Rate your ability to program the small scale model. Scale 1 (Very Poor) to 7 (Excellent)
	10. *Califique su destreza al construir los circuitos de la maqueta.* / Rate your ability to build the circuit of the small scale model. Scale 1 (Very Poor) to 7 (Excellent)
Individual contribution	8. *Califique cuán motivado(a) estuvo para tomar el curso de ININ 4057.* / Rate how motivated you were to take the course ININ 4057. Scale 1 (Not Motivated) to 7 (Extremely Motivated)
	17. *Califique su desempeño en la maqueta.* / Rate you performance in the small scale model. Scale 1 (Very Poor) to 7 (Excellent)
	18. *En comparación con sus compañeros de trabajo, ¿cuánto usted trabajo?* / In comparison with you teammates, how much did you worked? Scale 1 (Substantially less) to 7 (Substantially more)
	13. *Favor de evaluar su contribución en cada fase del proyecto.* / Please evaluate your contribution at each phase of the proyect. Scale 1 (Very Poor) to 7 (Excellent)
	14. *Favor de proveer un estimado de cuántas horas le dedicó al proyecto.* / Please provide an estimate on how many hours did you spend on the project.
	15. *¿Cuán confiado se siente en su estimado de las horas trabajadas?* / How confident do you feel in your time estimate?
Difficulty/ Complexity	11. *Favor proveer su percepción sobre la complejidad de cada parte de la maqueta: Diseno, Estructura de la Maqueta, Circuitos, Programación, "Troubleshooting" y HMI.* / Please provide your perception regarding each phase of the small scale model: Design, Structure, Circuit, Programming, *Troubleshooting and HMI.* Scale 1 (Extremely Simple) to 7 (Extremely Complex)
	12. *Favor proveer su percepción sobre la complejidad en la implementación de cada componente.* / Please provide your perception regarding the complexiy of each component. Scale 1 (Extremely Simple) to 7 (Extremely Complex)
Team Dynamic	16. *Favor de proveer la composición (cantidad) de miembros de su proyecto incluyéndose usted.* / Please provide the composition (Qty.) of your group memebers, including yourself.
	19. *¿Cuál fue el desempeño de su equipo?* / What was the group performace?
	20. *Califique la frecuencia de comunicación con su equipo de trabajo para el proyecto.* / Rate the communication frequency of your project team memebers. Scale 1 (Less Frequently) to 7 (Very Frequent)
	21. *¿Cómo categoriza la comunicación con su grupo?* / How do you categorizes the communication with your group?

Category	Student Factor Survey Questions
	22. *¿Cómo su grupo tomó las decisiones la mayoría del tiempo?* / How did your group made the decisión the mayority of time?

3.3.3.1 Student Factor Survey

Once the survey was implemented, variables were group into categories as observed in Table 3.4. to be used as independent variable for the prediction model discussed at Section 3.4.1

Table 3.4 Student Factor Survey Variables

Student Factor Survey Questions	Variables	Category	Connotation
1. Gender	Gender	Gender	0=Female, 1= Male
2. Age at the time enrolled at the course ININ4057	Age	Age	Age in Years
3. Approximately, how many credit did you had enrolled when you took ININ 4057?	Academic_Load	Academic Load	Academic credits
4. Select all the experience obtained before the course related to Industrial Engineering	Knowledge_Work	Knowledge	0= No Knowledge, 1= Knowledge
	Knowledge_Project		
	Knowledge_Internship		
	Knowledge_Coop		
5. Before taking the course ININ 4057, were you exposed to automated process relevant to the course?	Knowledge_Automation		
7. Rate your knowledge regarding programin before you took the course.	Knowledge_Programming		
9. Rate your ability to program the small scale model.	PerHabilities_Programming	Perceived Abilities/ Skills	Perceived abilities rating Scale 1 (Very Poor) to 7 (Excellent)
10. Rate your ability to build the circuit of the small scale model.	PerAbilities_Circuit		
11. Please provide your perception regarding each phase of the small scale model.	PerComplexity_Design	Perceived Difficulty/ Complexity	Perceived difficulty rating Scale 1 (Extremely Simple) to 7 (Extremely Complex)
	PerComplexity_Structure		
	PerComplexity_PLC		
	PerComplexity_Programming		
	PerComplexity_Troubleshooting		
	PerComplexity_HMI		
12. Please provide your perception regarding the complexiy of each component.	PerComplexity_Motors		
	PerComplexity_Neumuatic		
	PerComplexity_Sensors		
	PerComplexity_Relays		
8. Rate how motivated you were to take the course ININ 4057.	Individual_Motivation	Individual Contribution	Student Motivation on a Scale 1 (Not

Student Factor Survey Questions	Variables	Category	Connotation
			Motivated) to 7 (Extremely Motivated)
13. Please evaluate your contribution at each phase of the proyect.	Individual_ContributionDesign		Student Design Contribution Scale 1 (Very Poor) to 7 (Excellent)
	Individual_ContributionStructure		Student Structure Contribution Scale 1 (Very Poor) to 7 (Excellent)
	Individual_ContributionPLC		Student PLC Contribution Scale 1 (Very Poor) to 7 (Excellent)
	Individual_ContributionProgramming		Student Programming Contribution Scale 1 (Very Poor) to 7 (Excellent)
	Individual_ContributionTroubleshooting		Student Troubleshooting Contribution Scale 1 (Very Poor) to 7 (Excellent)
	Individual_ContributionHMI		Student Human Machine Interface Contribution Scale 1 (Very Poor) to 7 (Excellent)
14. Please provide an estimate on how many hours did you spend on the proyect.	Individual_ProjectHrs		Estimated hours spend in project
16. Please provide the composition (Qty.) of your group memebers, including yourself.	Group_Qty	Team Dynamic	Group Qty.
	Group_Female%		Group Female %Group
17. Rate you performance in the small scale model.	Individual_PerPerformance		Student Perceived Self Performance Scale 1 (Very Poor) to 7 (Excellent)
18. In comparison with you teammates, how much did you worked?	Individual_ComparedContribution		Student contribution among group members Scale 1 (Substantially less) to 7 (Substantially more)
19. What was the group performace?	Group_MemberEngage%		Group performance
20. Rate the communication frequency of your project team memebers.	Group_CommunicationFreq		Communication Frequency Scale 1 (Less Frequently) to 7 (Very Frequent)
	Goup_DecisionsUnanimity		0=No, 1=Yes
	Goup_DecisionsAuthority		

Student Factor Survey Questions	Variables	Category	Connotation
21. How do you categorizes the communication with your group?	Goup_DecisionsMinority		
	Goup_DecisionsMayority		
	Goup_DecisionsConsensus		
23. Currently, what is your major (Industrial Engineering) GPA?	Performance_ININ3.51	Performance	0=No, 1=Yes
	Performance_ININ3.01		0=No, 1=Yes
	Performance_ININ2.51		0=No, 1=Yes
24. Currently, what is your general GPA?	Performance_General3.51		0=No, 1=Yes
	Performance_General3.01		0=No, 1=Yes
	Performance_General2.51		0=No, 1=Yes
25. Please indicate what grade (A,B,C,D) you obtained in the following courses: Computer programming and Algorithm (INGE 3016) Circuit (INEL 4075) Electronic (INEL 4076) Electronic Lab. (INEL 4077) Mfg. Process (INME 4055) Mfg. Process Lab. (INME 4056) Process Automation (ININ 4057)	Performance_INEL4075		Grades where A=4, B=3, C=2
	Performance_INEL4076		
	Performance_INEL4077		
	Performance_INME4055		
	Performance_INME4056		
	Performance_ININ4057		
	Performance_INGE3016		

All students agreed that class reports could be used in this research. Additional variables were added and are summarized in Table 3.5.

Table 3.5: Additional Student's Variables

Source	Variables	Category	Connotation
Performance Measures from Process Automation (ININ 4057)	Performance_GrpLabs	Performance	Course Grade
	Performance_GrpDesign		
	Performance_GrpStructure		
	Performance_GrpDemo		
	Perfornance_IndExams		
	Perr Eval (max 30)		Peer Evaluation
	Absence		Count of Absence to the ININ 4057 course
	Lateness		Count of Lateness to the ININ 4057 course

Also, to consider group interaction, not only individual's characteristics, some variables were selected from Table 3.4 to represent group's maximum, minimum and median value. For instance, instead of analyzing student age, new variables, now represent group's maximum, minimum and median age. Accordingly, this calculation was done for all variables in Table 3.6. This was done to use these new variables in the Expanded Model II as explained in Section 3.4.1.

Table 3.6: Additional Group Variables

Age
Knowledge_Work
Knowledge_Project
Knowledge_Internship
Knowledge_Coop
Knowledge_Automation
Knowledge_Programming
Individual_Motivation
PerHabilities_Programming
PerHabilities_Circuit
PerComplexity_Design
PerComplexity_Structure
PerComplexity_PLC
PerComplexity_Programming
PerComplexity_Troubleshooting
PerComplexity_HMI
PerComplexity_Motors
PerComplexity_Neumuatic
PerComplexity_Sensors
PerComplexity_Relays
Individual_ContributionDesign
Individual_ContributionStructure
Individual_ContributionPLC
Individual_ContributionProgramming
Individual_ContributionTroubleshooting
Individual_ContributionHMI
Individual_ProjectHrs
Individual_PerPerformance
Individual_ComparedContribution
Performance_ININ3.51
Performance_ININ3.01
Performance_General3.51
Performance_General3.01
Performance_INEL4075
Performance_INEL4076
Performance_INEL4077
Performance_INME4055
Performance_INME4056

Performance_ININ4057
Performance_INGE3016
Perfornance_IndExams
Absense
Lateness

Next section, discusses how data was analyzed to answer last research question and to generate prediction models.

3.4 Stage 4: Data Analysis

To study the relationship of complexity with team characteristics and project outcomes, a complexity prediction model was generated integrating data obtained from stage 1, 2 and 3. Section 3.4.1 shows how prediction models were built and selected for each one of the complexity metrics.

3.4.1 Prediction Models

For the prediction models, three regression models were constructed for each response. See Figure 3.16. The first model is called “General Model”. This model used as predictor 58 variables as specified in Table 3.4 and Table 3.5. The second model is called “Reduced Model I”. This model used as predictors the performance variables only. Last, and third model is called “Expanded Model II”. It used as predictor 173 variables as specified in Tables 3.4, 3.5 and 3.6. These 173 variables were obtained from created new variables that represents group’s maximum minimum and median value.

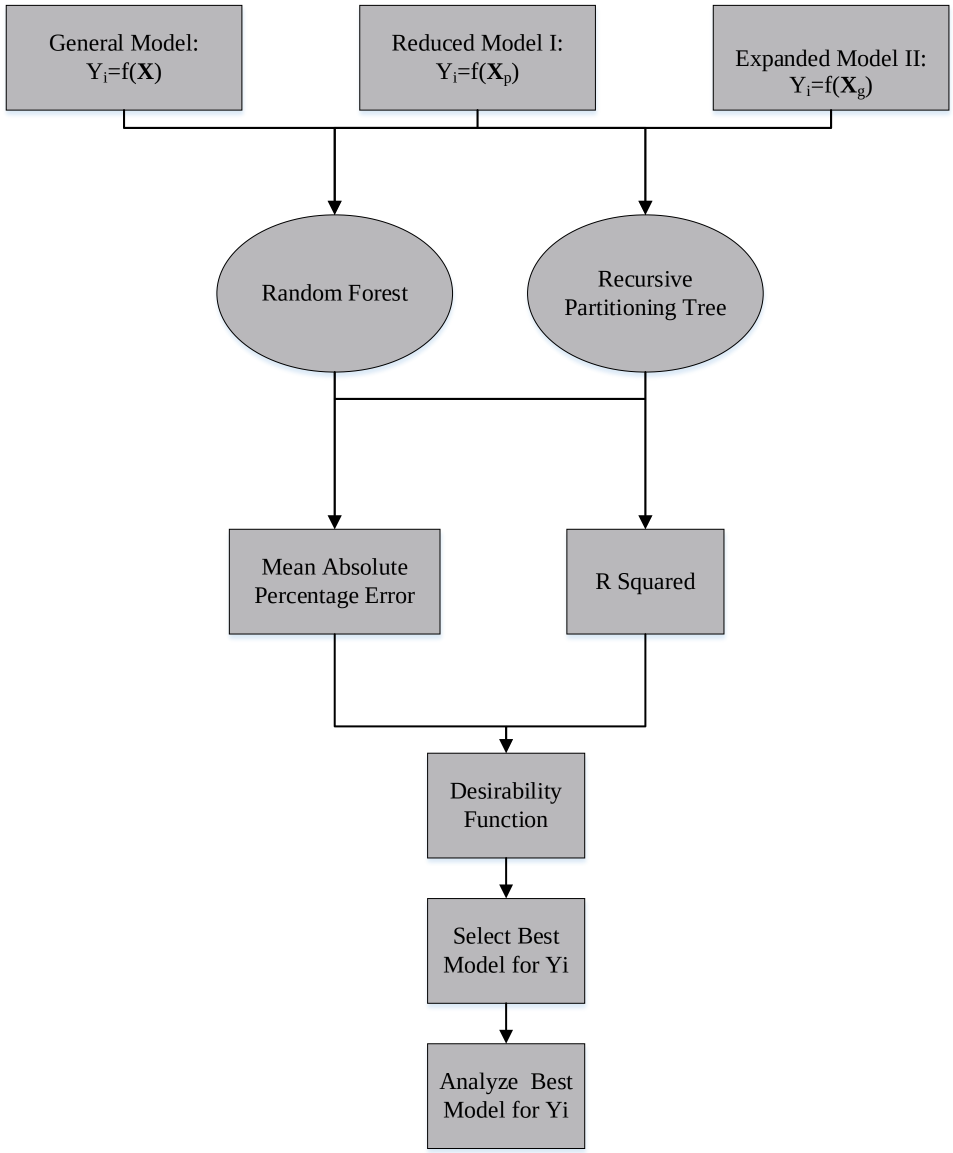


Figure 3.16: Prediction Models Methodology

Two regression method were used, random forests and decision tree, both explained in more detail in Section 3.4.4.1 and 3.4.4.2, respectively. In order to select only one best prediction model per response, two performance measures were calculated.

R^2, also called coefficient of determination, represents the proportion of the variance in the response that is explained by the model (basically how close the data is to the fitted regression line).

These measures explains the percentage of the variability of the response data around its mean. For both methods, R^2 was calculated using Equation 3.10:

$$R^2 = 1 - \frac{\sum(actual - predicted)^2}{\sum(actual - mean(actual))^2}\,. \tag{3.10}$$

The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD), is a measure of prediction accuracy of a forecasting method in statistics, express as a percentage of the error. MAPE was implemented as specified in Equation 3.11:

$$MAPE = 100 * \frac{\sum(\frac{abs(actual - predicted)}{actual})}{length(predicted)}\,. \tag{3.11}$$

These measures were used to build a desirability function (dF) as specified in Equation 3.12:

$$dF = \frac{R^2}{\max(R^2)} + \left(1 - \frac{MAPE}{Max(MAPE)}\right) \tag{3.12}$$

3.4.1.1 Random Forest

A known ensemble learning method and effective tool in prediction was used (Breiman, 2001). Random forest, a non-parametric statistical method, is commonly used when there are more predictors than responses. It allows the analyst to calculate the relative importance of predictors. The *randomForest* package (R Development Core Team, 2010) provides an R interface to the Fortran programs by Breiman and Cutler.

The principle of random forests is to combine many decision trees built using several bootstrap samples coming from the training data (observations used to fit the model) and choosing randomly a given number of input variables (denoted by mtry) at each node (Genuer, Poggi &

Tuleau-Malot, 2010). For this work, up to 58 variables were randomly chosen at each split. Note that the default values for regression is (p/3), where p is number of variables in x.

Also testing data was split to evaluate the performance of the model. Specifically K-folds cross-validation technique was applied using 10 folds. Therefore, 32 out of the 35 observations were used for training and 3 out of the 35 observations were used for testing purposes at a time.

No pruning step is performed so all the trees of the forest are maximal trees. The number of trees in the forest for this work was the default, 500 trees.

To summarize, the random forests algorithm was performed as followed (Liaw and Wiener, 2002):

1. Installed and loaded randomForest library (Brieman, 2001)
2. Fitted the model and estimated performance measure
3. Created testing and training folds
4. Performed K-folds cross validation
5. Calculated the relative importance of predictors
6. Built partial dependence plot

3.4.1.2 Basic Recursive Partitioning Trees

Recursive partitioning is an essential tool in data mining. It helps explore the structure of a set of data, while developing easy to visualize decision rules for predicting a categorical (classification tree) or continuous (regression tree) outcome (Kabacoff, 2017). Classification and regression trees can be generated through the *rpart* package (Breiman et al., 1984).

For decision tree, the following general steps were taken:

1. Installed and loaded rpart Packages
2. Created testing and training folds

3. Performed cross validation

4. Fitted the model and estimated performance measure

The flowing section demonstrate the result for each analysis.

3.4.1.3 Relation with Complexity Metrics

One way to investigate if there is a relationship between team characteristics and performance and complexity metrics is with partial dependence plots. These plots are graphical visualizations of the marginal effect of a given variable (or multiple variables) on an outcome. Partial Dependence Plot were constructed in R (Friedman, 2010).

Following section demonstrates the implementation of the methodology discussed at this chapter and the results obtained at each one of the stages.

Chapter 4: Results

4.1 Overview

The following chapter expose the results of the professor survey (Section 4.2) which reveals the current consideration of complexity in engineering education. Also, outcomes of the implementation of complexity metrics using the projects of the Process Automation (ININ 4057) course (Section 4.3) were displayed. Next, student factor survey results (Section 4.4) were presented in order to introduce the discussion of the last analysis, the complexity prediction model (Section 4.5).

4.2 Professor Survey

A total of thirteen engineering professors from various engineering department participated of the questionnaire as shown in the pie chart below.

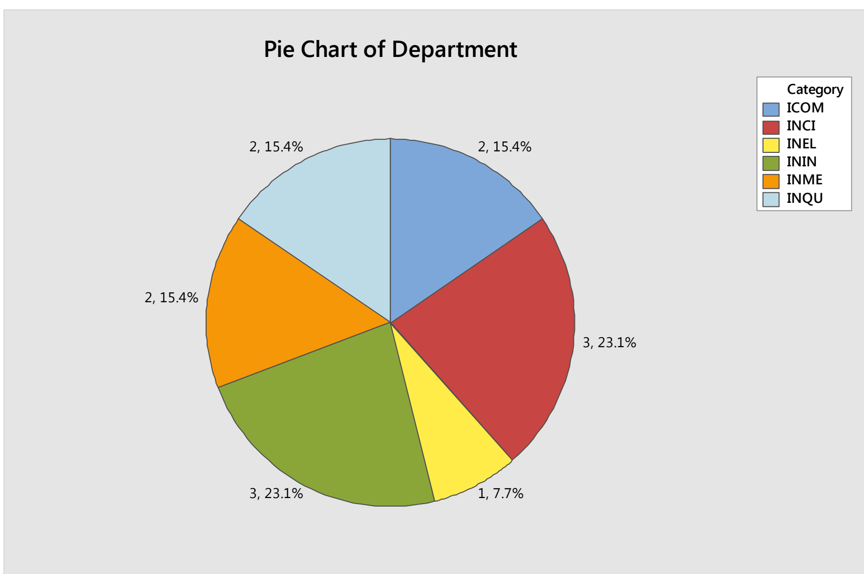


Figure 4.1: Pie chat of Engineering Department

The majority of professors interviewed were from Civil Engineering (INCI) and from Industrial Engineering department (ININ) each one with a 23.1% participation. Electrical Engineering has 7.7 % of participation.

Most of the questions, as observed in Figure 4.2, were answered with a 7 point Likert scale.

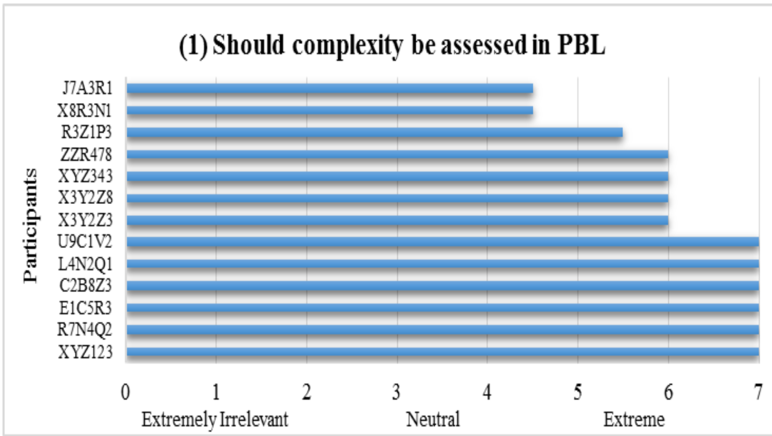


Figure 4.2: Example of one survey question answers

All of these questions were used to judge if there is sufficient evidence for the population median being greater or less that 4 (neutral in the Likert scale). 1-Sample Wilcoxon test was implemented using α = 0.05 as shown in Table 4.1 below.

Table 4.1: Professor Survey Summarized Result

Research Question	Variable	Professor Factor Survey Questions (Spanish / English)	Median	1-Sample Wilcoxon test	Result
Professor Profile	Department	1. Select the engineering department you are a part of.	N/A	N/A	N/A
	Years_ Experience	2. How many years of experience do you have as a professor?	18	N/A	N/A
	Years_ Experience_PBL	3. How many years of experience you have teaching Project-based learning course(s)?	13	N/A	N/A
	Course	4. What courses have you offered where learning is implemented through	N/A	N/A	N/A

Research Question	Variable	Professor Factor Survey Questions (Spanish / English)	Median	1-Sample Wilcoxon test	Result
		projects (Project-Based Learning)? 4a. Projects duration in weeks 4b. Projects Phases	13 4	N/A N/A	N/A N/A
Should complexity be assessed in PBL?	Complexity Impact_ Definition	6. Rate the impact of the following variables in project complexity in engineering courses. Scale 1(Not related significantly to 7 (Related significantly)	6.00	H_O: Median = 4 H_A: Median > 4	There is sufficient evidence to reject the null hypothesis ($p = 0.003$). The population median is statistically greater than 4.
	Complexity Impact_ Experience		5.50		There is sufficient evidence to reject the null hypothesis ($p = 0.011$). The population median is statistically greater than 4.
	Complexity Impact_ Methodology		5.00		There is sufficient evidence to reject the null hypothesis ($p = 0.026$). The population median is statistically greater than 4.
	Complexity Impact_ Solution		6.50		There is sufficient evidence to

Research Question	Variable	Professor Factor Survey Questions (Spanish / English)	Median	1-Sample Wilcoxon test	Result
					reject the null hypothesis (p = 0.001). The population median is statistically greater than 4
	Mesurement_ Importance_ Rating	7. Rate how important is to measure complexity in Project-Based Learning. Scale 1(Extremely irrelevant) to 7 (Extremely relevant)	6.00	H_O: Median = 4 H_A: Median > 4	There is sufficient evidence to reject the null hypothesis (p = 0.001). The population median is statistically greater than 4
	Availability Importance_ Rating	8. Rate how important is to have objective **metrics** to evaluate the complexity of project in engineering. Scale 1(Extremely irrelevant) to 7 (Extremely relevant)	6.00	H_O: Median = 4 H_A: Median > 4	There is sufficient evidence to reject the null hypothesis (p = 0.002). The population median is statistically greater than 4
Is complexity considered for PBL in engineering education?	Complexity Consideration_ PBL	9. Is complexity currently considered in Project-Based Learning? Scale 1(Substantially not considered) to 7 (Substantially considered)	5.5	H_O: Median = 4 H_A: Median < 4	There is sufficient evidence to reject the null hypothesis (0.985). The population median is statistically

Research Question	Variable	Professor Factor Survey Questions (Spanish / English)	Median	1-Sample Wilcoxon test	Result
					greater than 4.
	Metric_ Implementation	11. Are there objective metrics currently used to measure project definition or evaluation complexity? Scale 1(Substantially not used) to 7 (Substantially used)	3.00	H_O: Median = 4 H_A: Median < 4	There is insufficient evidence to reject the null hypothesis (p = 0.023). The population median is statistically less than 4.
	Complexity Measure_ Frequency	12. If complexity metrics are used, please specify which and evaluate how frequently you integrates them in project evaluation. Scale 1(Never) to 7 (Always)	6.00	N/A	N/A
Is the complexity of the <u>project definitions</u> significantly different?	Complexity Definition_ Difference_ Rating	13. When students are assigned *different project*, rate how different the project *definition* is (specifications). Scale 1(Substantially the same) to 7 (Substantially different)	4.00	H_O: Median = 4 H_A: Median > 4 H_O: Median = 4 H_A: Median < 4	There is insufficient evidence to reject the null hypothesis (p = 0.578). The population median is not statistically greater or equal than 4. There is insufficient evidence to reject the null hypothesis (p = 0.453). The population median is not

Research Question	Variable	Professor Factor Survey Questions (Spanish / English)	Median	1-Sample Wilcoxon test	Result
					statistically less than 4.
				H_O: Median = 4 H_A: Median ≠ 4	There is insufficient evidence to reject the null hypothesis (p = 0.906). The population median is not statistically different than 4.
Is the complexity of the <u>multiple solutions</u> generated significantly different?	Complexity Solution_ Difference_ Rating	14. When students are assigned the *same project*, rate how different the complexity of the *solutions* provided by students are. Scale 1(Substantially the same) to 7 (Substantially different)	4.50	H_O: Median = 4 H_A: Median < 4	There is insufficient evidence to reject the null hypothesis (p = 0.733). The population median is not ~~statistically~~ less than 4.
				H_O: Median = 4 H_A: Median > 4	There is insufficient evidence to reject the null hypothesis (p = 0.297). The population median is not statistically less than 4.
				H_O: Median = 4	There is insufficient evidence to

Research Question	Variable	Professor Factor Survey Questions (Spanish / English)	Median	1-Sample Wilcoxon test	Result
				H_A: Median ≠ 4	reject the null hypothesis (p = 0.594). The population median is not statistically different than 4.

This table summarized the answers to first four research questions, showing that engineering professors believed that complexity should be assessed in PBL, specifically in the definition and solution. Also, that there are currently no objective metrics used to measure project definition or evaluation complexity event though, complexity is considered for PBL. Responses showed that when students are assigned different project, project definition (specifications or requirements) are the same. Last, when students are assigned the same project, complexity of the solutions provided by students are the same.

Remaining survey questions were open-ended questions. For instance, Question 4, what courses have you offered where learning is implemented through projects (Project-Based Learning)? Results show that engineering professors offered 27 unique courses where learning is implemented through projects. Of those, three courses were listed by various professors: Project Design in Engineering in Computers (ICOM 5047), Process Automation Course (ININ 4057) and Integrated Project of Civil Engineering (INCI 4950). According to interviewed professors, ICOM 5047 course project was not the same for all students. For this project, students needed to design and implement prototypes in groups of 2-9 students. ININ 4057 course project consist on 4 phases: concept, design, structure and programming of an automated process. The project (problem statement) is the same for all student and is worked on teams of 2-3 students. Last, INCI 4950 course project focuses on creating a building of around 10-15 floors. This project is managed by groups of around 10-12 students.

Another question asked in the survey, Question 5, in the context of courses with project (PBL), define, what it is complexity for you; explain. Some professor defined it as a “challenge offered to student so they applied what they learned.” Other said, “Is when they used their creativity.” One mentioned “interrelation to resolve a problem.” Two professors mentioned the “Number of interactions used within a system.”

Lastly in Question 10, If so, how do you consider it (complexity in PBL), most professor said they considered it the definition and evaluation. In general, it was observed there is no standard way to measure complexity, even though most professors agreed they consider it during the definition and evaluation.

Next section shows the complexity metrics results.

4.3 Complexity Metrics

Fourteen projects from the Process Automation (ININ 4057) course, given at the University of Puerto Rico at Mayaguez, were used to implement all seven complexity metrics. Each project was done by group of two to three students with the same requirement. Particularly, for the fall semester of academic year 2015-16, the project consisted on automating eggs packing process. The following sections demonstrate results of *innovative methodologies* implemented to measure complexity in a standardized and objective matter. This section reveals how can project complexity be assessed, when student have the same project requirements.

4.3.1 Visual Component Interaction (VCI)

VCI metric was implemented to fourteen samples from first semester 2015-16. Results are showed in the Table 4.2 below. Ranges were obtain and it is observed that VCI range is [169, 2853], component (M) quantities range is [12, 42] while physical interaction (I) range is [5, 38].

Table 4.2: Visual Component Interaction Result

Visual Component Interaction (VCI)			
Sample ID	C	I	$VCI= M^2+I^2$
Group 1	34	38	2600
Group 2	19	10	461
Group 3	17	12	433
Group 4	22	23	1013
Group 5	18	13	493
Group 6	28	31	1745
Group 7	16	10	356
Group 8	26	25	1301
Group 9	12	5	169
Group 10	27	18	1053
Group 11	12	7	193
Group 12	32	25	1649
Group 13	19	13	530
Group 14	42	33	2853

Also, group VCI complexity was displayed in Figure 4.3.

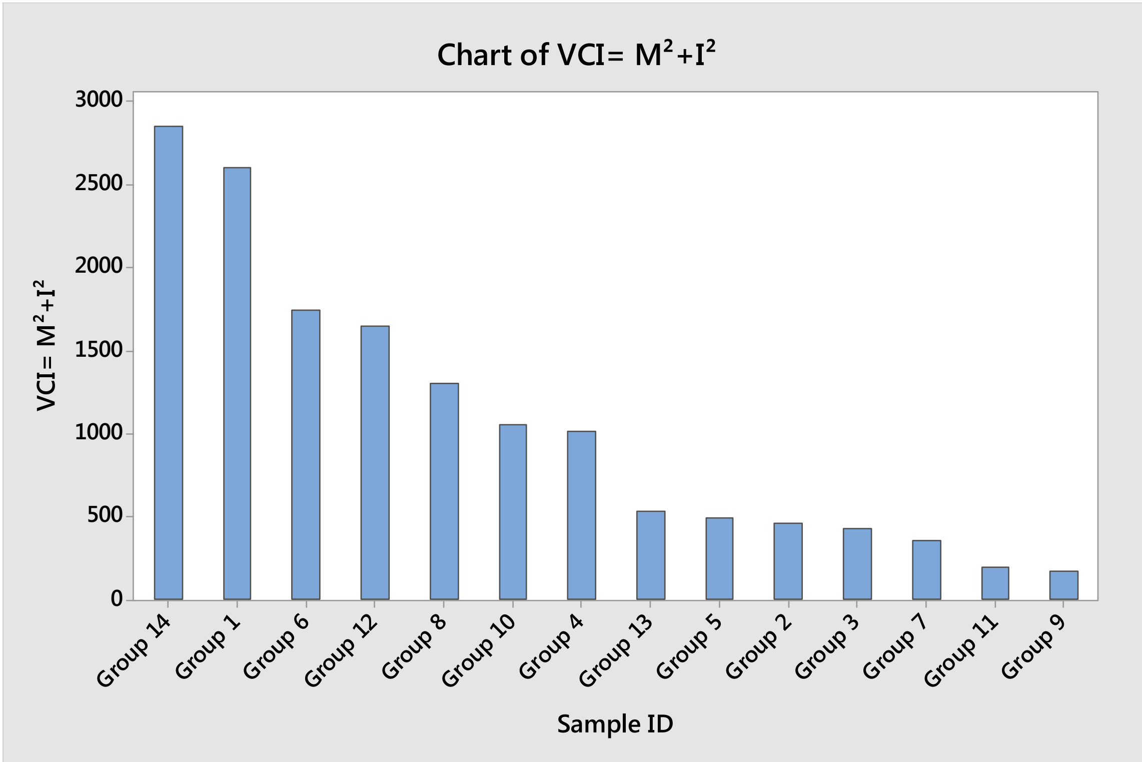


Figure 4.3: VCI Results

It is observe that Group 14 and Group 1 has the highest value, while Group 11 and Group 9 has the lowest value. In more detail, Group 14 has 43 unique components (M) and 33 physical interactions (I), compared to Group 9 that 12 unique components (M) and 5 physical interactions (I). This metric can quantify complexity among different groups, showing which group is more complex in terms of their visual component interaction. In this case evidently, Group 14 is more complex.

4.3.2 Software Component Interaction (SCI)

SCI metric was implemented to fourteen samples from first semester 2015-16. Results are shown in the Table 4.3 below. It is observed that SCI range is [277, 20213], component (M) quantities range is [9, 41] while interaction (I) range is [14, 137].

Table 4.3: SCI Results

Software Component Interaction (SCI)			
Sample ID	Components (M)	Interactions (I)	SCI= M^2+I^2
Group 1	36	113	14065
Group 2	23	29	1370
Group 3	16	38	1700
Group 4	38	137	20213
Group 5	23	43	2378
Group 6	33	108	12753
Group 7	25	45	2650
Group 8	26	86	8072
Group 9	9	14	277
Group 10	30	44	2836
Group 11	13	23	698
Group 12	39	71	6562
Group 13	30	89	8821
Group 14	41	87	9250

Also, group SCI complexity was displayed in Figure 4.4.

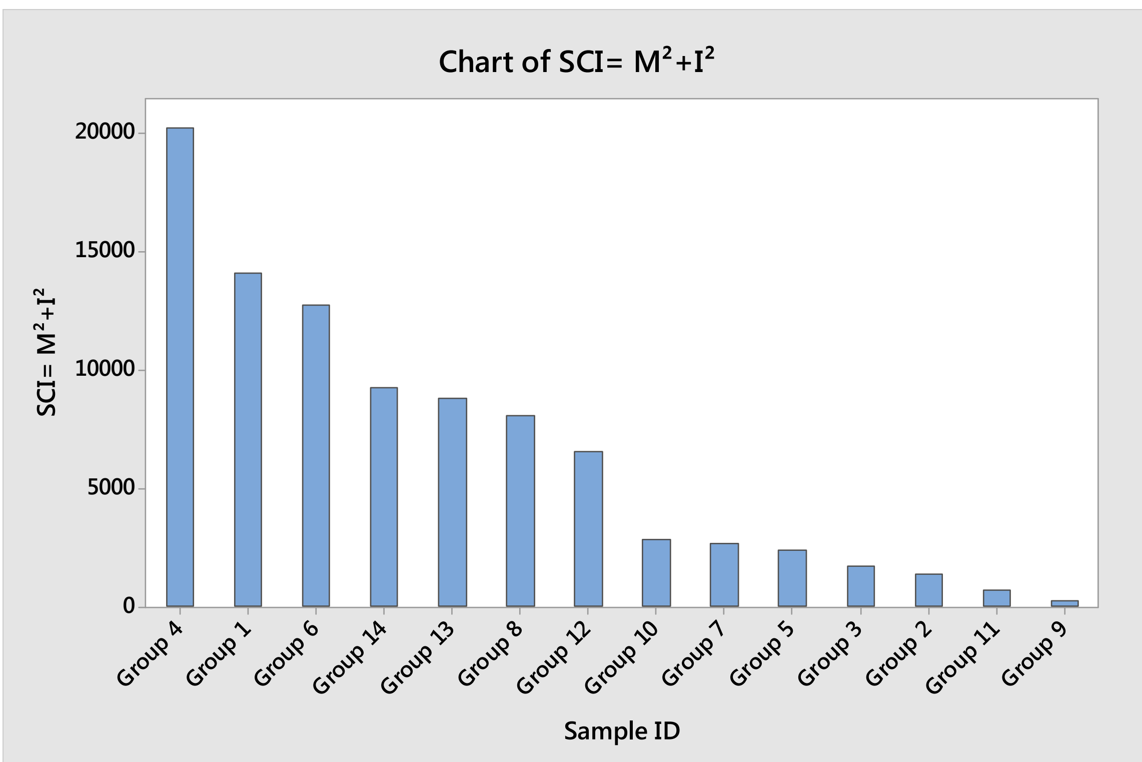


Figure 4.4: SCI results by Group

It is observed that Group 4 has the highest value, while Group 9 has the lowest value. In more detail, Group 4 had 38 unique components (M) and 137 interactions (I) in the ladder logic, compared to Group 9 that had 9 unique components (M) and 14 interactions (I). This metric can quantify software complexity among different groups, showing which group is more complex in terms of their software component interaction. In this case evidently Group 4 is more complex.

Please note, that group 9 had additional physical components that those used in the software. This could be a result of using components to enable a physical interaction but not necessarily using them in the software as an actuator or sensor.

4.3.3 Software Component Interaction with stages (SCIS)

SCIS metric was implemented in fourteen samples from first semester 2015-16. Results are shown in the Table 4.4 below. Ranges were obtained and it is observed that SCIs range is [1044, 80801], component (M) quantities range is [12, 63] while interactions (I) range is [30 to 280].

Table 4.4: SCIS Results

Sample ID	Components (M)	Interactions (I)	SCIS= M^2+I^2
Group 1	50	237	58669
Group 2	34	68	5780
Group 3	26	103	11285
Group 4	49	280	80801
Group 5	30	101	11101
Group 6	43	222	51133
Group 7	34	116	14612
Group 8	38	117	15133
Group 9	12	30	1044

Sample ID	Components (M)	Interactions (I)	SCIS= M^2+I^2
Group 10	37	103	11978
Group 11	21	77	6370
Group 12	50	152	25604
Group 13	35	143	21674
Group 14	63	183	37458

Group SCIS complexity was displayed in Figure 4.5.

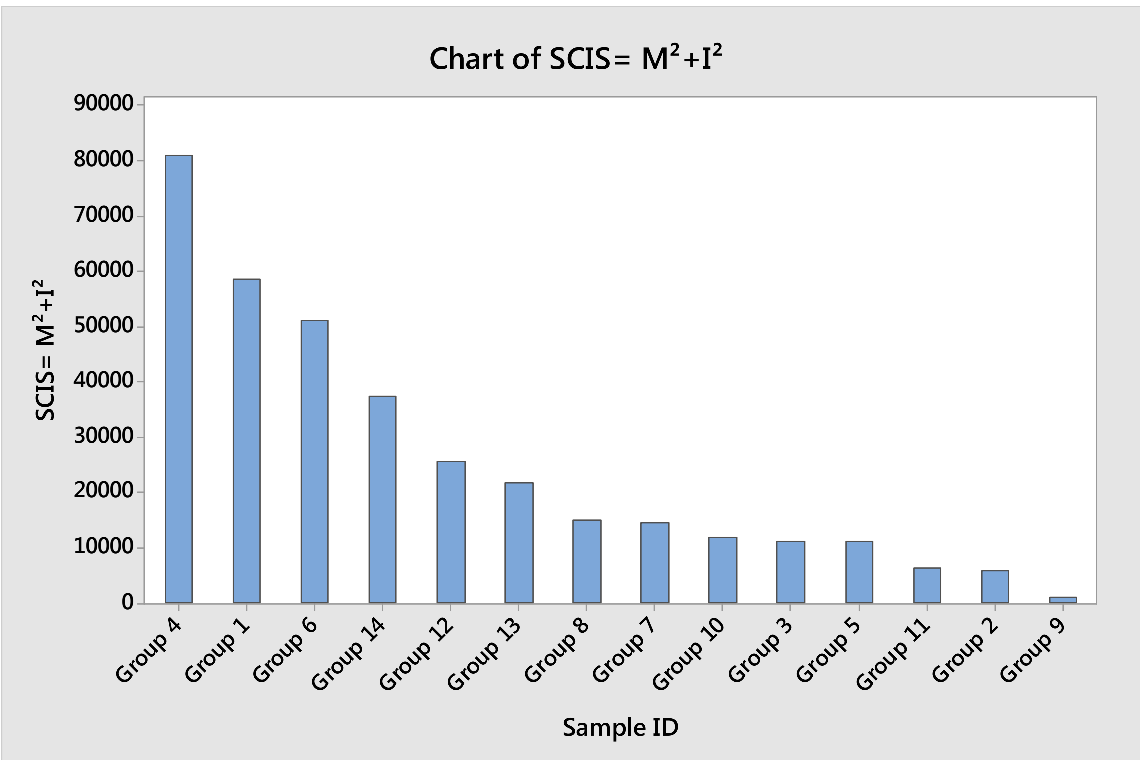


Figure 4.5: SCIS results by Group

It is observed that Group 4 has the highest value, while Group 9 has the lowest value. In more detail, Group 4 had 49 unique software components (M) and 280 interactions (I) in the ladder logic, compared to Group 9 that had 12 unique software components (M) and 30 interactions (I). This metric can quantify software complexity, including stages as an additional component among different groups, showing which group is more complex in terms of their software component interaction with stages. In this case, evidently, Group 4 is more complex.

We can say that if we compare SCI components vs SCIS components for group 4, there is a difference. That difference account for the amount of stages that were not considered in SCI but are considered in SCIS. These stages that are being considered add complexity since they interact with the rest of components in the ladder logic. Consequently it is expected to see more interactions accounted for in the SCIS implementation within the same group. For group 4, there are 143 additional interaction.

4.3.4 Lines of Code (LOC)

LOC metric was implemented to fourteen samples from first semester 2015-16. Results are shown in the Table 4.5 below. Ranges were obtain and it is observed that LOC range is [9, 113], lines (L) quantities range is [9, 114] while blank lines (n) range is [0, 10].

Table 4.5: LOC Results

Sample ID	L	n	LOC = L-n
Group 1	75	7	68
Group 2	38	2	36
Group 3	114	1	113
Group 4	63	0	63
Group 5	45	10	35
Group 6	59	0	59
Group 7	55	3	52
Group 8	49	0	49
Group 9	9	0	9

Sample ID	L	n	LOC = L-n
Group 10	43	6	37
Group 11	36	0	36
Group 12	55	0	55
Group 13	52	0	52
Group 14	62	0	62

Also, group LOC complexity was displayed in Figure 4.6.

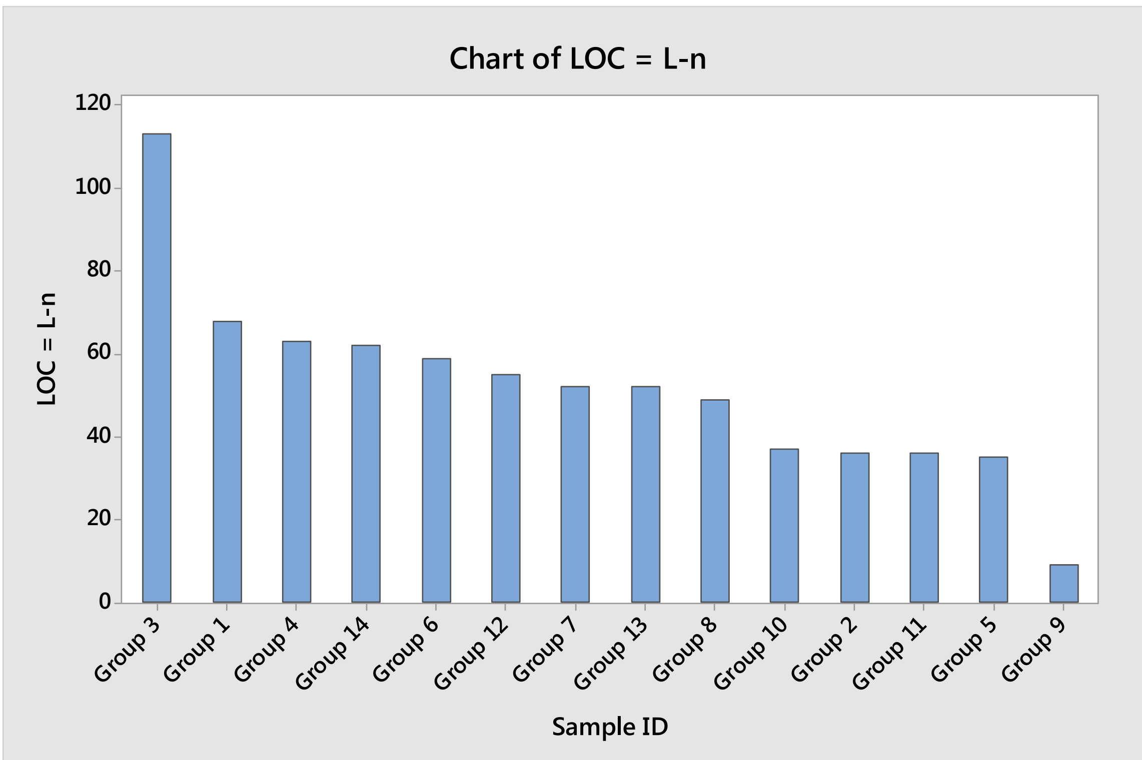


Figure 4.6: LOC results by Group

It is observed that Group 3 has the highest value while Group 9 has the lowest value. In more detail, Group 3 had 114 lines of code (L) and 1 blank line (n) compared to Group 9 that had 9 lines of code (L) and 0 blank lines (n). This metric quantifies size of the program based on the number of lines (LOC). In this case, evidently Group 3 has more lines of code, hence we can say that it is the biggest program among all groups, hence the more complex in that matter.

4.3.5 Cyclomatic Complexity (CC)

CC metric was implemented as well. Results are showed in the Table 4.6 below. Ranges were obtained and it is observed CC range is [1, 12], that jumps (e) quantities range is [4, 25], stages (n) varied range is [3 to 23] and initial stage (p) range is [1, 2].

Table 4.6: CC Results

Cyclomatic Complexity (CC)				
Sample ID	e	n	p	CC = e-n+p
Group 1	25	14	1	12
Group 2	17	11	1	7
Group 3	15	10	1	6
Group 4	15	11	1	5
Group 5	9	6	1	4
Group 6	13	10	1	4
Group 7	12	9	1	4
Group 8	13	12	1	2
Group 9	4	3	1	2
Group 10	6	6	2	2
Group 11	8	8	1	1
Group 12	15	12	1	4
Group 13	7	5	1	3
Group 14	25	23	2	4

Also, group LOC complexity was displayed in Figure 4.7.

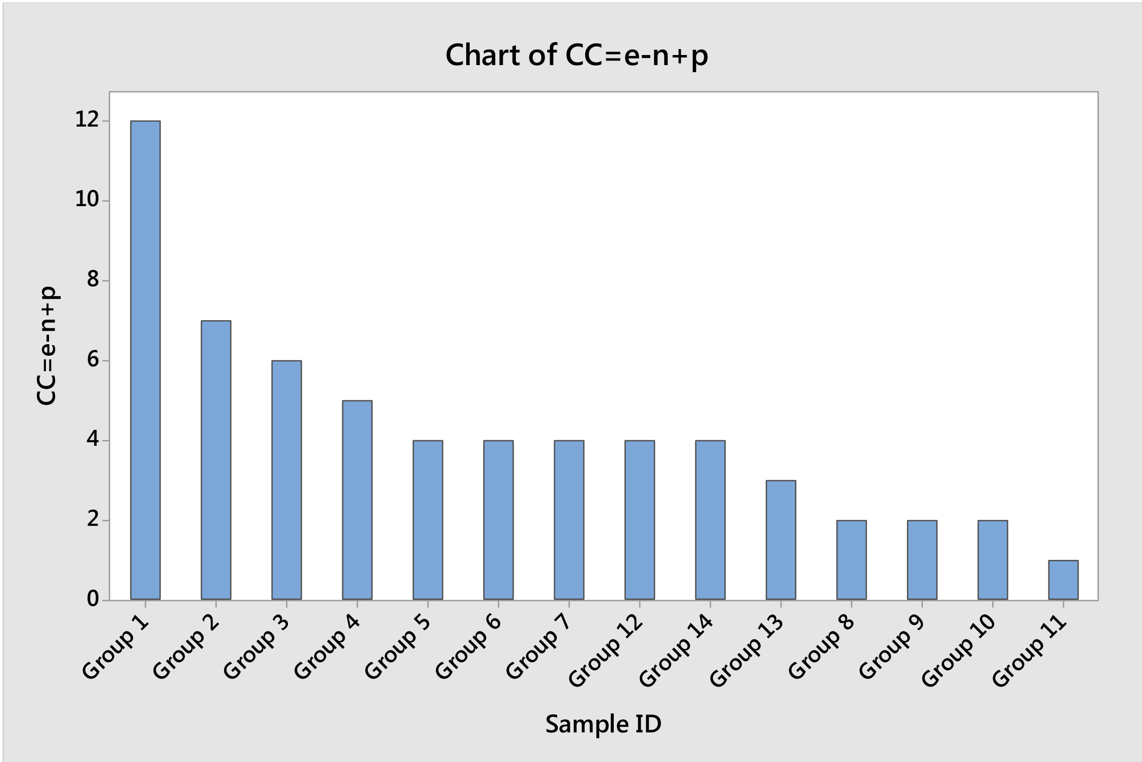


Figure 4.7: CC Complexity Metric Results

Result shows that Group 1 has the most independent paths in the Direct Soft Ladder Logic. On the contrary, Group 11 had only one linear independent path. In more detail, Group 1 had 25 jumps (e), 14 stages (n) and 1 initial stage (p) compared to Group 11 that had 8 jumps (e), 8 stages (n) and 1 initial stage (p). Surprisingly, for the first time until now, 4 groups are equally complex in terms of their independent paths. These groups are groups 6,7,12 and 14 with four linearly independent paths.

4.3.6 Process Hierarchical Function

PHF metric was implemented to determine the Automated Process functions decomposed in multiple levels. Results are showed in the Table 4.7 below. Ranges were obtained and it is observed that PHF range is [31, 59], the number of functions at each level (F) varied range is [0, 12], the number of level (l) range is [4, 7] and weight for level (k) range is [1 to 4].

Table 4.7: PHF Results

Process Hierarchical Function (PHF)					
Sample ID	Level 1 K=1	Level 2 K=2	Level 3 K=3	Level 4 K=4	PFH =∑Fl * K_i
Group 1	5	9	12	0	59
Group 2	4	9	7	0	43
Group 3	7	9	2	0	31
Group 4	4	8	6	0	38
Group 5	4	9	6	0	40
Group 6	4	9	8	0	46
Group 7	4	9	6	0	40
Group 8	4	9	9	0	49
Group 9	4	8	7	0	41
Group 10	4	8	7	0	41
Group 11	4	8	6	0	38
Group 12	4	8	8	0	44
Group 13	4	7	7	2	47
Group 14	4	8	8	0	44

Also, group PHF complexity was displayed in Figure 4.8.

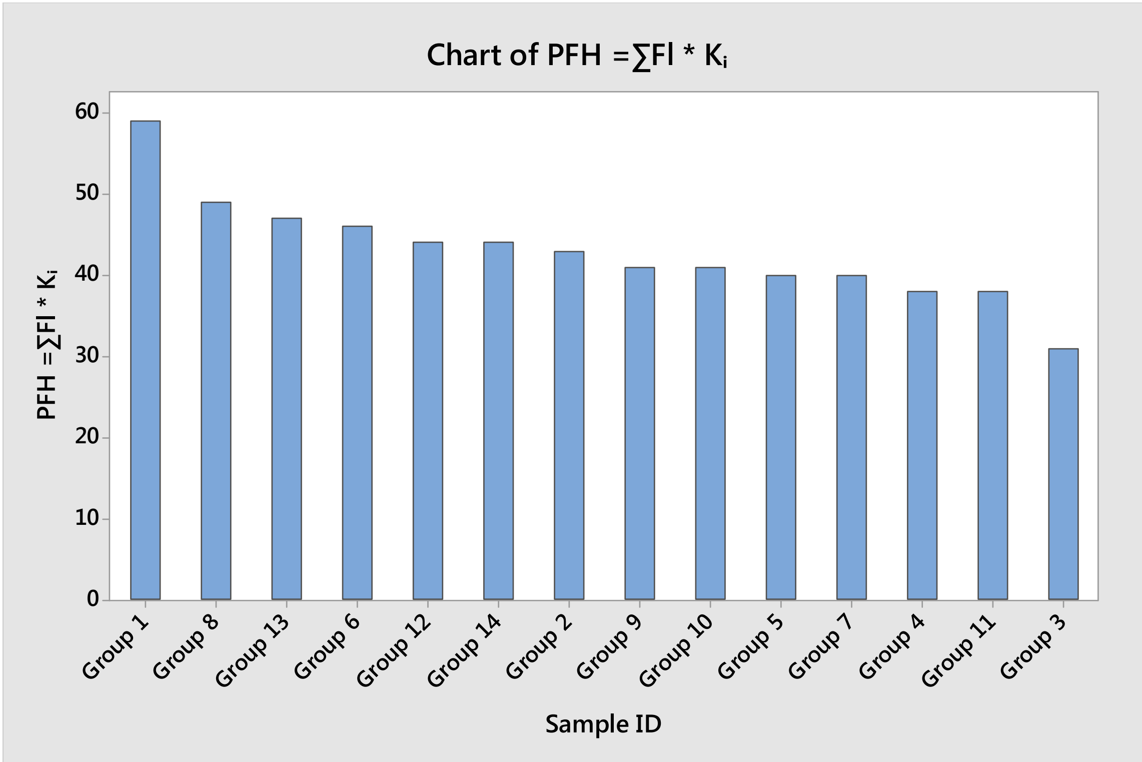


Figure 4.8: PHF Complexity Metric Results

Result shows that Group 1 has the most process functions decomposed in multiple levels. On the contrary, Group 3 has the fewest. In more detail, Group 1 had up to 12 function at level 3 compared to Group 3 that had up to 9 at level 2. Again, it is observed that there are some groups are equally complex in terms of their process functions decomposition. These groups are group 4 and 12 with a value of 38. Also groups 5 and 7 have a value of 40, while Groups 9 and 10 have a value of 41.

4.3.7 Station Hierarchical Function

SHF metric was implemented to determine the process automation stations functions decomposed in multiple levels. Results are showed in the Table 4.8 below. Ranges were obtained and it is observed that Sum SHF results varied from [53, 76], while Max SHF values were [22, 31] and Product SHF [15048, 73304].

Table 4.8 SHF Results

Group	Station 1	Station 2	Station 3	Station 4	Station 5	Sum SHF	Max SHF	Product SHF
	SHF_1	SHF_2	SHF_3	SHF_4	SHF_5			
1	11	28	17	14	6	76	28	73304
2	19	22	11	6	0	58	22	27588
3	6	22	19	6	0	53	22	15048
4	14	22	11	6	0	53	22	20328
5	11	25	11	9	0	56	25	27225
6	17	26	11	9	0	63	26	43758
7	13	22	14	6	0	55	22	24024
8	16	31	11	6	0	64	31	32736
9	11	25	14	6	0	56	25	23100
10	11	25	14	6	0	56	25	23100
11	11	22	14	6	0	53	22	20328
12	11	28	11	9	0	59	28	30492
13	14	29	14	6	0	63	29	34104
14	11	25	17	6	0	59	25	28050

Figure 4.9 displayed Sum SHF metric. These metrics were selected among the others because it captures functional decomposition of all stations of the process automation but at the same time is analyzed as a single measure.

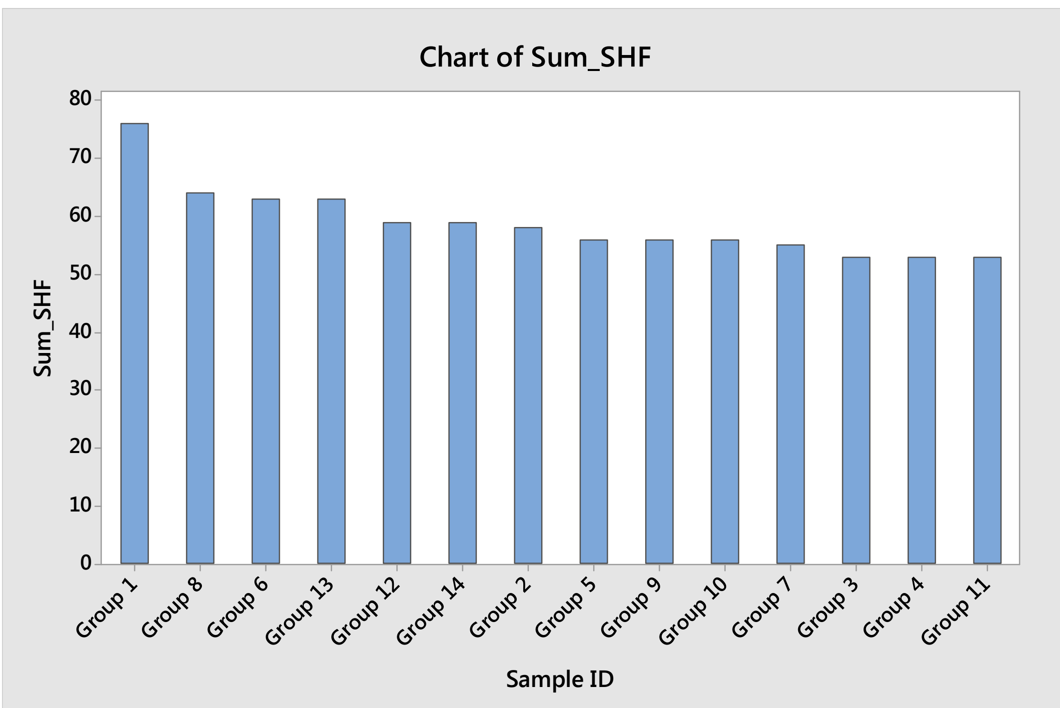


Figure 4.9: SHF Complexity Metric Results

Results shows that Group 1 has the most station functions decomposed in multiple levels. On the contrary Group 11 has the fewest, but not by much. In more detail, Group 1 had up to 15 function at level 3 compared to Group 11 that had up to 9 at level 3. It is observed that there are some groups are equally complex in terms of their station functions decomposition. These groups are group 3, 4 and 11 with a value of 53. Also groups 5, 9 and 10 have a value of 56, while Groups 13 and 6 have a value of 63.

4.3.8 Complexity Metric Comparison

Beside the individual metrics results, a spider diagram was created to compare all seven metric among the fourteen groups as shown in Figure 4.10 below.

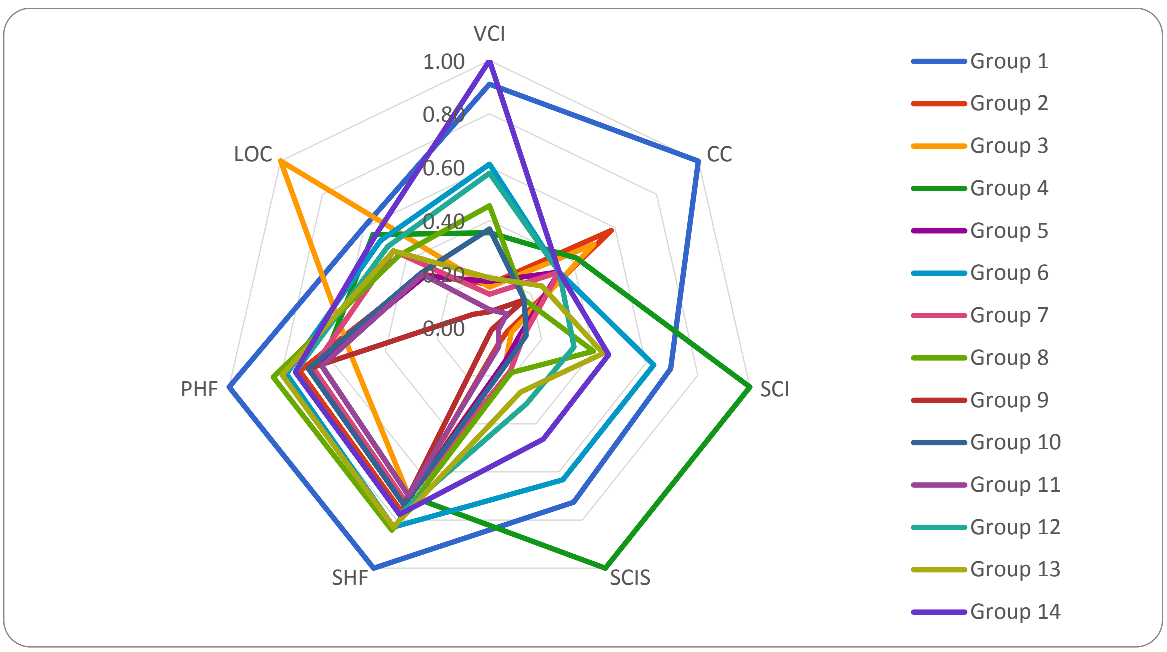


Figure 4.10: Complexity Metric Results by Group (Collado, Medina & Soto, 2016)

Each one of the seven complexity metrics results~~,~~ were normalized for all groups. A higher value within the scale of 0 to 1 indicates a higher value of complexity. It is observed that the normalized result for various complexity metrics varied from group to group. Particularly, group 1, compared to the rest of the group, created more independent paths in the software (CC) and added more functions to their overall process (PHF) and stations (SHF). Similar, group 8, generated their automated process with higher number of software component interactions (SCI and SCIS).

Another analysis conducted is a correlation analysis among complexity metrics. It is of interest to determine the extent to which two metrics correlates. Pearson correlation was used to evaluate the linear relation between pair of metrics as shown in Figure 4.11.

Correlation: VCI, CC, SCI, SCIS, SHF, PHF, LOC

	VCI	CC	SCI	SCIS	SHF	PHF
CC	0.460 0.005					
SCI	0.665 0.000	0.376 0.026				
SCIS	0.660 0.000	0.475 0.004	0.960 0.000			
SHF	0.703 0.000	0.615 0.000	0.530 0.001	0.453 0.006		
PHF	0.682 0.000	0.477 0.004	0.519 0.001	0.423 0.011	0.955 0.000	
LOC	0.248 0.151	0.435 0.009	0.264 0.125	0.323 0.059	0.068 0.699	-0.185 0.286

Cell Contents: Pearson correlation
P-Value

Figure 4.11: Correlation Analysis

When coefficient absolute correlation is closer to 1 or -1, the data points fall on a line more tightly. When value is cero, no linear relationship exist. It is observed from the result of this analysis that there is correlation among almost all metrics expect SHF- LOC. The most significant correlation is within SCI and SCIS with a correlation of 0.960 and SHF and PHF with 0.955. This was expected since both, SCI and SCIS measures component and interaction in the software, however, SCI does not consider stages and SCIS does. Similar both SHF and PHF measure complexity based on functionality, the difference is that SHF measures it by station and PHF assess the process automation as a whole.

In order to answer research question three, is the complexity of the multiple solutions generated significantly different? Friedman rank sum test was used to compare all seven complexity measures between fourteen groups. For this test, complexity metrics were normalized using the following equation:

$$\frac{Y}{\max(Y)} \tag{4.1}$$

Figure 4.12 below shows that according to this test, there is sufficient evidence to reject null hypothesis. Hence there is statistical significant difference between the groups.

```
        Friedman rank sum test

data:  data
Friedman chi-squared = 52.868, df = 13, p-value = 9.527e-07
```

Figure 4.12: Friedman Rank Sum Test

Another approach used was the Pairwise comparison using Conover's test.

```
        Pairwise comparisons using Conover's test for a two-way
                 balanced complete block design

data:  data

        Grupo1  Grupo2  Grupo3  Grupo4  Grupo5  Grupo6  Grupo7  Grupo8  Grupo9  Grupo10 Grupo11 Grupo12 Grupo13
Grupo2  0.05772 -       -       -       -       -       -       -       -       -       -       -       -
Grupo3  0.05153 1.00000 -       -       -       -       -       -       -       -       -       -       -
Grupo4  1.00000 1.00000 1.00000 -       -       -       -       -       -       -       -       -       -
Grupo5  0.01113 1.00000 1.00000 1.00000 -       -       -       -       -       -       -       -       -
Grupo6  1.00000 1.00000 1.00000 1.00000 0.53142 -       -       -       -       -       -       -       -
Grupo7  0.03252 1.00000 1.00000 1.00000 1.00000 1.00000 -       -       -       -       -       -       -
Grupo8  1.00000 1.00000 1.00000 1.00000 1.00000 1.00000 1.00000 -       -       -       -       -       -
Grupo9  0.00017 1.00000 1.00000 0.29125 1.00000 0.01600 1.00000 0.39477 -       -       -       -       -
Grupo10 0.05772 1.00000 1.00000 1.00000 1.00000 1.00000 1.00000 1.00000 1.00000 -       -       -       -
Grupo11 6.5e-05 1.00000 1.00000 0.13958 1.00000 0.00681 1.00000 0.19214 1.00000 1.00000 -       -       -
Grupo12 1.00000 1.00000 1.00000 1.00000 1.00000 1.00000 1.00000 1.00000 0.21343 1.00000 0.10075 -       -
Grupo13 1.00000 1.00000 1.00000 1.00000 1.00000 1.00000 1.00000 1.00000 0.39477 1.00000 0.19214 1.00000 -
Grupo14 1.00000 1.00000 1.00000 1.00000 0.85864 1.00000 1.00000 1.00000 0.02894 1.00000 0.01257 1.00000 1.00000

P value adjustment method: bonferroni
```

Figure 4.13: Pairwise Comparison

This approach shows exactly which group differs from one another. For instance, Group 1 is statistically different from group 11, 5, 7 and 9. Similar, group 6 differ from group 9 and so on. Multiple analysis on the same variable, increase the chance of committing a Type I error. Therefore adjusting the p-value to a more stringent value making it less likely to commit Type I Error (Bonferroni Correction, 2015). Equation 4.2 shows the p value adjustment used for this pairwise comparison test.

$$\alpha_{critical} = 1 - (1 - \alpha_{altered})^{k} \quad (4.2)$$

Where, $\alpha_{altered} = \alpha/n$

Next section, shows student factor survey results.

4.4 Student Factor Survey

Student Factor survey questionnaire was employed to measure age, gender, knowledge perceived abilities, perceived difficulty, individual contribution, team dynamic and performance. A total of thirty five students agreed to voluntarily participant in this survey. All they are industrial engineering students that took the Process Automation (ININ 4057) course. Also they are the designers of the small scale automated process used to implement complexity metrics, discussed in the previous Section (4.3 Complexity Metrics). Table 4.9 summarizes results obtained from student factor survey and complexity metrics implementation results.

Table 4.9: Student Factor Survey and Complexity Metric Result Range

Variables	Designation	Category	Data Range	Connotation
VCI	Output	Complexity Metric	[169, 2853]	Visual Component Interaction

Variables	Designation	Category	Data Range	Connotation
CC			[1, 12]	Number of Independ path in a program
SCI			[277, 20213]	Software Component Interaction
SCIS			[1044, 80801]	Software Component Interaction with stages
SHF			[53, 76]	Station Functions
PHF			[31, 59]	Process Functions
LOC			[9, 113]	Program Volume
Gender	Input/ Predictors	Gender	[0, 1]	0=Female, 1= Male
Age		Individual Age	[21, 31]	Age in Years
Age_Median		Group Age	[22,25]	Age in Years
Academic_Load		Academic Load	[6, 21]	Academic credits
Knowledge_Work	Input/ Predictors	Individual Knowledge	[0, 1]	0= No Knowledge, 1= Knowledge
Knowledge_Project			[0, 1]	
Knowledge_Internship			[0, 1]	
Knowledge_Coop			[0, 1]	
Knowledge_Automation			[0, 1]	
Knowledge_Programming			[0, 1]	
Know_Work_Median		Group Knowledge	[0, 1]	Percentage of knowledge within the group From 0 = No Knowledge to 1= Knowledge
Know_Project_Median			[0, 1]	
Know_Intern_Median			[0, 1]	
Know_Coop_Median			[0, 1]	
Knowe_Auto_Median			[0, 1]	
Know_Prog_Median			[0, 1]	
PerHabilities_Programming	Input/ Predictors	Perceived Abilities/ Skills	[2, 7]	Perceived abilities rating Scale 1 (Very Poor) to 7 (Excellent)
PerAbilities_Circuit			[2, 7]	
PerComplexity_Design	Input/ Predictors	Difficulty/ Complexity	[3, 6]	Perceived difficulty rating Scale 1 (Extremely Simple) to 7 (Extremely Complex)
PerComplexity_Structure			[2, 6]	
PerComplexity_PLC			[2, 6]	
PerComplexity_Programming			[1, 6]	
PerComplexity_Troubleshooting			[2, 6]	
PerComplexity_HMI			[1, 6]	
PerComplexity_Motors			[1, 4]	

Variables	Designation	Category	Data Range	Connotation
PerComplexity_Neumuatic			[1, 4]	
PerComplexity_Sensors			[1, 4]	
PerComplexity_Relays			[1, 4]	
Individual_Motivation	Input/ Predictors	Individual Contribution	[2, 7]	Student Motivation on a Scale 1 (Not Motivated) to 7 (Extremely Motivated)
Individual_ContributionDesign			[3, 6]	Student Design Contribution Scale 1 (Very Poor) to 7 (Excelent)
Individual_ContributionStructure			[3, 6]	Student Structure Contribution Scale 1 (Very Poor) to 7 (Excelent)
Individual_ContributionPLC			[2, 6]	Student PLC Contribution Scale 1 (Very Poor) to 7 (Excelent)
Individual_ContributionProgramming			[2, 6]	Student Programming Contribution Scale 1 (Very Poor) to 7 (Excelent)
Individual_ContributionTroubleshooting			[3, 6]	Student Troubleshooting Contribution Scale 1 (Very Poor) to 7 (Excelent)
Individual_ContributionHMI			[1, 6]	Student Human Machine Interface Contribution Scale 1 (Very Poor) to 7 (Excelent)
Individual_ProjectHrs			[40, 400]	Estimated hours spend in project
Individual_PerPerformance	Input/ Predictors	Team Dynamic	[4, 7]	Student Perceived Self Performance Scale 1 (Very Poor) to 7 (Excelent)

Variables	Designation	Category	Data Range	Connotation
Individual_ComparedContribution			[7, 7]	Student contribution among group members Scale 1 (Substantially less) to 7 (Substantially more)
Group_Qty			[2, 3]	Group Qty.
Group_CommunicationFreq			[4,7]	Group Communication Frequency
Group_Female%			[0, 1]	Group Female %
Group_MemberEngage%			[0.33, 1]	Group performance
Goup_DecisionsUnanimity			[0, 1]	0=No, 1=Yes
Goup_DecisionsAuthority			[0, 1]	
Goup_DecisionsMinority			[0, 1]	
Goup_DecisionsMayority			[0, 1]	
Goup_DecisionsConsensus			[0, 1]	
Performance_ININ3.51	Input Predictors	Performance	[0, 1]	0=No, 1=Yes
Performance_ININ3.01			[0, 1]	0=No, 1=Yes
Performance_ININ2.51			[0, 1]	0=No, 1=Yes
Performance_General3.51			[0, 1]	0=No, 1=Yes
Performance_General3.01			[0, 1]	0=No, 1=Yes
Performance_General2.51			[0, 1]	0=No, 1=Yes
Performance_INEL4075			[2 , 4]	Grades where A=4, B=3, C=2
Performance_INEL4076			[2 , 4]	
Performance_INEL4077			[2 , 4]	
Performance_INME4055			[2 , 4]	
Performance_INME4056			[2 , 4]	
Performance_ININ4057			[2 , 4]	
Performance_INGE3016			[2 , 4]	
Performance_GrpLabs			[68, 97]	Course Grade
Performance_GrpDesign			[86, 100]	
Performance_GrpStructure			[88, 100]	
Performance_GrpDemo			[58, 98]	
Perfornance_IndExams			[46, 100]	
Perr Eval (max 30)			[0, 30]	Peer Evaluation
Absence			[0, 4]	Count of Absence to the ININ 4057 course
Lateness			[0, 9]	Count of Lateness to the

Variables	Designation	Category	Data Range	Connotation
				ININ 4057 course

Since Reduced Model I search to predict complexity based on performance only (see Figure 3.16), Cronbach's alpha was used to determine the scale of internal consistency for performance construct. Table 4.10 shows that Cronbach's alpha is > 0.7, hence predictors are a reliable measure of performance.

Table 4.10: Cronbach Alfa Analysis

Category	Variables	Cronbach's alpha > 0.7
Performance	Performance_INEL4075 Performance_INEL4076 Performance_INEL4077 Performance_INME4055 Performance_INME4056 Performance_ININ4057 Performance_INGE3016 Performance_GrpLabs Performance_GrpDesign Performance_GrpDemo Performance_GrpStructure Perfornance_IndExams	0.7465

Next section shows prediction models results.

4.5 Prediction Models

As mentioned in the prediction model methodology (Figure 3.16), two methods of regression, random forest and decision tree, were used to run three models (General Model, Reduced Model I and Expanded Model II) and predict each one of the seven complexity metrics. Hence, there are in total 42 models. Table 4.11 summarizes each model with its R^2, MAPE and dF value.

Table 4:11 Prediction Models Result Summary

Method	Model	Response	Rsq	MAPE	dF
randomForest	General Model	CC	0.548536	0.397585	1.426677
randomForest	Reduced Model I	CC	0.535984	0.333267	1.496542
randomForest	Expanded Model II	CC	0.41384	0.154355	1.531863
decision tree	Expanded Model II	CC	0.024134	0.340146	0.553503
decision tree	Reduced Model I	CC	-0.85128	0.693474	-1.55191
decision tree	General Model	CC	-0.96172	0.657799	-1.7018
randomForest	Expanded Model II	LOC	0.738041	0.095312	1.799446
randomForest	Reduced Model I	LOC	0.523686	0.242122	1.200094
randomForest	General Model	LOC	0.333015	0.272883	0.877021
decision tree	Expanded Model II	LOC	0.097521	0.348448	0.398937
decision tree	General Model	LOC	-0.84076	0.332532	-0.83888
decision tree	Reduced Model I	LOC	-0.93683	0.475244	-1.26935
randomForest	Expanded Model II	PHF	0.922536	0.014024	1.847335
randomForest	Reduced Model I	PHF	0.801439	0.045817	1.369966
randomForest	General Model	PHF	0.738634	0.038414	1.382474
decision tree	General Model	PHF	0.196712	0.08819	0.253179
decision tree	Expanded Model II	PHF	0.044322	0.063748	0.35407
decision tree	Reduced Model I	PHF	-0.15297	0.09186	-0.16582
randomForest	Expanded Model II	SCI	0.887782	0.583176	1.705368
randomForest	Reduced Model I	SCI	0.786818	1.520328	1.118175
randomForest	General Model	SCI	0.773659	0.99664	1.36793
decision tree	Expanded Model II	SCI	0.381896	0.661754	1.095838
decision tree	General Model	SCI	0.273575	1.979339	0.308155
decision tree	Reduced Model I	SCI	0.051351	1.286805	0.407724
randomForest	Expanded Model II	SCIS	0.873276	0.299414	1.845223
randomForest	General Model	SCIS	0.590381	0.983414	1.167694
decision tree	Expanded Model II	SCIS	0.272862	0.492326	1.057958
randomForest	Reduced Model I	SCIS	0.183463	1.477243	0.446451
decision tree	General Model	SCIS	0.070582	1.110499	0.506771
decision tree	Reduced Model I	SCIS	-0.88769	1.934487	-1.01651
randomForest	Expanded Model II	SHF	0.862238	0.011113	1.820617
randomForest	General Model	SHF	0.505902	0.029423	1.111795
randomForest	Reduced Model I	SHF	0.426447	0.031977	0.978417
decision tree	Expanded Model II	SHF	0.038019	0.0463	0.29673
decision tree	Reduced Model I	SHF	-0.74776	0.060155	-0.83823
decision tree	General Model	SHF	-0.76182	0.061951	-0.88354
randomForest	Reduced Model I	VCI	0.790529	0.449659	1.390954
randomForest	General Model	VCI	0.735884	0.523428	1.258034
randomForest	Expanded Model II	VCI	0.935437	0.138508	1.860113
decision tree	Expanded Model II	VCI	0.673461	0.293108	1.423917
decision tree	Reduced Model I	VCI	0.224424	0.990142	0.239913

Method	Model	Response	Rsq	MAPE	dF
decision tree	General Model	VCI	0.213185	0.69718	0.523778

Each model was analyzed based on the desirability function score to determine which one is the best prediction model for each complexity metric. This dF seeks to minimize the mean absolute percentage error and maximize the variability explained by the model. See equation below:

$$dF = \frac{R^2}{\max(R^2)} + \frac{1}{MAPE} \tag{4.3}$$

Once analyzed, models with higher score were select for each complexity as shown in Table 4.12 below.

Table 4.12: Selected Prediction Models for Complexity

Method	Model	Response	Rsq	MAPE	DF
randomForest	Expanded Model II	VCI	0.935437	0.138508	1.860113
randomForest	Expanded Model II	PHF	0.922536	0.014024	1.847335
randomForest	Expanded Model II	SCIS	0.873276	0.299414	1.845223
randomForest	Expanded Model II	SHF	0.862238	0.011113	1.820617
randomForest	Expanded Model II	LOC	0.738041	0.095312	1.799446
randomForest	Expanded Model II	SCI	0.887782	0.583176	1.705368
randomForest	Expanded Model II	CC	0.4138399	0.1543549	1.531863

Now that the best model for each complexity model was selected, each one is analyzed in order to determine important variables and the relation between team performance and characteristic and performance.

Using student t-test algorithm (See Appendix G), important variables for each model were identified as shown in Table 4.13.

Table 4.13: Prediction Models Important Variables

Variable		VCI	CC	SCIS	SHF	PHF	LOC
Individual_ContributionProgramm	Original		X				

Variable		VCI	CC	SCIS	SHF	PHF	LOC
Performance_ININ4057	Original		X				
Perfornance_IndExams	Original		X				
Knowledge_Project	Median		X				
Knowledge_Programming	Median	X	X	X	X	X	X
Individual_Motivation	Median	X	X	X	X	X	X
PerHabilities_Programming	Median		X		X	X	X
PerComplexity_Troubleshooting	Median	X	X	X	X	X	
PerComplexity_HMI	Median	X	X	X	X	X	X
PerComplexity_Neumuatic	Median	X	X	X	X	X	X
PerComplexity_Sensors	Median		X	X	X	X	X
Individual_ContributionDesign	Median		X		X	X	X
Individual_ContributionPLC	Median	X	X	X	X	X	X
Individual_ContributionProgramming	Median	X	X	X	X	X	X
Individual_ContributionTroubleshooting	Median	X	X	X	X	X	
Individual_ProjectHrs	Median	X	X	X	X	X	X
Performance_ININ4057	Median		X			X	
Performance_INGE3016	Median	X	X	X	X	X	X
Perfornance_IndExams	Median	X	X	X	X	X	X
Absense	Median	X	X	X	X	X	X
Knowledge_Project	Minimum		X				
Knowledge_Programming	Minimum	X	X	X		X	X
Individual_Motivation	Minimum	X	X	X	X	X	X
PerHabilities_Circuit	Minimum	X	X	X	X	X	X
PerComplexity_Design	Minimum	X	X	X	X	X	X
PerComplexity_PLC	Minimum	X	X	X	X	X	X
PerComplexity_Programming	Minimum	X	X	X	X	X	X
PerComplexity_Troubleshooting	Minimum	X	X		X	X	X
PerComplexity_Relays	Minimum	X	X	X	X	X	X
Individual_ContributionPLC	Minimum	X	X	X		X	X
Individual_ContributionProgramming	Minimum	X	X	X	X	X	X
Individual_ContributionHMI	Minimum	X	X	X	X	X	X
Individual_ProjectHrs	Minimum	X	X	X	X	X	X
Individual_ComparedContribution	Minimum	X	X	X	X	X	X
Performance_INEL4075	Minimum	X	X	X	X	X	X
Performance_ININ4057	Minimum		X	X		X	X
Performance_INGE3016	Minimum		X				
Perfornance_IndExams	Minimum	X	X	X	X	X	X
Knowledge_Programming	Maximum	X	X	X	X		X
PerHabilities_Programming	Maximum		X				
PerComplexity_PLC	Maximum		X		X	X	
PerComplexity_Sensors	Maximum		X	X			
Individual_ContributionProgramming	Maximum		X				

Variable		VCI	CC	SCIS	SHF	PHF	LOC
Individual_ProjectHrs	Maximum	X	X	X	X		X
Performance_ININ4057	Maximum		X				
Perfornance_IndExams	Maximum	X	X	X	X	X	X
Lateness	Maximum	X	X	X	X	X	X
Total important variables		74	47	73	73	78	75
Common variables		24					

As observed, six of the seven models had important variables, 24 of them in common. Please note SCI model did not had statistically important variable.

Subsequently, to determine the marginal effect of independent variable on the response, partial dependence plot were constructed in R for each one of the responses (complexity metrics). For instance, a subset of all the plots can be seen in Figure 4.14 (remaining are in Appendix H).

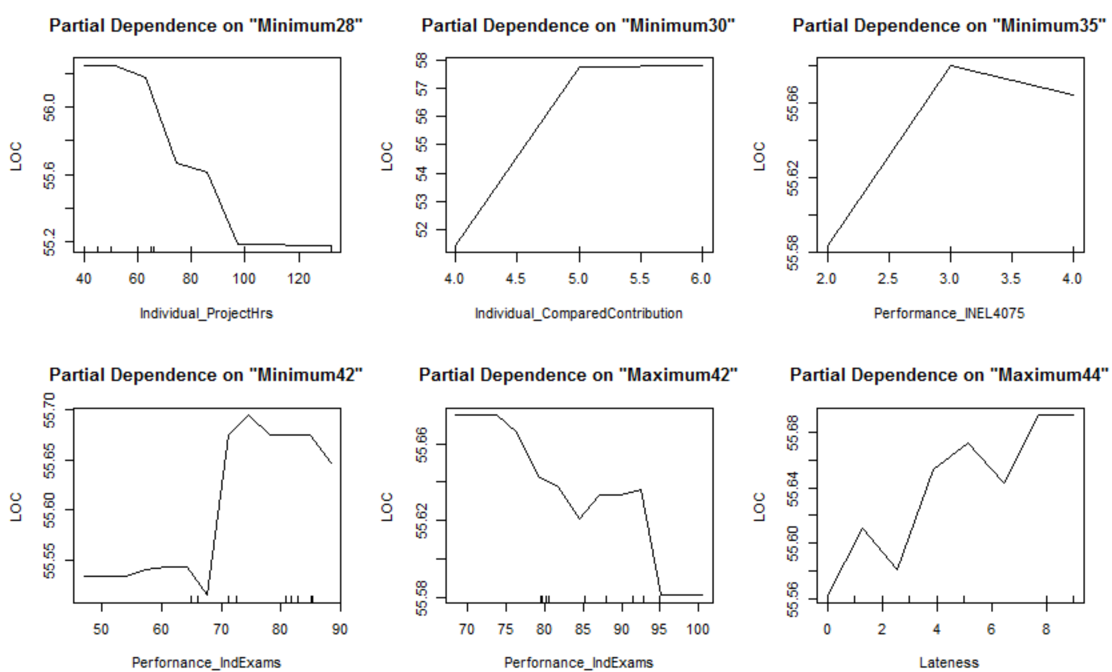


Figure 4.14: Partial Dependence Plot for LOC

These partial dependence plots provide the answer to last research question, is there a relationship between project outcome complexity with team characteristics and performance? As observed in the top left plot in Figure 4.14, LOC complexity begin to decrease as the group minimum individual hour's increases'. Also it is observed that LOC increase when the contribution from one membered compared to the others increase. In terms of performance, plot indicates that when group's minimum GPA is 3 (out of maximum GPA of 4) in the INEL 4075 course, LOC reaches is max value. Similarly, bottom left plot shows that LOC increase abruptly when the minimum ININ 4057 exam grade within the group was 75 in a scale of 100 (also known as grade C). These finding are very interesting. Most importantly, the relationship between project outcome complexity with team characteristics and performance is now known.

Chapter 5: Conclusion

In summary, this research presented current considerations of complexity for Project-Based Learning (PB) through the implementation of professor survey to thirteen UPRM engineering professor from various engineering departments. It was confirmed that 83% of the participants believed complexity is considered in engineering education. Also, 46% acknowledged that it is necessary to assess complexity, since they believed that project definitions differ when students are assigned different projects. Lastly, 54% said that multiple solutions generated by students who are assigned the same project, differ.

Most noteworthy, this research studied a variety of complexity metrics, in order to present the adaptation, development and implementation of seven complexity metric to asses design complexity in PBL. These metrics are: (1) Visual Component Interaction metric (VCI), that was developed to measure overall design complexity based on the component interaction that can be observed physically in the process, independently of the program or software. (2) Similar, Software Component Interaction (SCI) metric is intended to measure the complexity of process automation components and interactions through the software, particularly, the ladder logic. (3) The Software Component Interaction with Stages (SCIS) metric is proposed as a modification of the SCI metric, that include the assessment of the software stages. (4) Lines of Codes (LOC) metric is used to measure the lines of code of ladder logic. (5) Cyclomatic Complexity (V(G)) metric measures the number of linearly independent paths in the Direct Soft Ladder Logic software. (6)

The Process Hierarchical Functionality (PHF) metric is intended to assess specific functions of completed process automation projects. Last but not least (7), the Station Hierarchical Functionality (SHF) measures the complexity in relation to station functions.

Even though the simplest and easiest metric to implement is LOC, it does not considered the interaction in each line of code. Therefore, this work presented a robust metric such as SCIS, which account not only for the amount of components, but the interaction in each line of code of the software to assess complexity.

A student factor survey was submitted to collect variables that represent designer's individual and group characteristic and performance among others. In total, a maximum of 173 variables were used as predictor in the prediction model using both *randomForest* and *rpart* function in R. Results indicated that Random Forest provided better models based on the coefficient of determination (R^2) and the mean absolute deviation percentage (MAPE) than basic recursive partitioning trees. For instance, VCI random forest prediction model R^2 is 0.935 while MAPE is 0.13.

Finally, one prediction model was developed for each one of the seven responses and variable importance was assessed. Through Partial dependence plot, the relationship among designer's characteristic and performance with complexity was revealed. Out of the 173 variables used to generate prediction model, 24 variables resulted commonly important in six of the seven model.

A future application for this method can be used in school systems and higher education using a more generalized assessment of complexity, for instance, not only for process automation. Moving forward this assessment can be extended to control PBL complexity, not only to quantify and analyze it.

References

Aitken, N. D. (1982). College student performance, satisfaction, and retention: Specification and estimation of a structural model. *The Journal of Higher Education, 53,* 32–50

Baccarini, D. (1996). The concept of project complexity—a review. *International Journal of Project Management*, 14(4), 201-204.

Balve, P., & Albert, M. (2015). Project-based learning in production engineering at the Heilbronn Learning Factory. *The 5th Conference on Learning Factories 2015*, 104-108. doi: 10.1016/j.procir.2015.02.215

Bashir, H. A., & Thomson, V. (1999). Estimating design complexity. *Journal of Engineering Design*, 10(3), 247-257.

Basili, Victor R., & Perricone, Barry T. (1984). Software errors and complexity: an empirical investigation. *Communications of the ACM*, 42 – 52.

Bajwa, S. S., Gencel, C., & Abrahamsson, P. (2014). Software product size measurement methods: A systematic mapping study. *Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement*, 176-190.

Bonferroni Correction. (2015, March 31). Retrieved May 07, 2017, from http://www.statisticssolutions.com/bonferroni-correction/

Bosch-Rekveldt, M., Bakker, Herman M., & Verbraeck, A. (2010). *Evaluating a complexity framework - a practitioners view on project complexity*. Retrieved February 8, 2016, from pubs.iids.org/index.php/attachments/single/46

Buck Institute for Education (2016). *What is project based learning (PBL)?* Retrieved February 07, 2016, from http://bie.org/about/what_pbl

Blay-Fornarino, M, Charfi A., Emsellem D., Pinna-Dery A., & Riveill M. (2004). Software interactions. *Journal of Object Technology*, 3(10), 161–180. Retrieved from http://www.jot.fm/issues/issues 2004 11/article4

Breiman, L. (2001). Random forests, *Machine Learning* 45(1), 5-32.

Breiman, L & Cutler A. (n.d.). *Random forests Leo Breiman and Adele Cutler*. Retrieved April 06, 2017, from https://www.stat.berkeley.edu/~breiman/RandomForests/reg_home.htm

Breiman L., Friedman J. H., Olshen R. A., & Stone, C. J. (1984). *Classification and regression trees*. Wadsworth.

Colón, M., Collet. M., Cruz, G., Del Pilar, D., & Martínez, F. (2013). *Undergraduate research report: Developing implementation methods for process automation projects* [research report*]*. Department of Industrial Engineering, University of Puerto Rico, Mayagüez, PR.

Collado, A., Medina, L.A. & Soto, Z. (2016). *Undergraduate research report: complexity metrics-VCI and V(G)* [research report*]*. Department of Industrial Engineering, University of Puerto Rico, Mayagüez, PR.

Crespo-Varela, J.R., Tucker, C.S., Okudan-Kremer, G.E. & Medina, L.A. (2012). An analysis of complexity measures in product design and development. *Proceedings of the ASME 2012 International Design Engineering Technical Conferences*, 3, 523-532.

Davis, J. H. (1969). Individual-group problem solving, subject preference, and problem type. *Journal of Personality and Social Psychology*, 13, 362-374.

DeMarco, T. (1982). *Controlling software projects: Management, measurement, and estimates* (1st ed.).

Doppelt, Y. (2003). Implementation and assessment of project-based. *International Journal of Technology and Design Education*, *13*, 255-272.

Faulconbridge, R. I., & Ryan, M. J. (2003). *Managing complex technical projects: a systems engineering approach*. Boston: Artech House.

Fast-Berglund, Å, Fässberg, T., Hellman, F., Davidsson, A., & Stahre, J. (2013). Relations between complexity, quality and cognitive automation in mixed-model assembly. *Journal of Manufacturing Systems*, 32(3), 449-455.

Friedman, J. (2001). Greedy function approximation: the gradient boosting machine, *Ann. of Stat.*

Genuer, R., Poggi, J., & Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, *31*(14), 2225-2236. doi:10.1016/j.patrec.2010.03.014

Gottfredson, M., & Rigby, D. (2009). *The power of managing complexity*. Retrieved July 30, 2016, from http://www.bain.com/publications/articles/the-power-of-managing-complexity.aspx

Gul, S. & S. Khan (2011). Revisiting project complexity: Towards a comprehensive model of project complexity. *2nd International Conference on Construction and Project Management*. Singapore, 15, 148-155.

Guzzo, R. A., & Shea, G. P. (1990). Group Performance and intergroup relations in organizations. *In Handbook of Industrial and Organizational Psychology* (2nd ed., Vol. 3, pp. 269-313). Palo Alto, CA: Consulting Psychologist Press.

Gransberg, D. D., Shane, J. S., Strong, K., & Lopez, C. (2013, October). Project complexity mapping in five dimensions for complex transportation projects. *Journal of Management in Engineering, 29*, 316-326. Retrieved March 3, 2015, from Ascelibrary.org.

Halstead, M. H. (1977). *Elements of software science*. New York: Elsevier North-Holland.

H. Hotelling. (1933). Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.*, 24 (6), 417–441.

Hussein, B. A., Pigagaitea, G., & Silva, P. P. (2014). Identifying and dealing with complexities in new product and process development projects. *Procedia - Social and Behavioral Sciences, 119*, 702-710. Retrieved February 8, 2016, from ScienceDirect.

Higgs M., Plewnia U., & Ploch J. (2005). Influence of team composition and task complexity on team performance. *Team Performance Management: An International Journal*, Vol. 11 Issue: 7/8, pp.227-250, doi: 10.1108/13527590510635134

Hill, G. W. (1982). *Group versus individual performance: Are N + 1 heads better than one? Psychological Bulletin*, 91 (3), 517-539.

Indiramma, M (2014). Project based learning – Theoretical foundation. *International Conference on Interactive Collaborative Learning (ICL)*, 841-844. Retrieved November 16, 2016, from IEEE

Ireland, L. (2007, October). *Project complexity: A brief exposure to difficult situations*. Retrieved April 1, 2017, from www.ipma-usa.org

Intel, (2007). *Designing Effective Projects: Characteristics of projects*. Retrieved December 22, 2016, from http://download.intel.com/education/common/pk/resources/dep/projectdesign/dep_pbl_research.pdf

Jusino, C., Medina, L.A. & Soto, Z. (2016). *Undergraduate research report: Complexity metrics-PHF and SHF* [research report]. Department of Industrial Engineering, University of Puerto Rico, Mayagüez, PR.

Kabacoff, R. (2017). *Tree-Based models*. Retrieved April 06, 2017, from http://www.statmethods.net/advstats/cart.html

Keating, M. (2000). Measuring design quality by measuring design complexity. *Proceedings of the IEEE 2000 First International Symposium on Quality Electronic Design*, 103-108.

Krajcik, J. S., & Blumenfeld, P. C. (2006). Chapter 19: Project-based learning. In R. K. Sawyer (Ed.), *The Cambridge Handbook of the Learning Sciences* (pp. 317-334). Cambridge University Press.

Lewis, J. P. (1999). *The project manager's desk reference* (2nd ed.). McGraw Hill Professional**.**

Liaw, A., & Wiener, M. (2002). *Classification and regression by randomForest*. *R News, 3*, 18-22.

Martínez, E., Jusino, C., Medina, L. A. & Soto, Z. (2016). Undergraduate research report: complexity metrics SCI vs. SCIS [research report]. Department of Industrial Engineering, University of Puerto Rico, Mayagüez, PR.

Medina, L. A. (2013). *Real time process control* [course syllabus]. Department of Industrial Engineering, University of Puerto Rico, Mayaguez, PR.

Medina, L. A., Colón, M., Cruz, G., Collet, M., & Soto, Z. (2015). Measuring the complexity of design projects, *Proceedings of the 2015 Industrial and Systems Engineering Research Conference*.

Mathieson, J. L., Wallace, B. A., & Summers, J. D. (2010). Assembly time modeling through connective complexity metrics. *2010 International Conference on Manufacturing Automation*. doi:10.1109/icma.2010.21

McQuilken, T. (2014). *Automation is the future of print workflows*. Editor & Publisher, 147(1), 32-33. Retrieved from http://search.proquest.com/docview/1498026957?accountid=28498

McCabe, T. (1976) A complexity measure. *IEEE Transactions on Software Engineering*, 2(4), 308-320.

Owens, J., Ahn, J., Shane, J. S., Strong, K. C., & Gransberg, D. D. (2011). Defining complex project management of large U.S. transportation projects: A comparative case study analysis. *Public Works Management & Policy*, 17(2), 170-188. doi:10.1177/1087724x11419306

Project Management Institute, Inc (2016). *What is project management?* Retrieved July 30, 2016, from https://www.pmi.org/about/learn-about-pmi/what-is-project-management

Sam Houston State University (2016). *Project based learning in higher education*. Retrieved February 07, 2016, from http://www.shsu.edu/centers/project-based-learning/higher-education.html

R Development Core Team. (2010). *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria.

Remington, K., R. Zolin & R. Turner (2009). A model of project complexity: distinguishing dimensions of complexity from severity. *Proceedings of the 9th International Research Network of Project Management Conference, IRNOP*.

Roy, R., Evans, R., Low, M.J., & Williams, D.K. (2011). Addressing the impact of high levels of product variety on complexity in design and manufacture. *Journal of Engineering Manufacture*, 225, 1939-1950.

Sonnert, G., & Fox, M. F. (2012, January/February). Women, men, and academic performance in science and engineering: The gender difference in undergraduate grade point averages. *The Journal of Higher Education*, 83, 73-101. Retrieved February 5, 2015.

Soto, Z., Rosado, M., & Medina, L. (2015). *Undergraduate research report: developing implementation methods for process automation projects* [research report]. Department of Industrial Engineering, University of Puerto Rico, Mayagüez, PR.

Tessema, M., Ready, K., & Malone, C. (2012). Effect of gender on college students' satisfaction and achievement: The case of a midsized midwestern public university. *International Journal of Business and Social Science*, 3. Retrieved December 28, 2015, from www.ijbssnet.com.

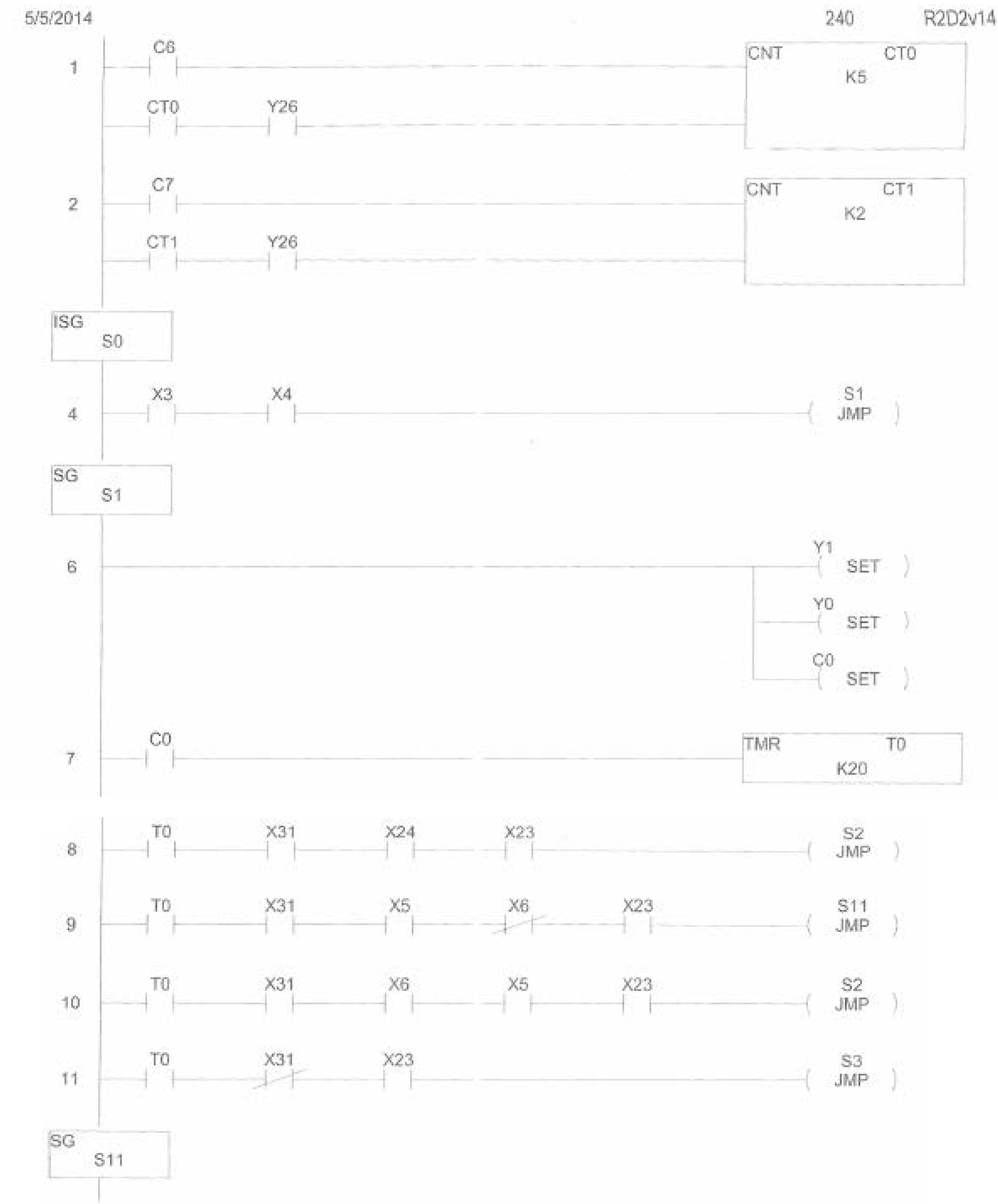
Thomas, J. W. (2000, March). A Review of research on project based learning | project based learning | BIE. Retrieved July 30, 2016, from http://bie.org/object/document/a_review_of_research_on_project_based_learning

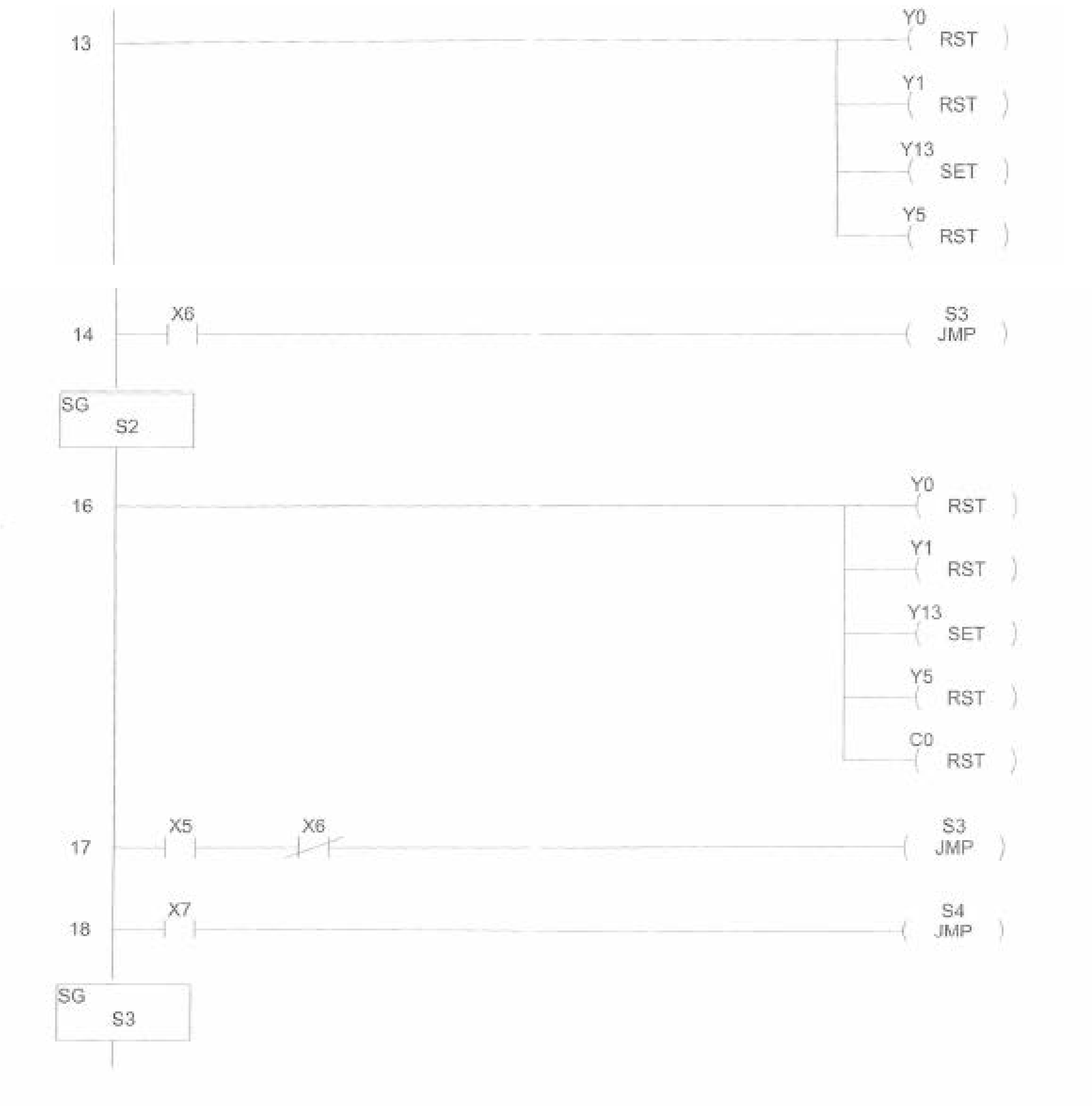
Vega, V. (2015, December 1). *Project-based learning research review*. Retrieved July 31, 2016, from http://www.edutopia.org/project-based-learning

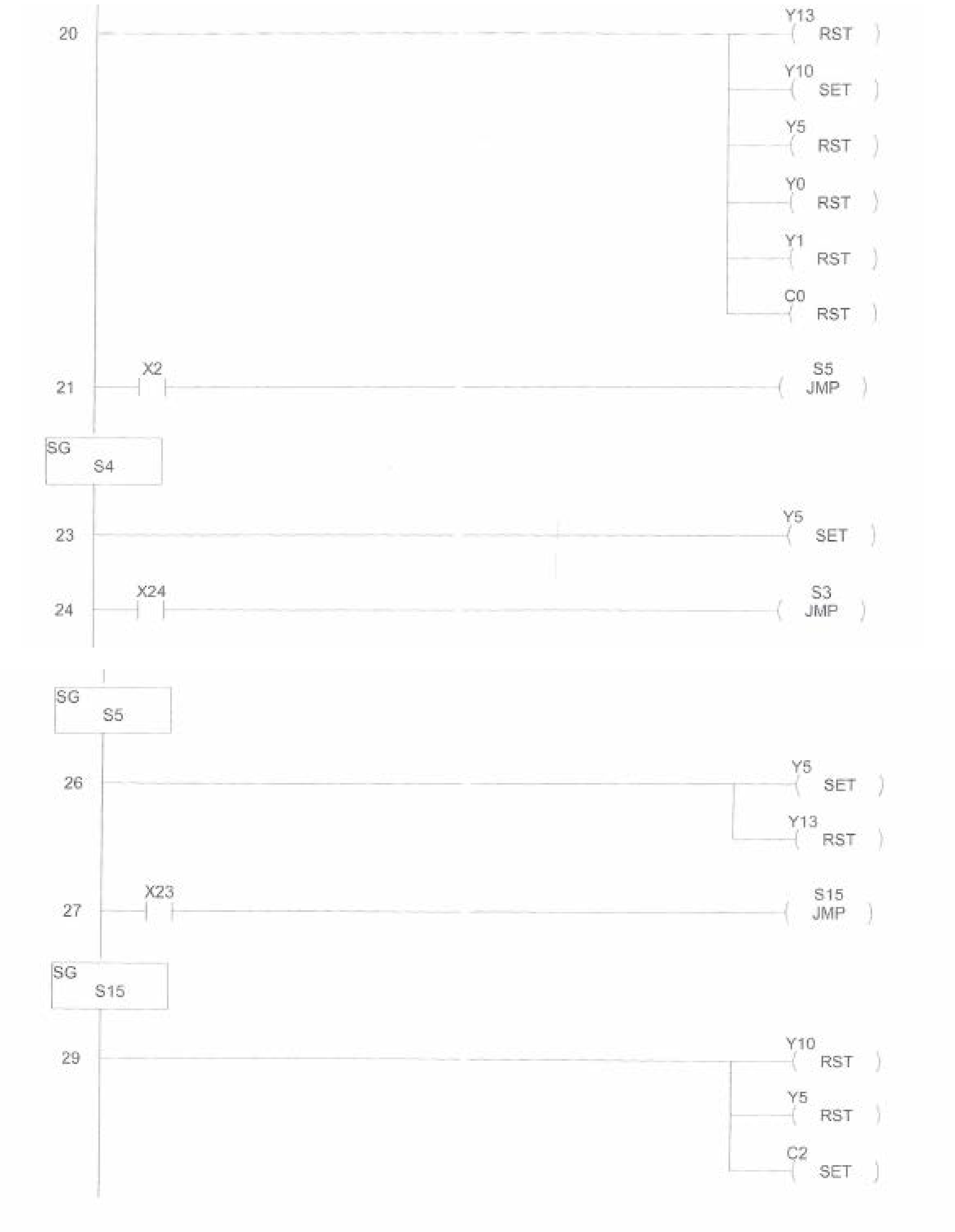
Webb, Noreen M., Kariane Mari Nemer & Zuniga, Stephen (2002). Short circuits or superconductors? Effects of group composition on high-achieving student's science assessment performance. *American Educational Research Journal*, 39(4), 943-989.

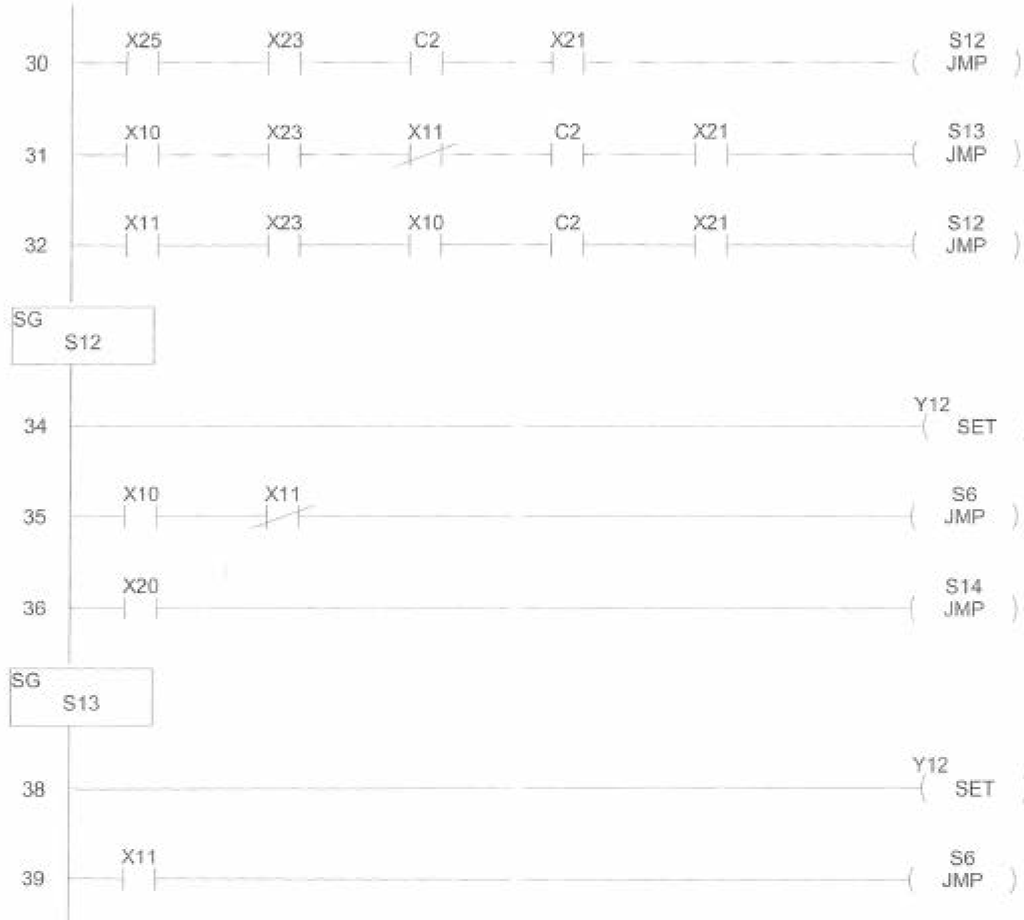
Yu, S., & Zhou, S. (2010). A survey on metric of software complexity. *2010 2nd IEEE International Conference on Information Management and Engineering*. doi:10.1109/icime.2010.5477581

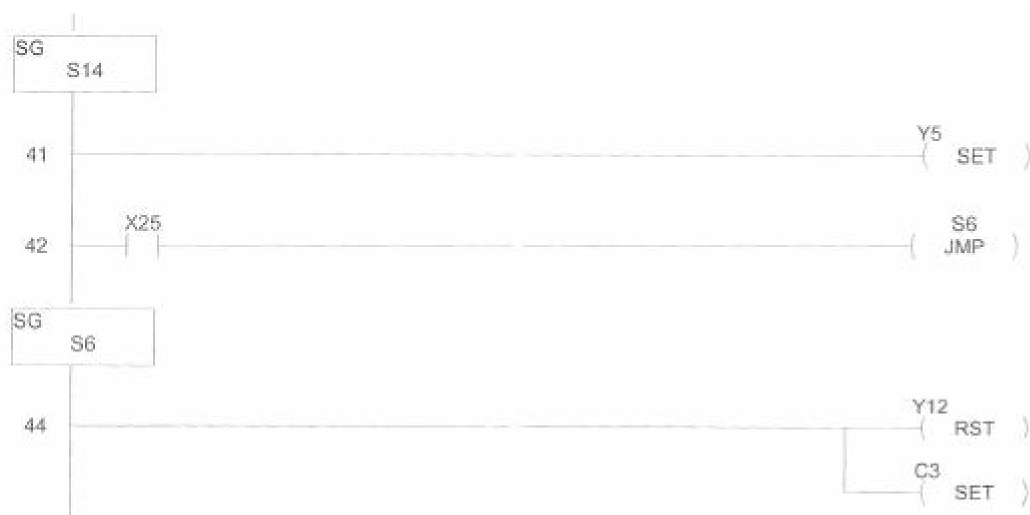
Appendix A: R2D2 Ladder Logic

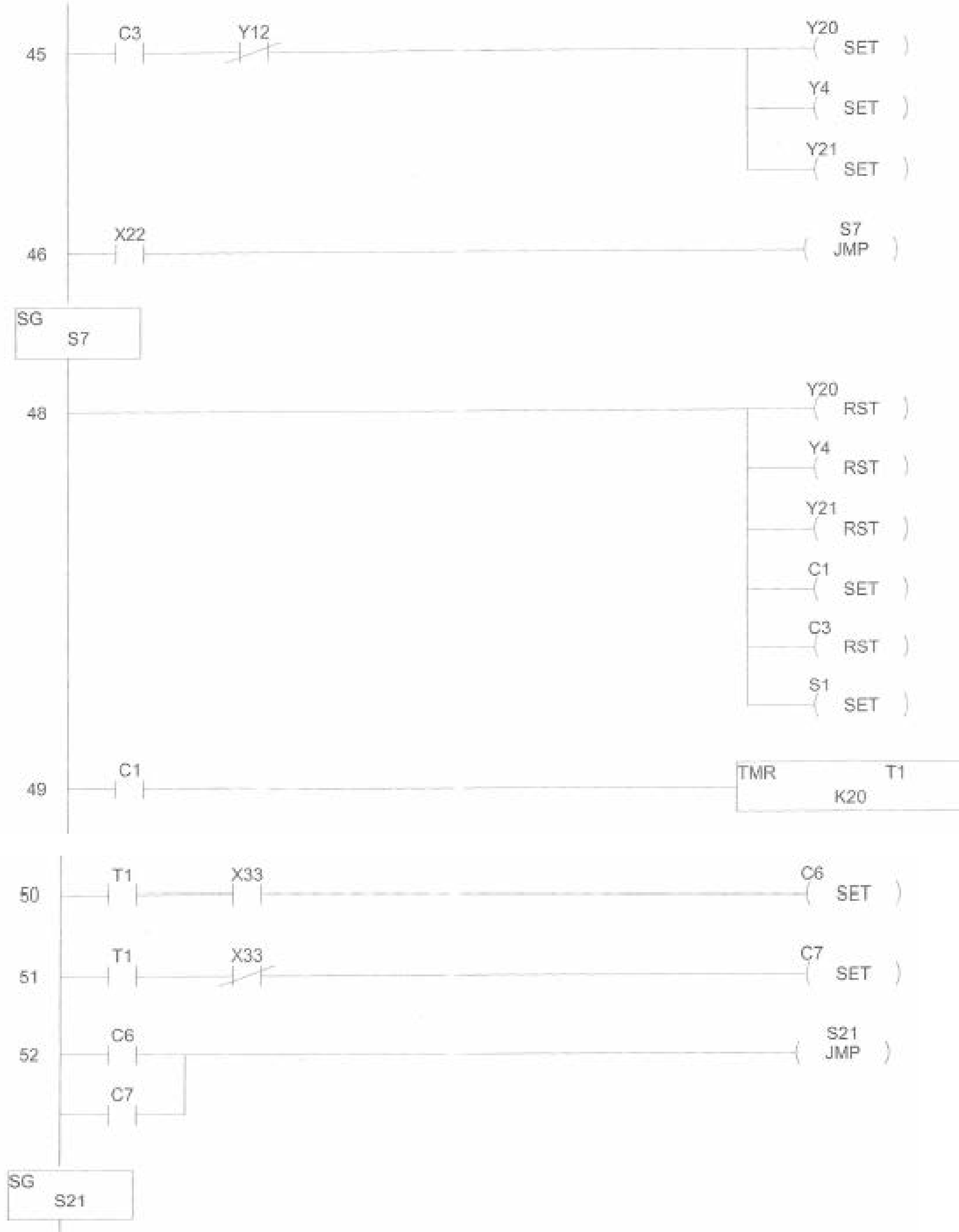


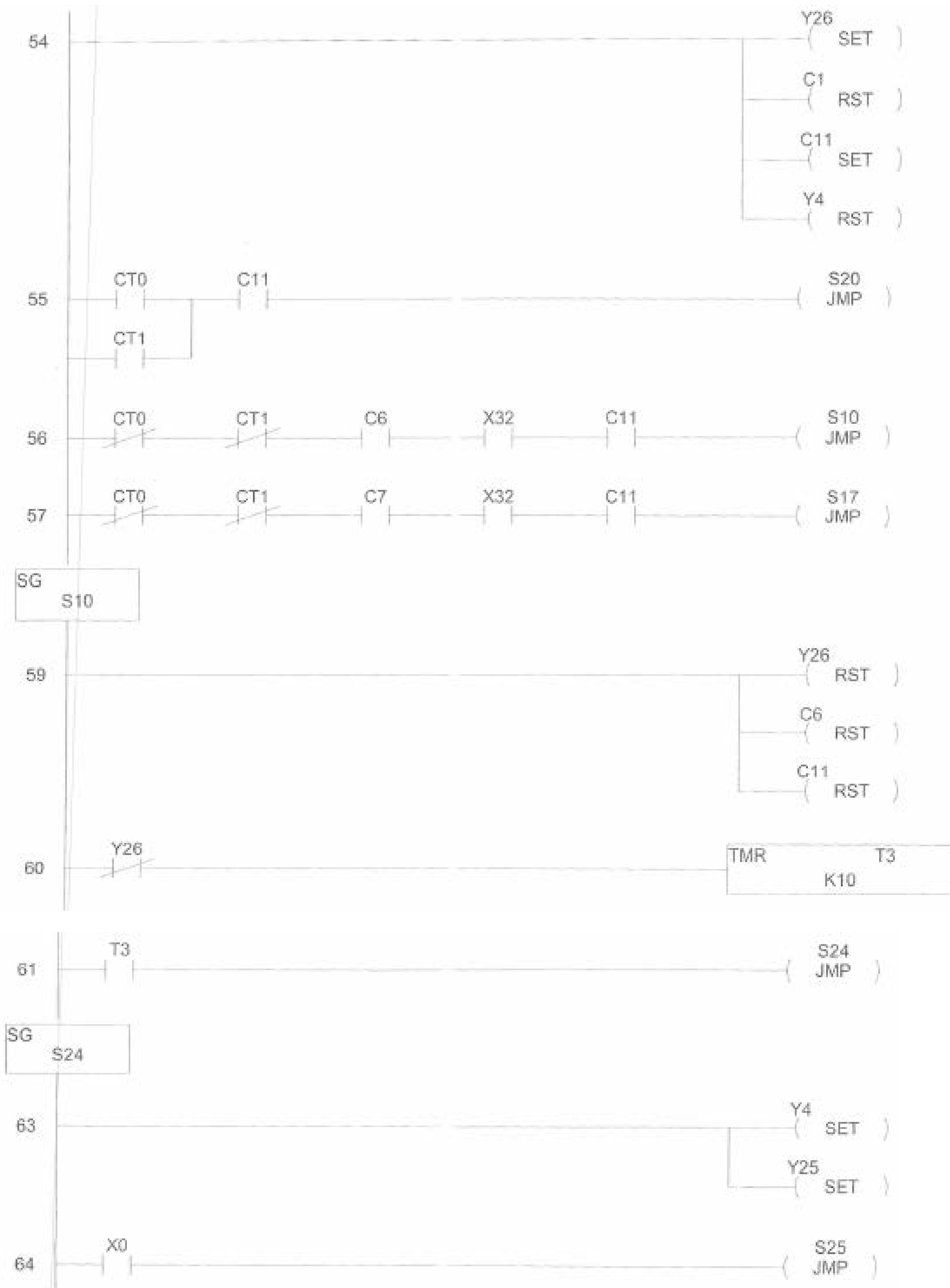

13
Y0
RST
Y1
RST
Y13
SET
Y5
RST
14
X6
S3
JMP
SG
S2
16
Y0
RST
Y1
RST
Y13
SET
Y5
RST
C0
RST
17
X5
X6
S3
JMP
18
X7
S4
JMP
SG
S3


20
Y13 RST
Y10 SET
Y5 RST
Y0 RST
Y1 RST
C0 RST
21
X2
S5 JMP
SG S4
23
Y5 SET
24
X24
S3 JMP
SG S5
26
Y5 SET
Y13 RST
27
X23
S15 JMP
SG S15
29
Y10 RST
Y5 RST
C2 SET

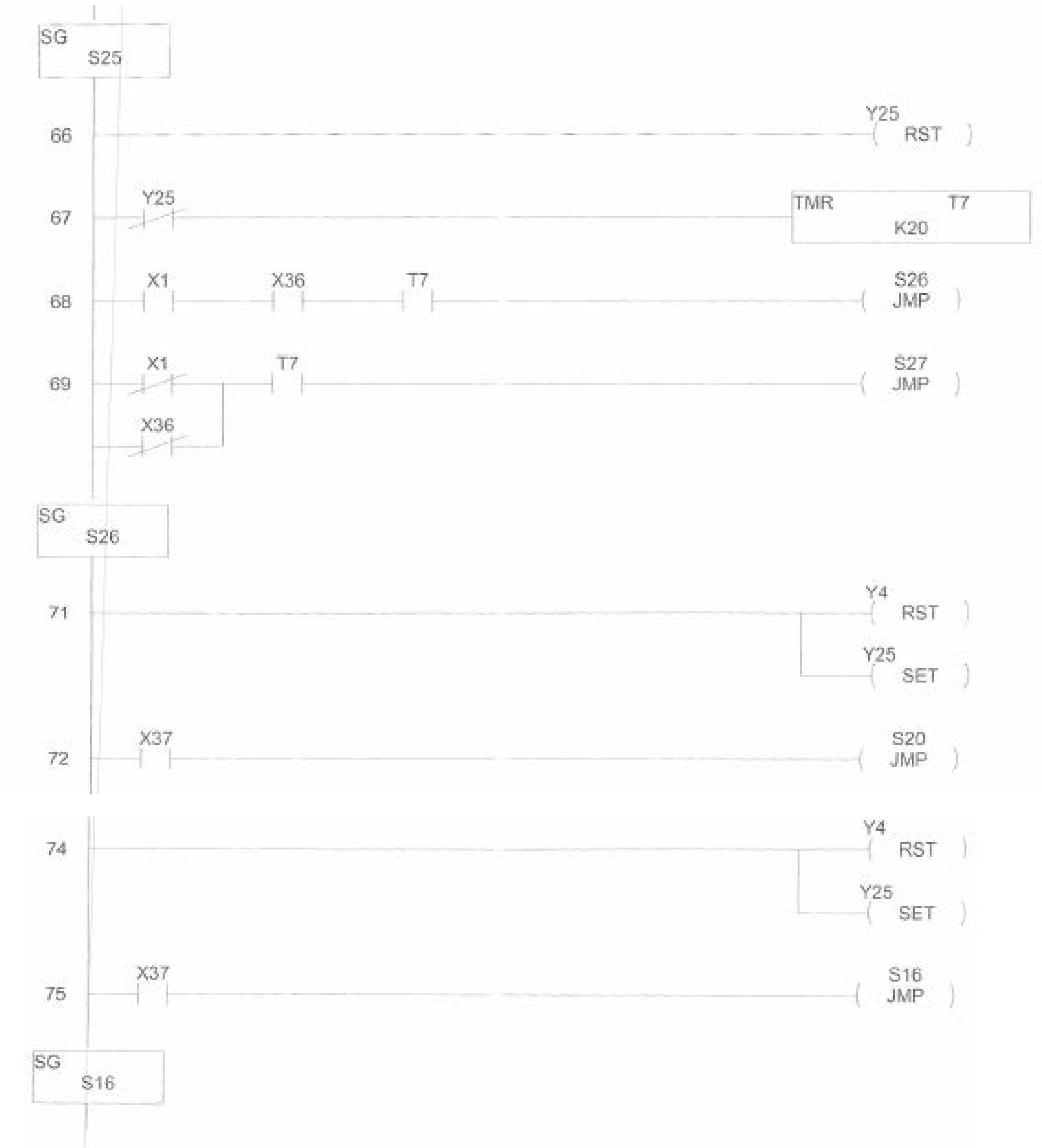

30
X25
X23
C2
X21
S12
JMP
31
X10
X23
X11
C2
X21
S13
JMP
32
X11
X23
X10
C2
X21
S12
JMP
SG
S12
34
Y12
SET
35
X10
X11
S6
JMP
36
X20
S14
JMP
SG
S13
38
Y12
SET
39
X11
S6
JMP

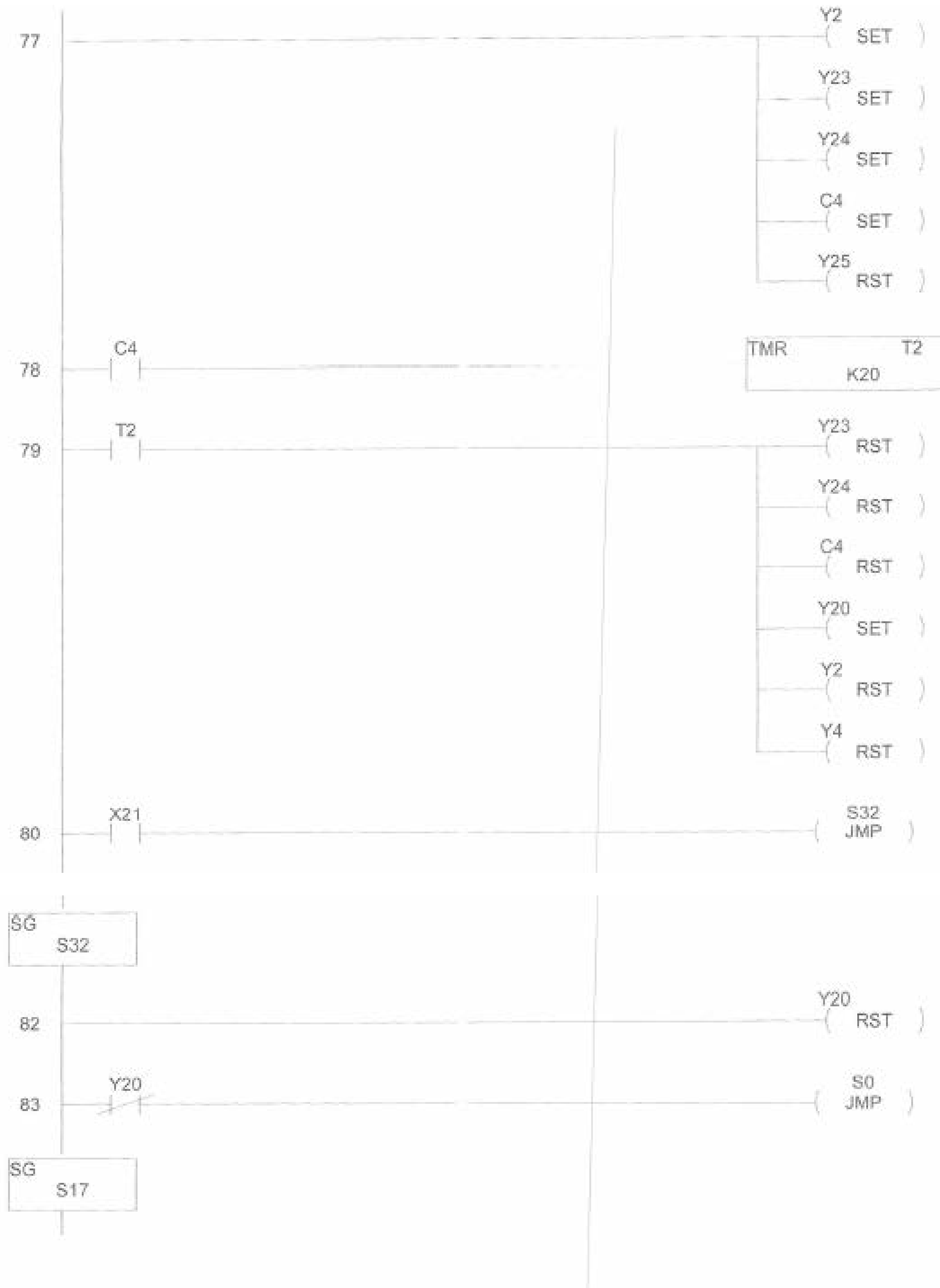

SG
S14
41
Y5
SET
42
X25
S6
JMP
SG
S6
44
Y12
RST
C3
SET


45
C3
Y12
Y20 SET
Y4 SET
Y21 SET
46
X22
S7 JMP
SG S7
48
Y20 RST
Y4 RST
Y21 RST
C1 SET
C3 RST
S1 SET
49
C1
TMR T1 K20
50
T1
X33
C6 SET
51
T1
X33
C7 SET
52
C6
C7
S21 JMP
SG S21

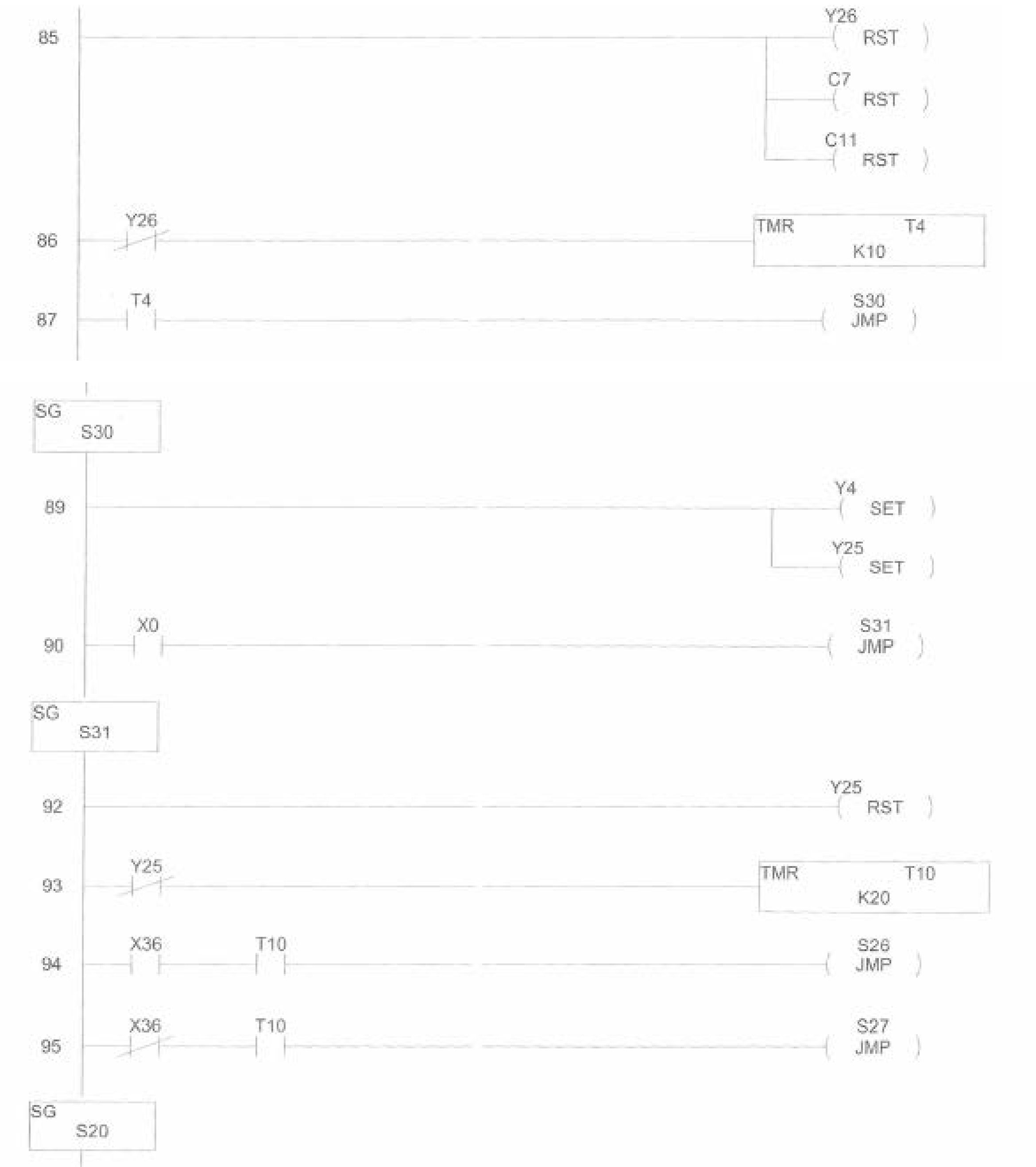


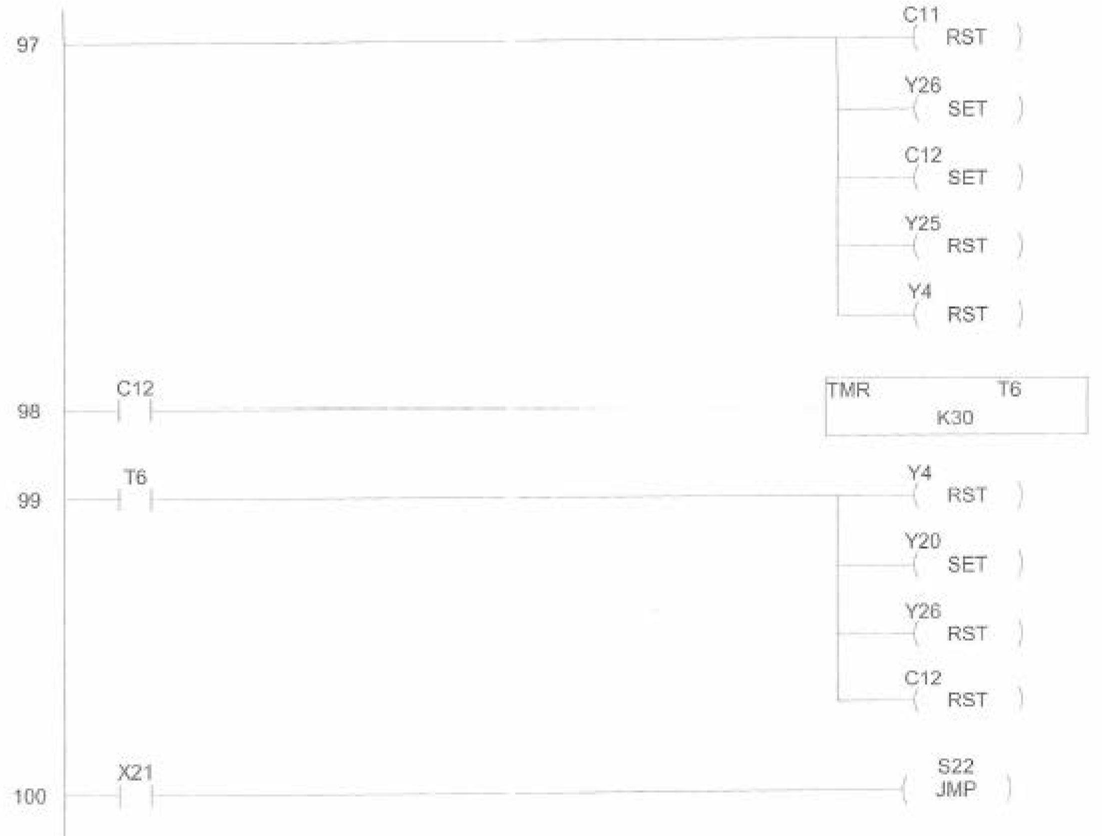
54
Y26 SET
C1 RST
C11 SET
Y4 RST
55
CT0
C11
S20 JMP
CT1
56
CT0
CT1
C6
X32
C11
S10 JMP
57
CT0
CT1
C7
X32
C11
S17 JMP
SG S10
59
Y26 RST
C6 RST
C11 RST
60
Y26
TMR T3 K10
61
T3
S24 JMP
SG S24
63
Y4 SET
Y25 SET
64
X0
S25 JMP

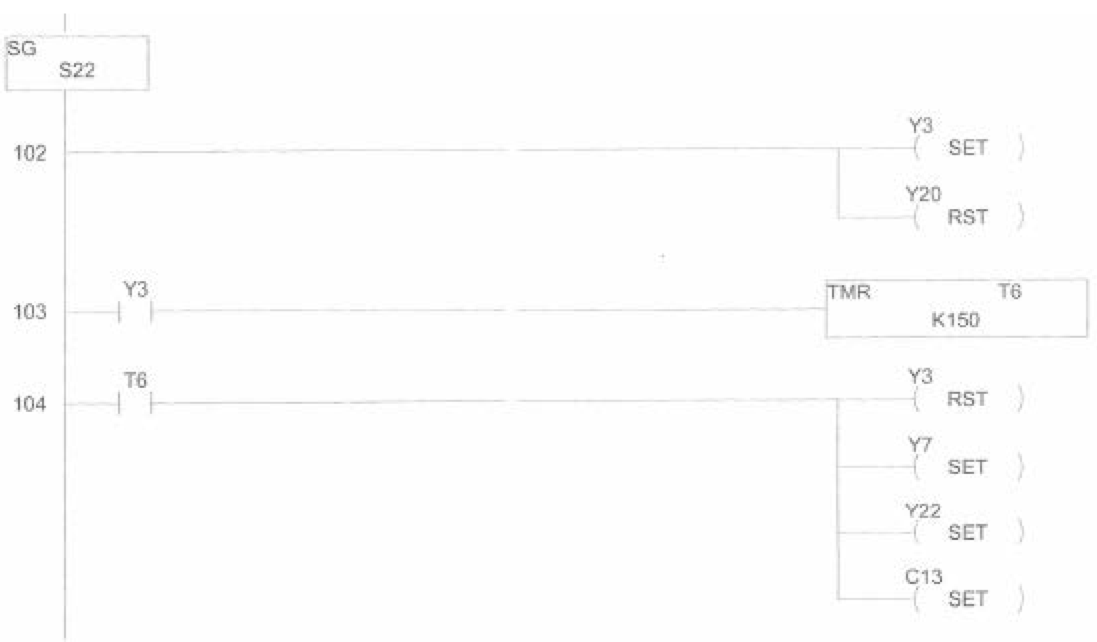

SG
S25
66
Y25
RST
67
Y25
TMR
T7
K20
68
X1
X36
T7
S26
JMP
69
X1
T7
S27
JMP
X36
SG
S26
71
Y4
RST
Y25
SET
72
X37
S20
JMP
74
Y4
RST
Y25
SET
75
X37
S16
JMP
SG
S16



77
Y2 SET
Y23 SET
Y24 SET
C4 SET
Y25 RST
78
C4
TMR T2
K20
79
T2
Y23 RST
Y24 RST
C4 RST
Y20 SET
Y2 RST
Y4 RST
80
X21
S32 JMP
SG S32
82
Y20 RST
83
Y20
S0 JMP
SG S17


85
Y26 RST
C7 RST
C11 RST
86
Y26
TMR T4
K10
87
T4
S30 JMP
SG S30
89
Y4 SET
Y25 SET
90
X0
S31 JMP
SG S31
92
Y25 RST
93
Y25
TMR T10
K20
94
X36
T10
S26 JMP
95
X36
T10
S27 JMP
SG S20


97
C11
RST
Y26
SET
C12
SET
Y25
RST
Y4
RST
98
C12
TMR
T6
K30
99
T6
Y4
RST
Y20
SET
Y26
RST
C12
RST
100
X21
S22
JMP


SG
S22
102
Y3
SET
Y20
RST
103
Y3
TMR
T6
K150
104
T6
Y3
RST
Y7
SET
Y22
SET
C13
SET

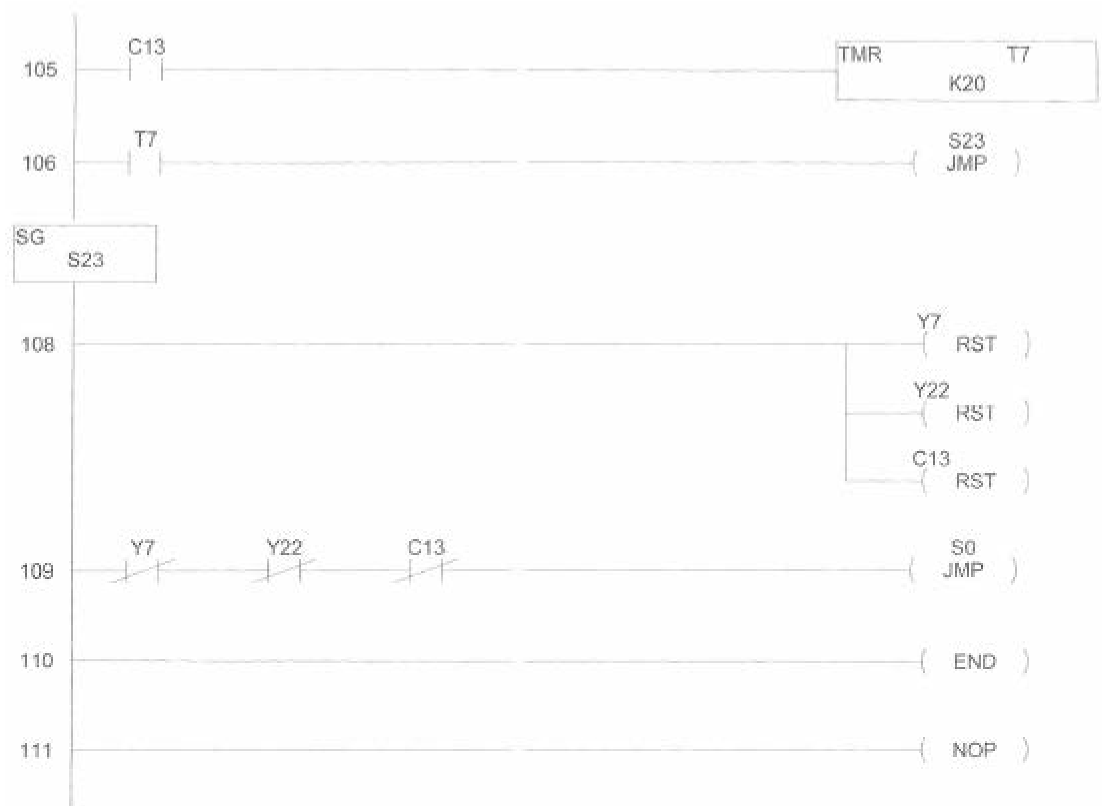


Figure A1: R2D2 Ladder Logic

Appendix B: SCI Metric Applied to R2D2

Making components-interaction diagrams for each line of code

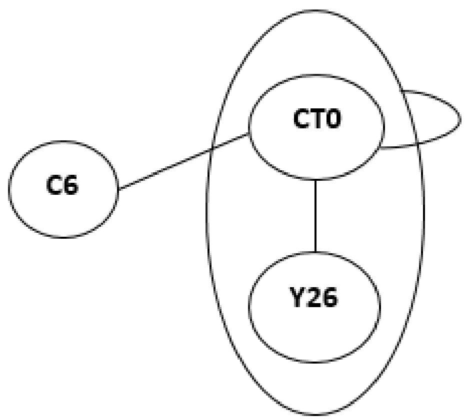
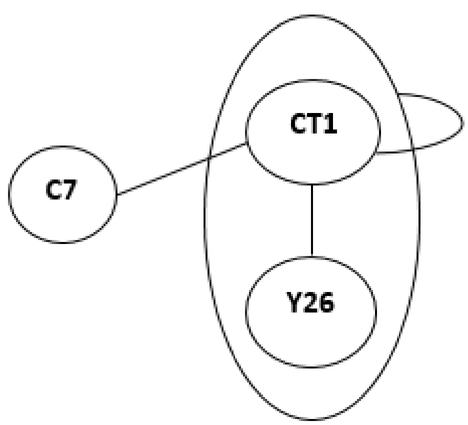
Table B1: Automated Process Components

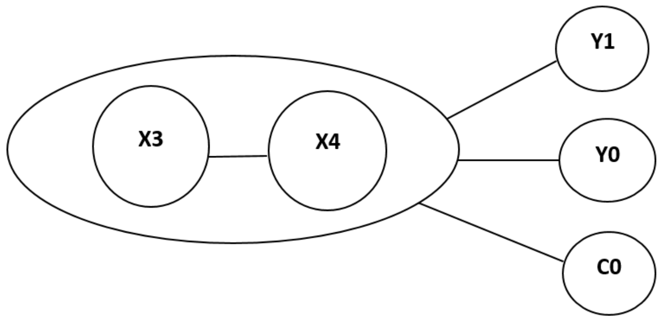
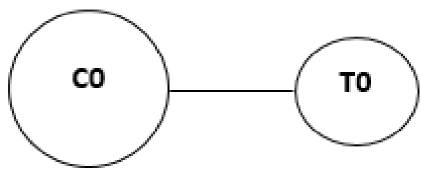
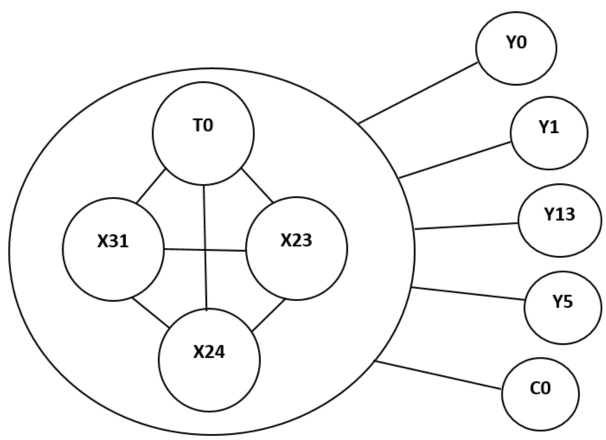
Components (M)				
X's	**Y's**	**C's**	**CT's**	**T's**
X0	Y0	C0	CT0	T0
X1	Y1	C1	CT1	T1
X2	Y2	C2		T2
X3	Y3	C3		T3
X4	Y4	C4		T4
X5	Y5	C6		T6
X6	Y7	C7		T7
X7	Y10	C11		T10
X10	Y12	C12		
X11	Y13	C13		
X20	Y20			
X21	Y21			
X22	Y22			
X23	Y23			
X24	Y24			
X25	Y25			
X31	Y26			
X32				
X33				

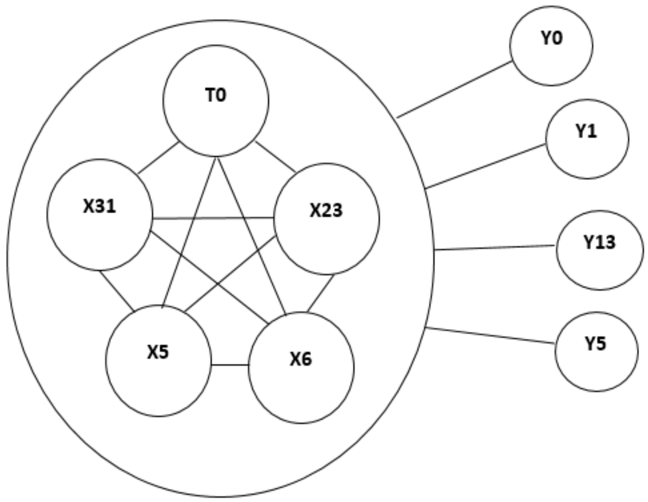
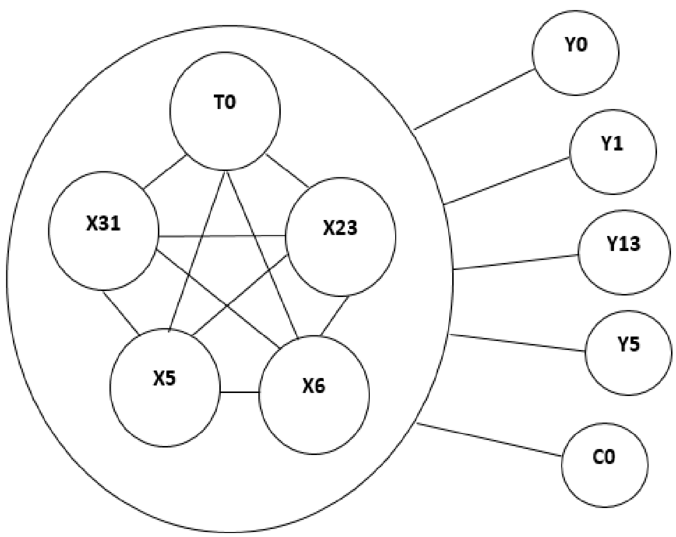
X36				
X37			**Total**	**58**

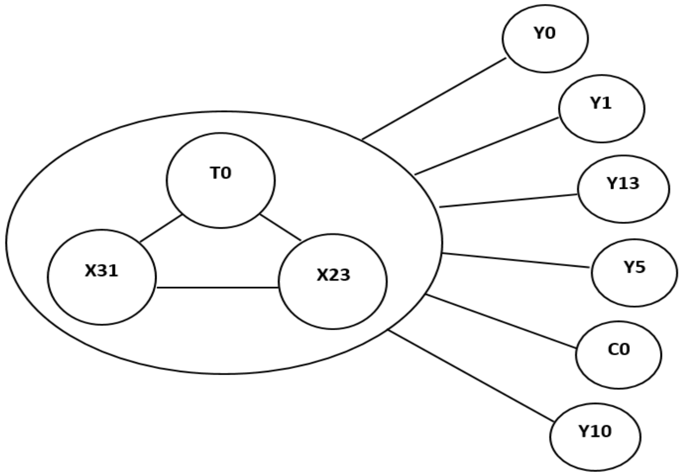
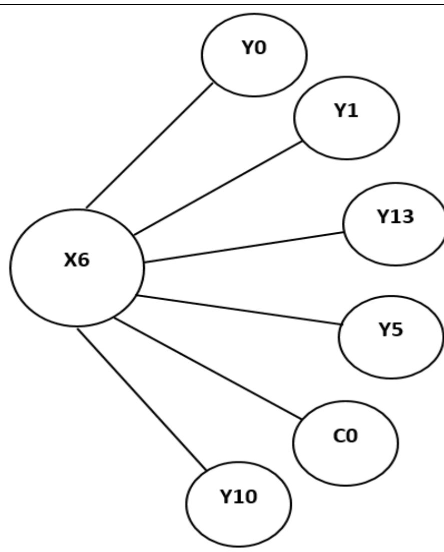
Line by Line Analysis

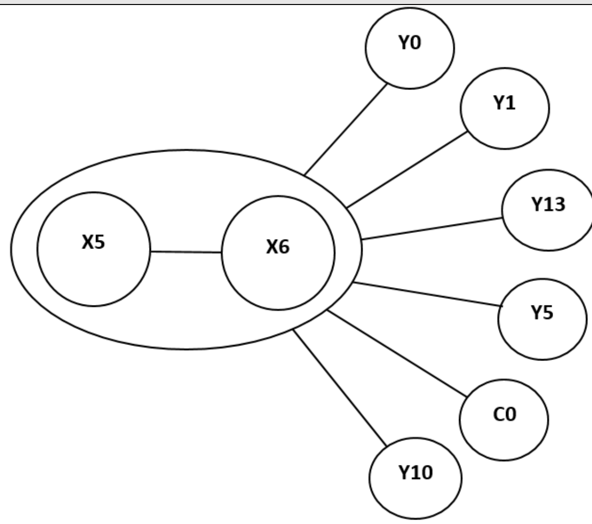
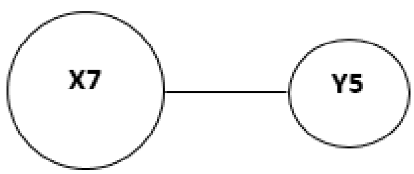
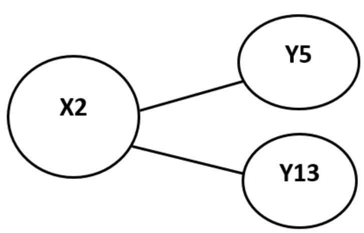
Table B2: Diagrams based on Lines of Code

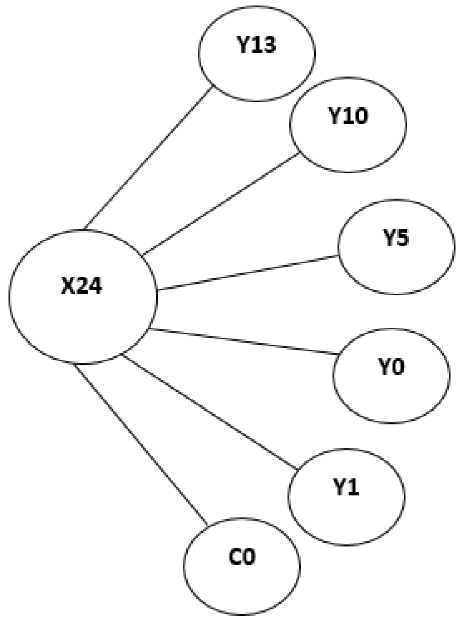
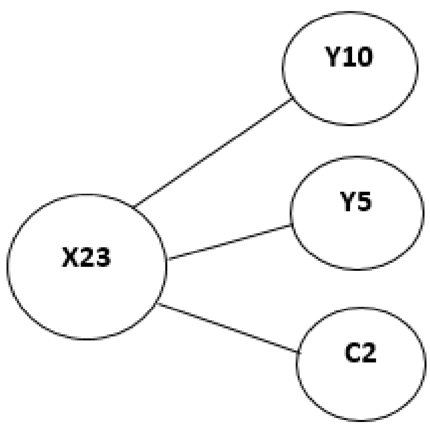
Line of Code	**Diagrams**	**Interaction**
1		4
2		4
4,6		5

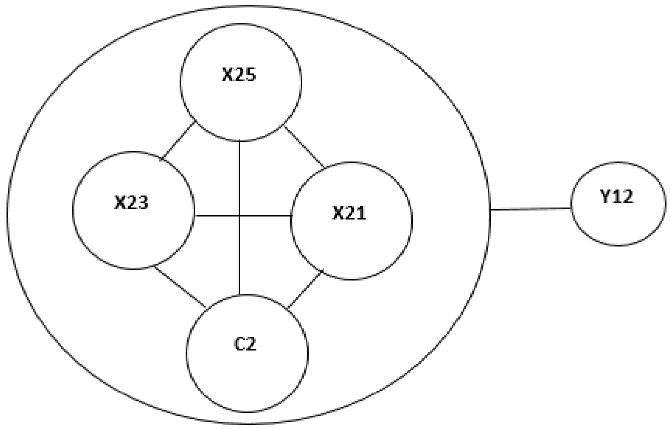
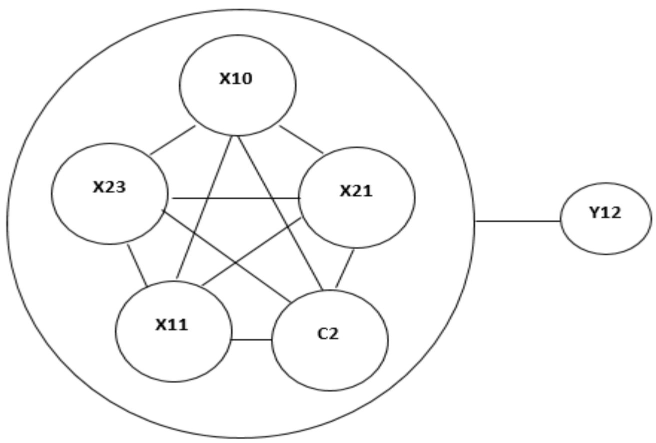
Line of Code	**Diagrams**	**Interaction**
		
7		1
8,16		12
9,13		15

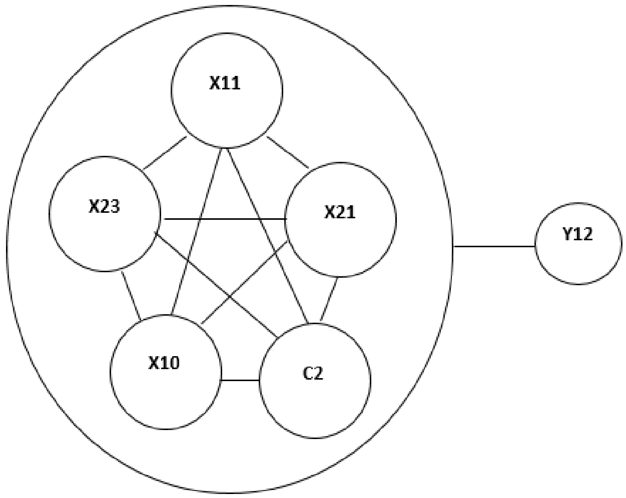
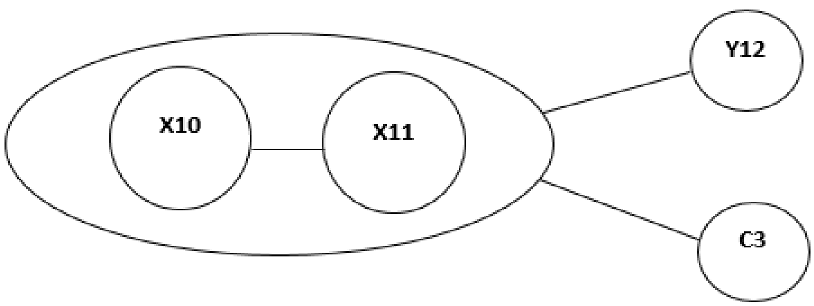
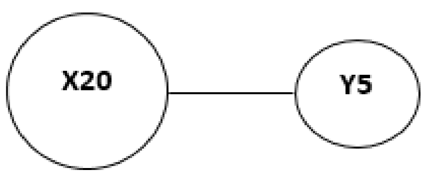
Line of Code	Diagrams	Interaction
		
10,16		16

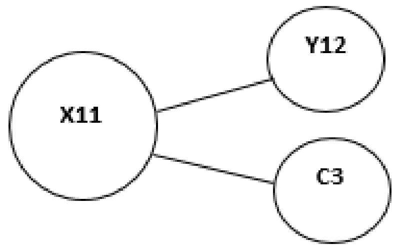
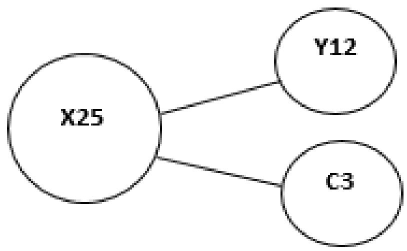
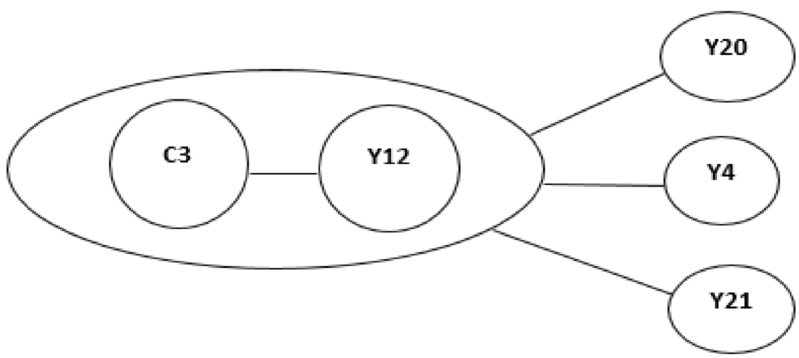
Line of Code	Diagrams	Interaction
11,20		10
14,20		6

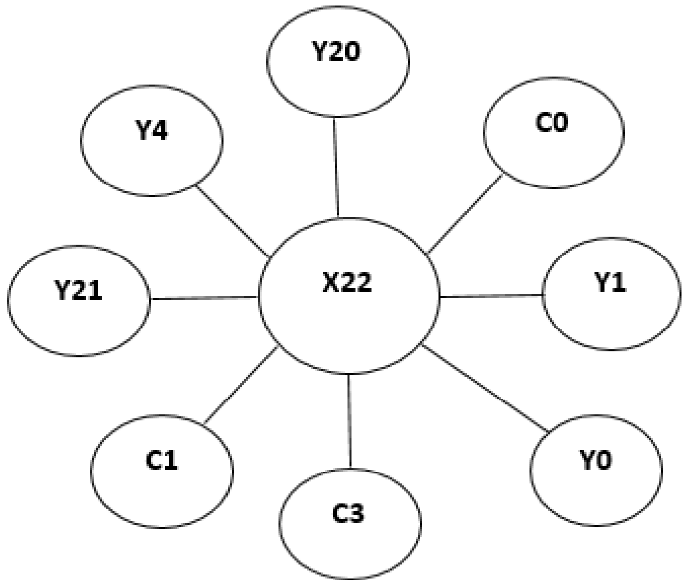
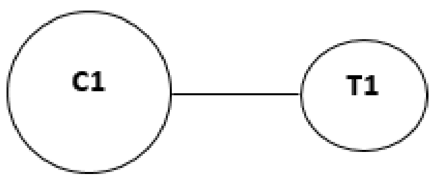
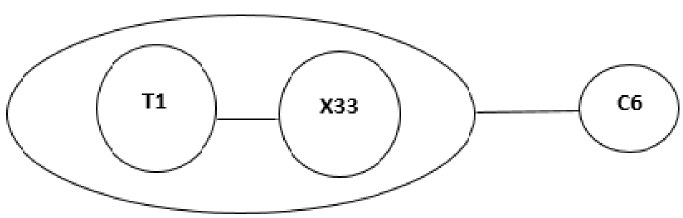
Line of Code	Diagrams	Interaction
17,20		8
18,23		1
21,26		2

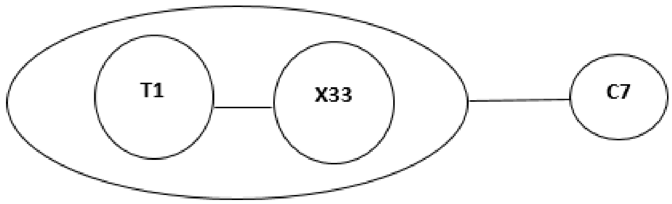
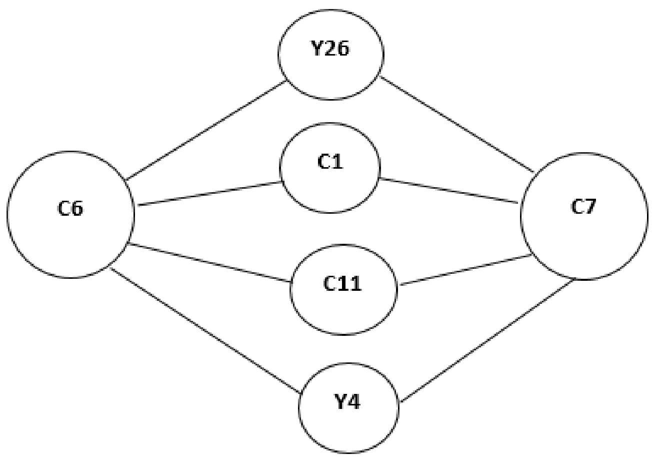
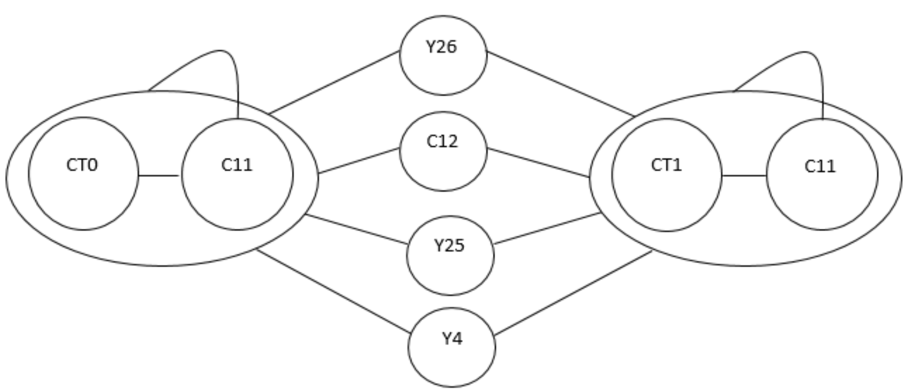
Line of Code	Diagrams	Interaction
24,20		6
27,29		3

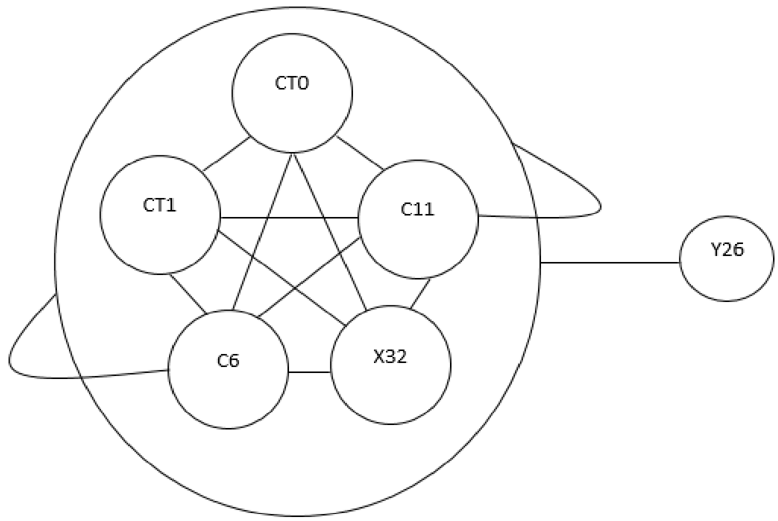
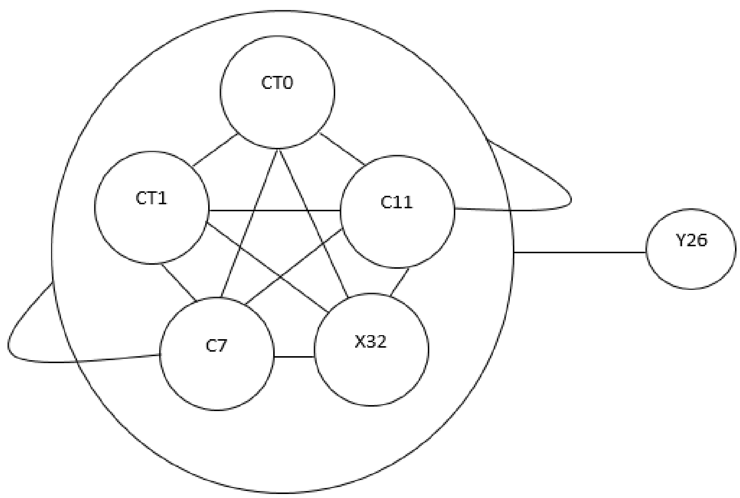
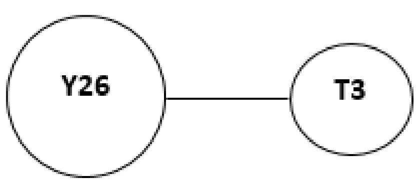
Line of Code	Diagrams	Interaction
30,34		8
31,38		12

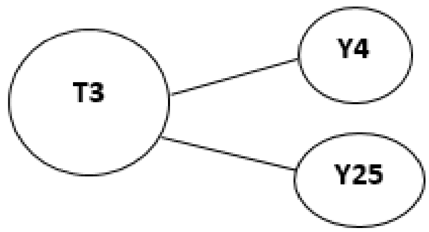
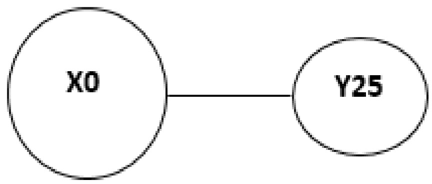
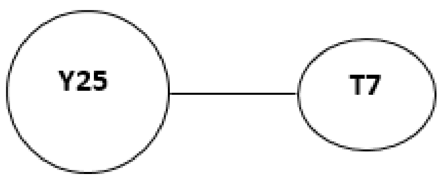
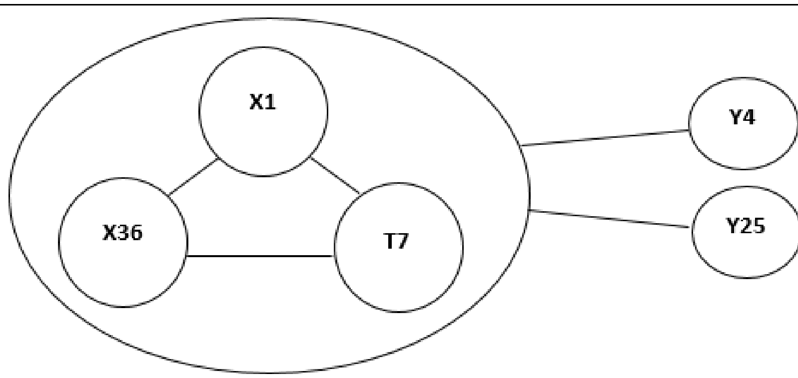
Line of Code	Diagrams	Interaction
32,34		12
35,44		4
36,41		1

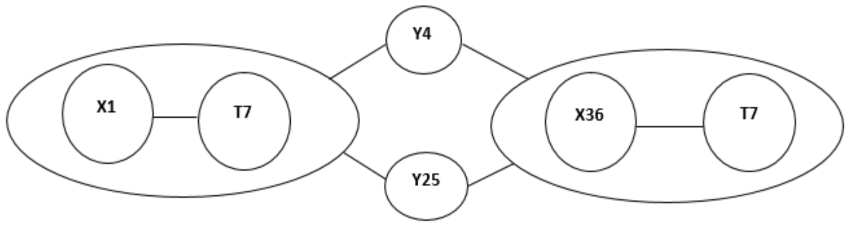
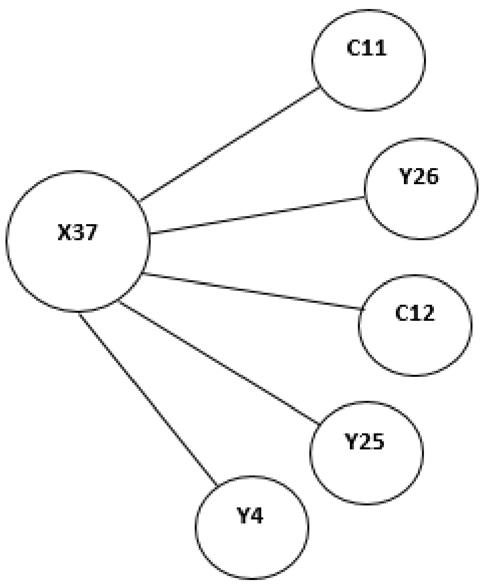
Line of Code	Diagrams	Interaction
39,44		2
42,44		2
45		5

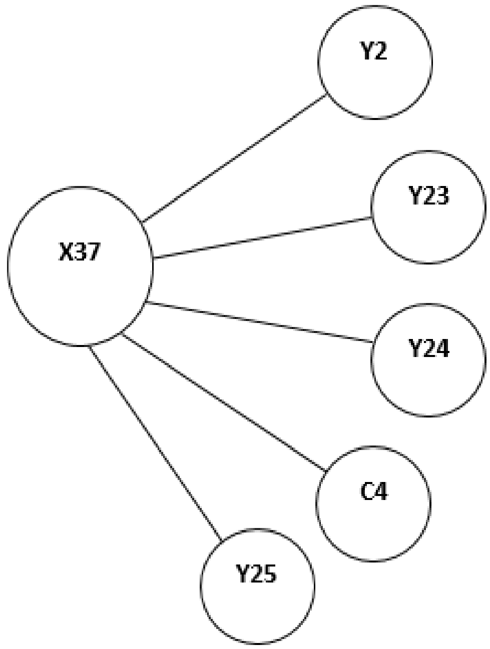
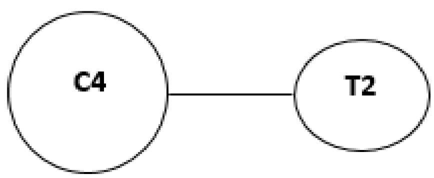
Line of Code	Diagrams	Interaction
46,48		8
49		1
50		3

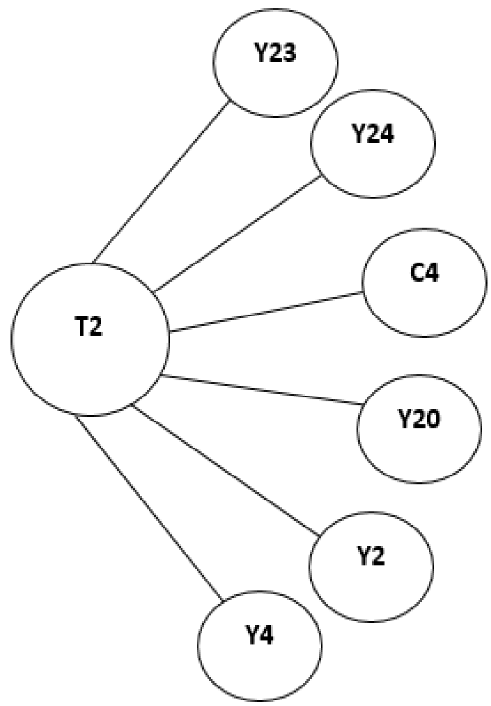
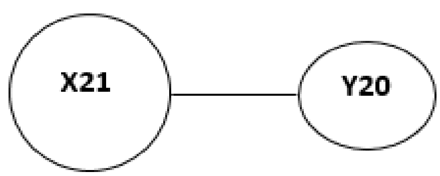
Line of Code	Diagrams	Interaction
51		3
52,54		8
55,97		14

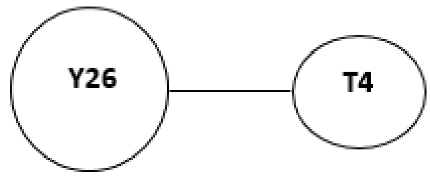
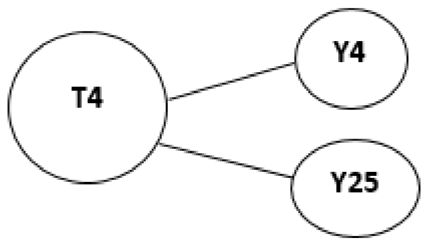
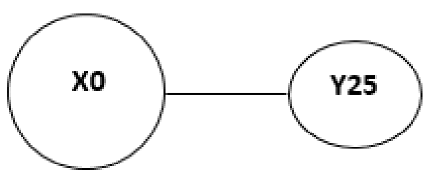
Line of Code	Diagrams	Interaction
56,59		14
57,85		14
60		1

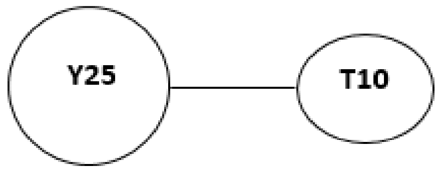
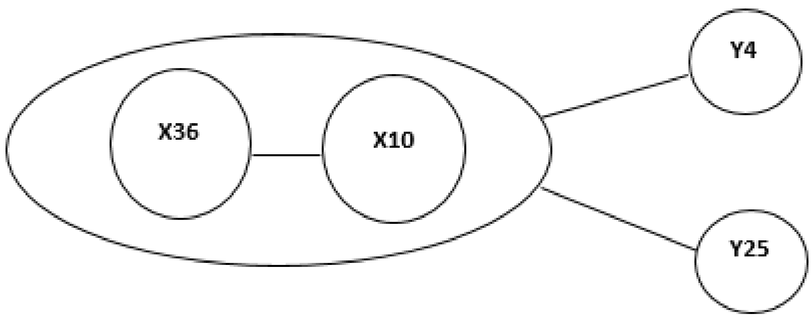
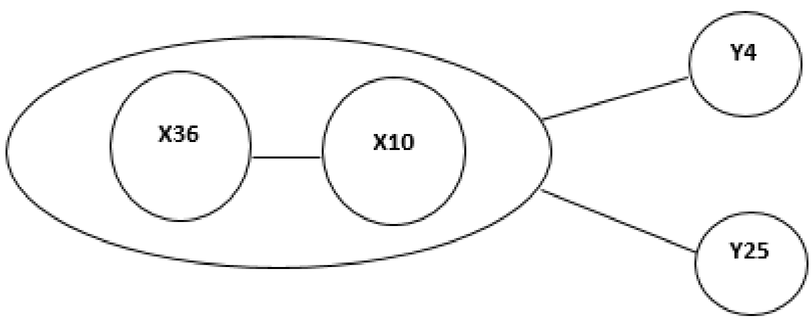
Line of Code	Diagrams	Interaction
61,63		2
64,66		1
67		1
68,71		6

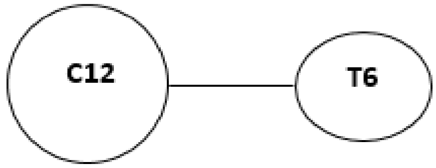
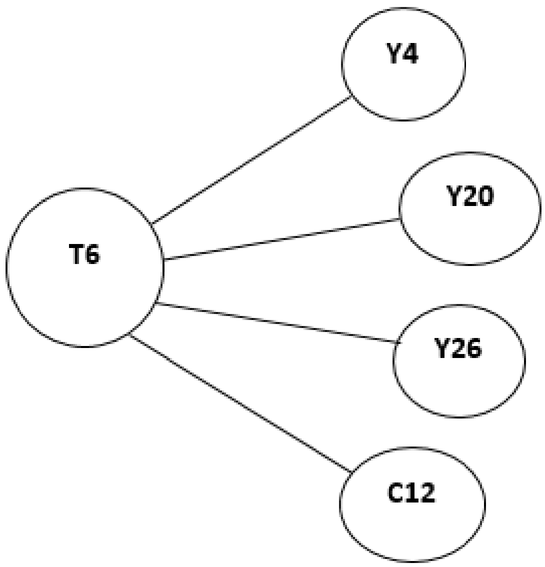
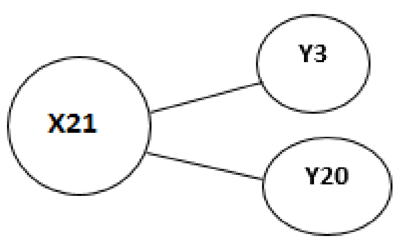
Line of Code	Diagrams	Interaction
69,74		8
72,97		5

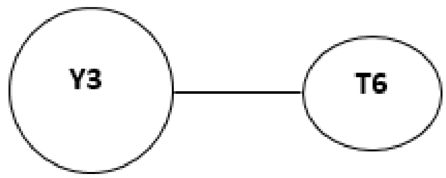
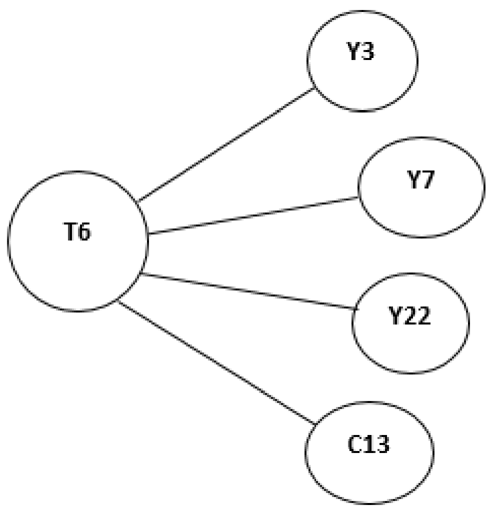
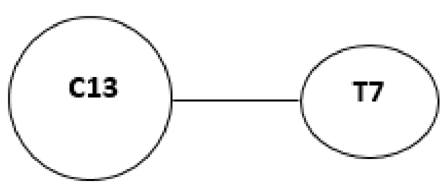
Line of Code	Diagrams	Interaction
75,77		5
78		1

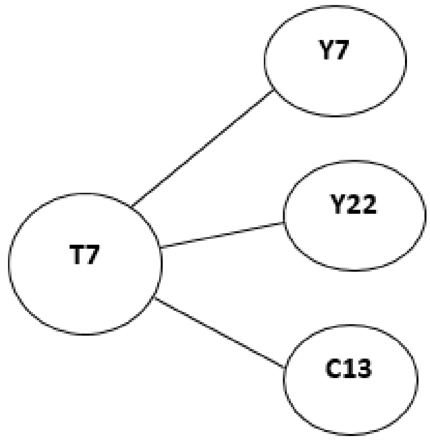
Line of Code	Diagrams	Interaction
79		6
80,82		1

Line of Code	Diagrams	Interaction
86		1
87,89		2
90,92		1

Line of Code	Diagrams	Interaction
93		1
94,71		4
95,74		4

Line of Code	Diagrams	Interaction
98		1
99		4
100,102		2

Line of Code	Diagrams	Interaction
103		1
104		4
105		1

Line of Code	Diagrams	Interaction
106,108		3
Total Interactions		**270**

Eliminating redundant diagrams:

Table B3: Elimination of Redundant Diagrams Based on Lines of Codes

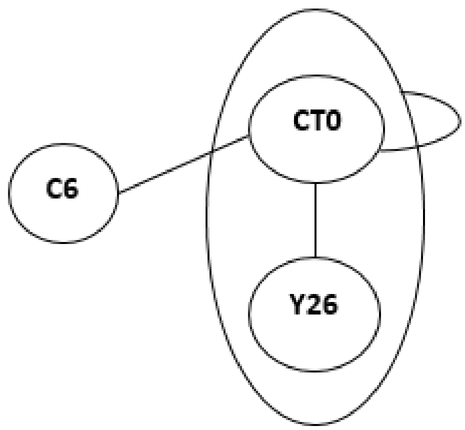
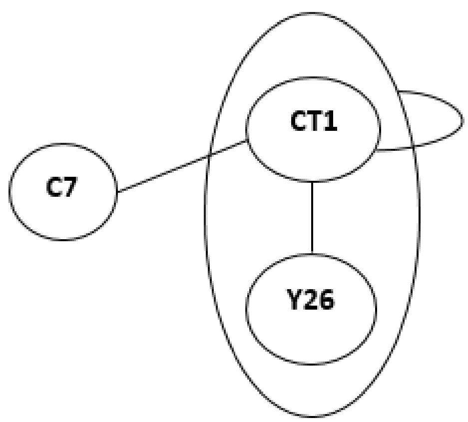
Lines of Code	Interaction	Lines of Code Comparison	Interaction	Final Lines of Code	Final Interaction
1	4				4
2	4				4
4,6	5				5
7	1				1
8,16	12				12
10,16	16	9,13	15	10,16	16
11,20	10				10
14,20	6				6
17,20	8				8
18,23	1				1
21,26	2				2
24,20	6				6
27,29	3				3
30,34	8				8
31,38	12	32,34	12	31,38	12
35, 44	4				4

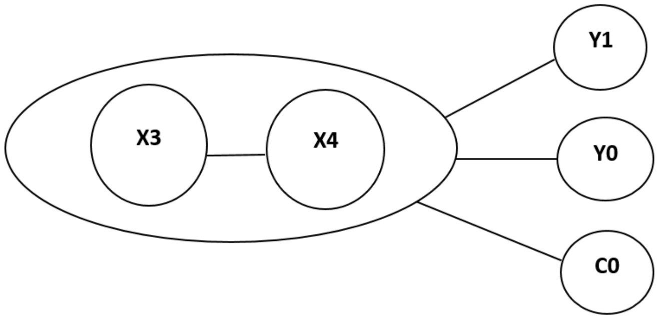
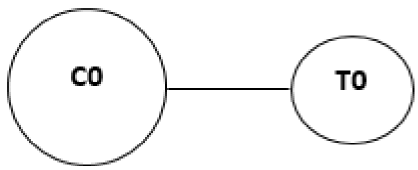
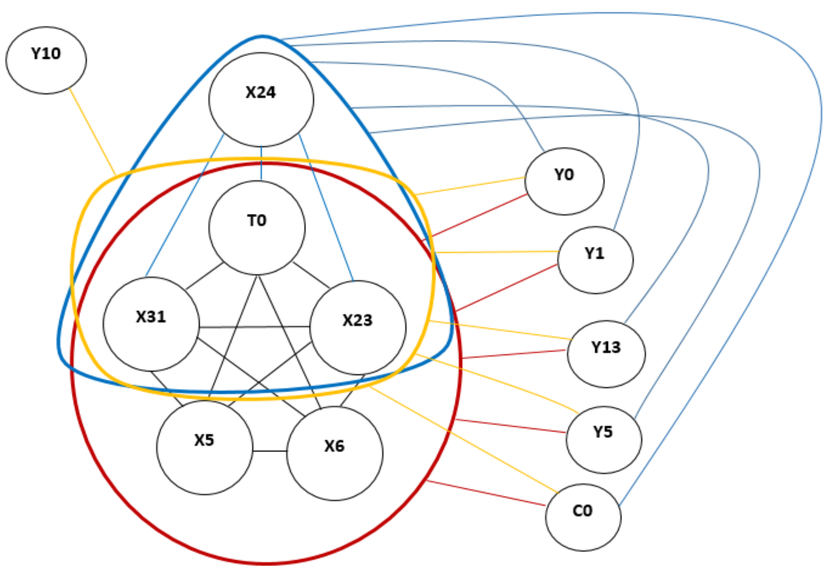
Lines of Code	Interaction	Lines of Code Comparison	Interaction	Final Lines of Code	Final Interaction
36,41	1				1
39,44	2				2
42,44	2				2
45	5				5
46,48	6				8
49	1				1
50	3				3
51	3				3
52,54	8				8
55,97	14				14
56,59	14				14
57,85	14				14
60	1				1
61,63	2				2
64,66	1	90,92	1	64,66	1
67	1				1
68,71	6				6
69,74	8				8
72,92	5				5
75,77	5				5
79	6	78	1	79	6
86	1				1
87,89	2				2
93	1				1
94,71	4	95,74	4	94,71	4
99	4	98	1	99	4
100,102	2	80,82	1	100,102	2
104	4	103	1	104	4
106,108	3	105	1	106,108	3
Total without elimination	270	**Delta**	37	**Total wit elimination**	233

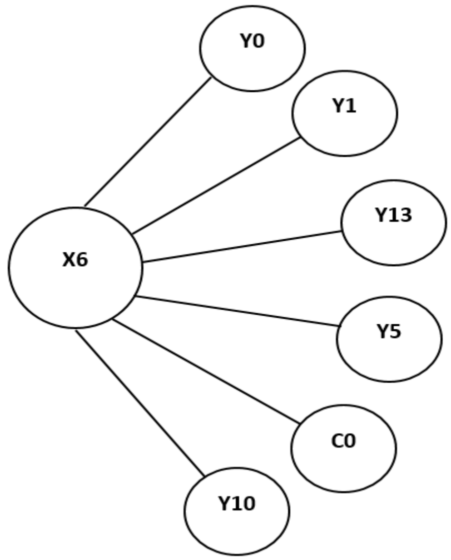
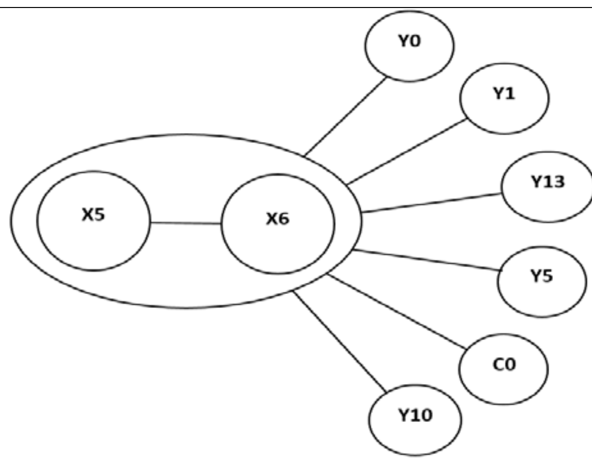
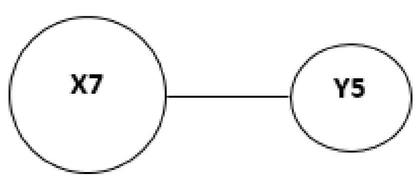
Overlapping diagrams when interactions are reduced

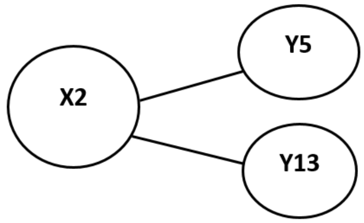
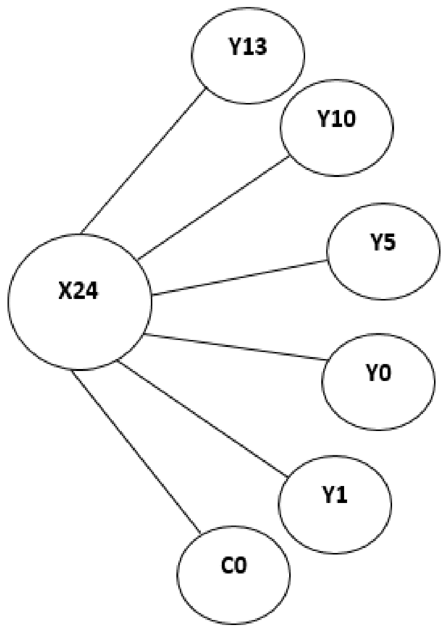
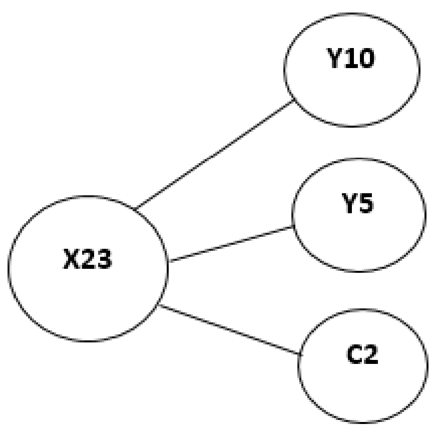
Table B4: Overlapped Diagram based on Lines of Code

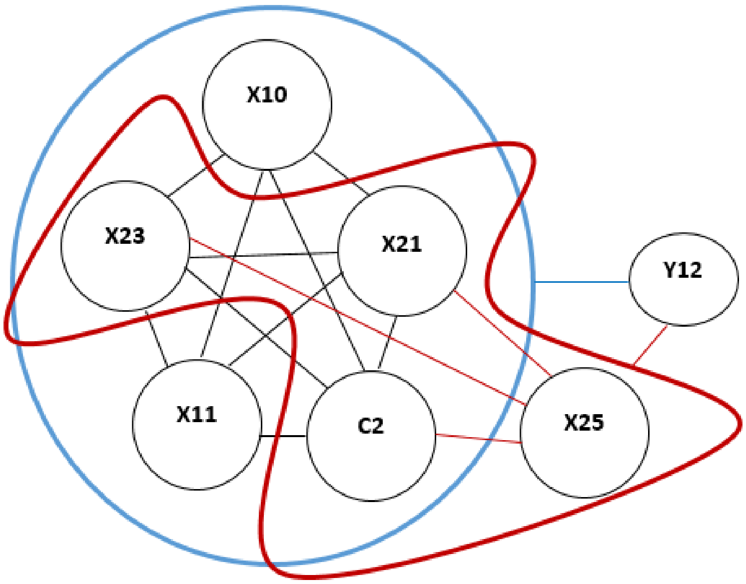
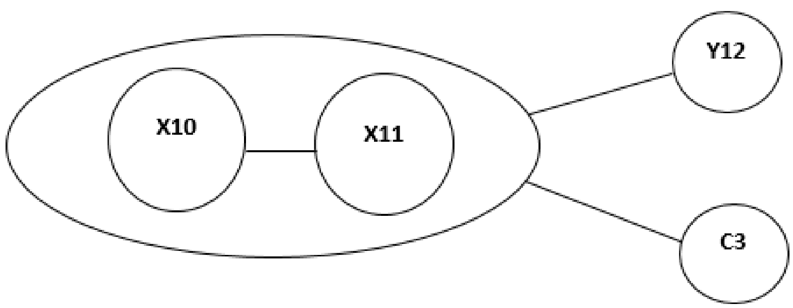
Line of Code	Diagrams	Interaction
1		4

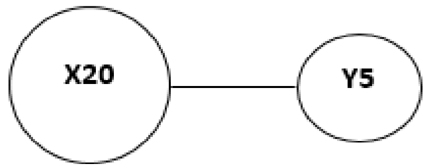
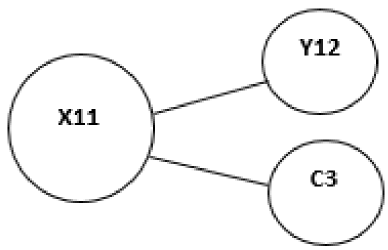
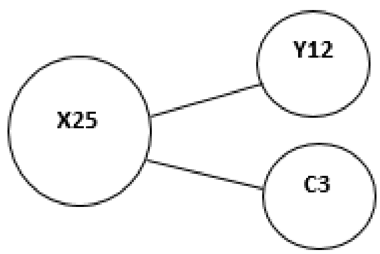
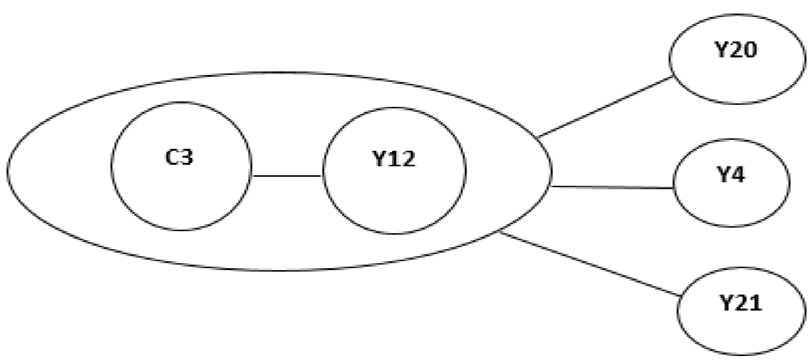
Line of Code	Diagrams	Interaction
		
2		4
4,6		5

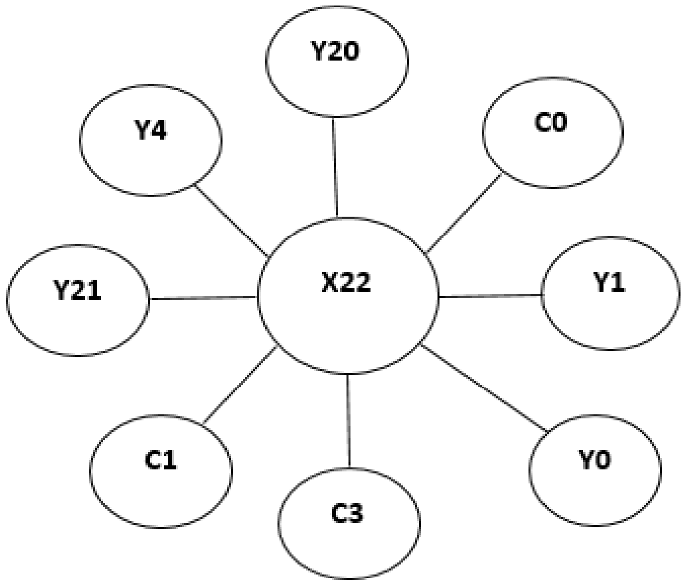
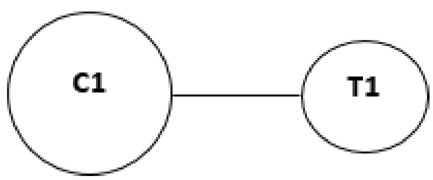
Line of Code	Diagrams	Interaction
		
7		1
8,16; 10,16; 11,20		32

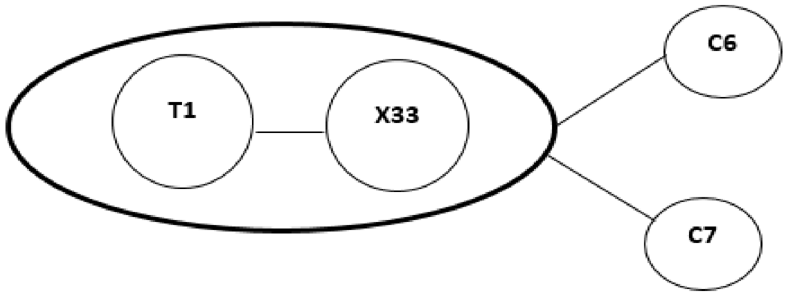
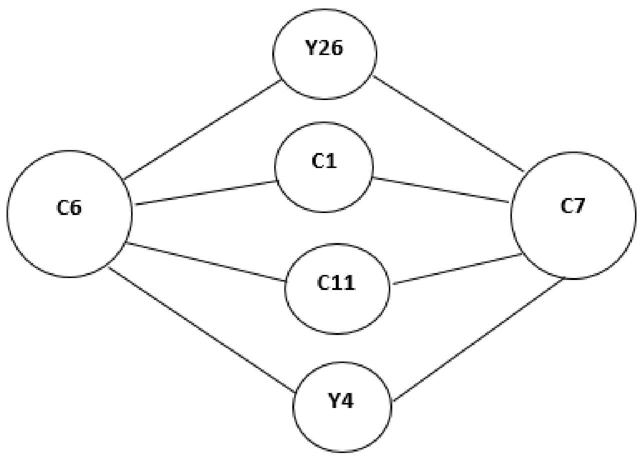
Line of Code	Diagrams	Interaction
14,20		6
17,20		8
18,23		1
21,26		2

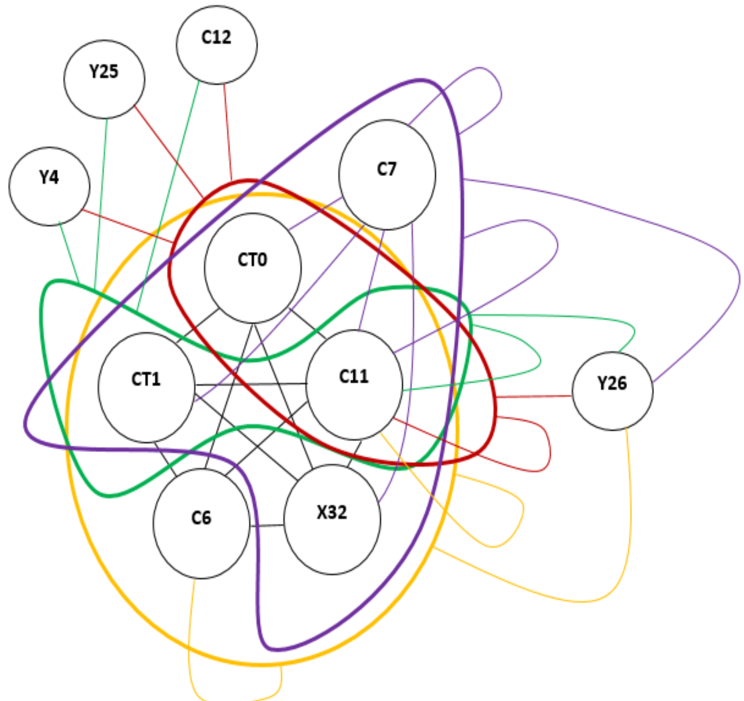
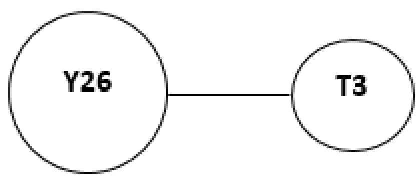
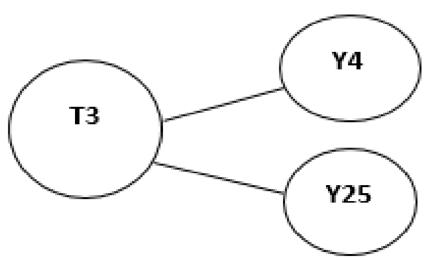
Line of Code	Diagrams	Interaction
		
24,20		6
27,29		3

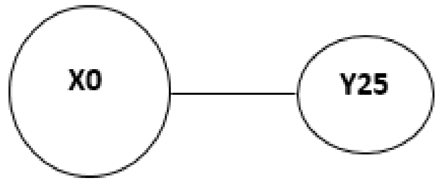
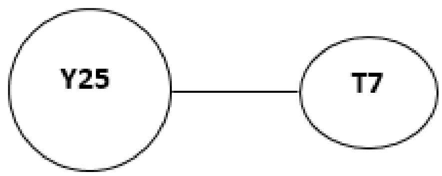
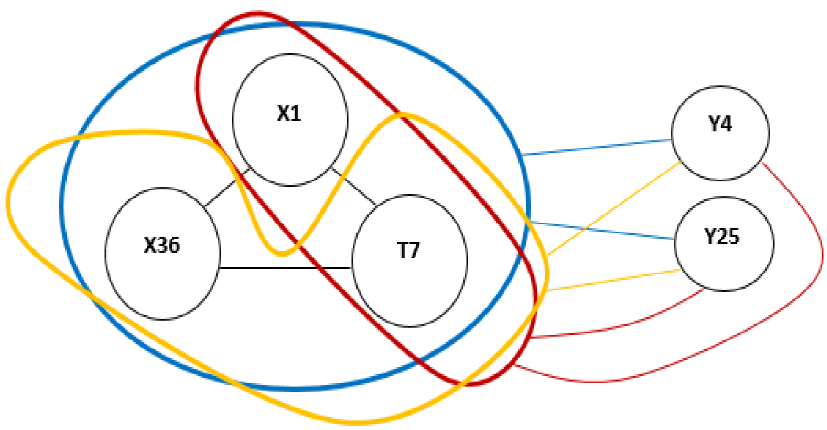
Line of Code	Diagrams	Interaction
30,34; 31,38		17
35,44		4

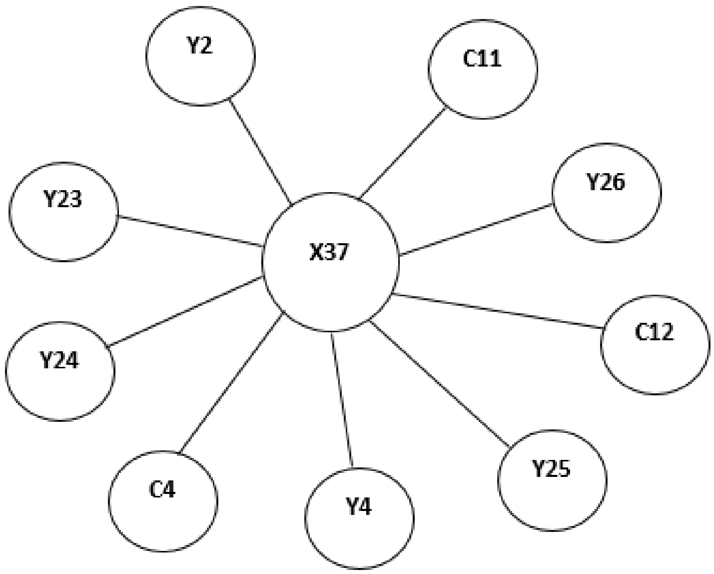
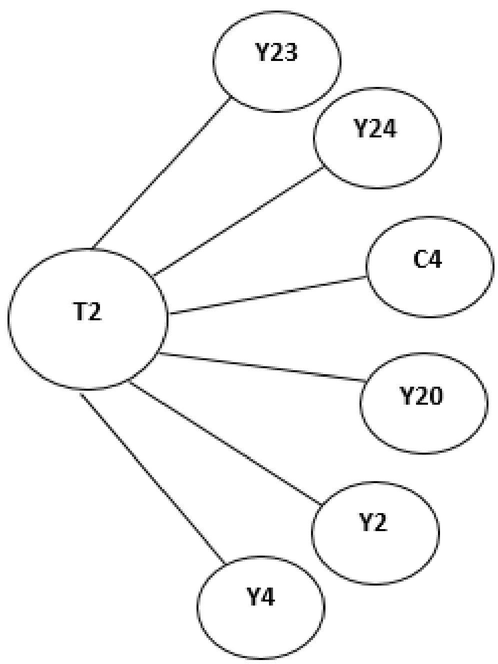
Line of Code	Diagrams	Interaction
36,41		1
39,44		2
42,44		2
45		5

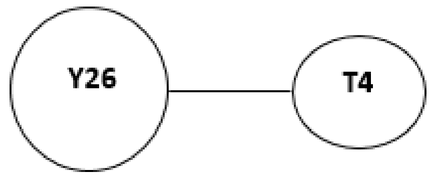
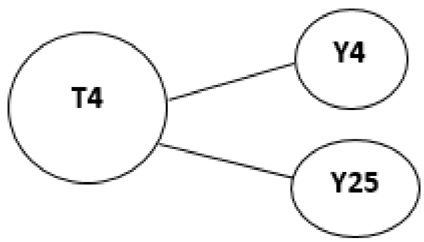
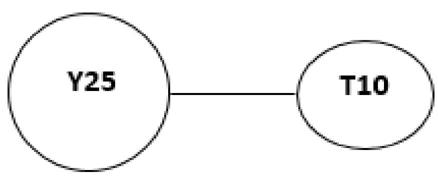
Line of Code	Diagrams	Interaction
46,48		8
49		1
50; 51		4

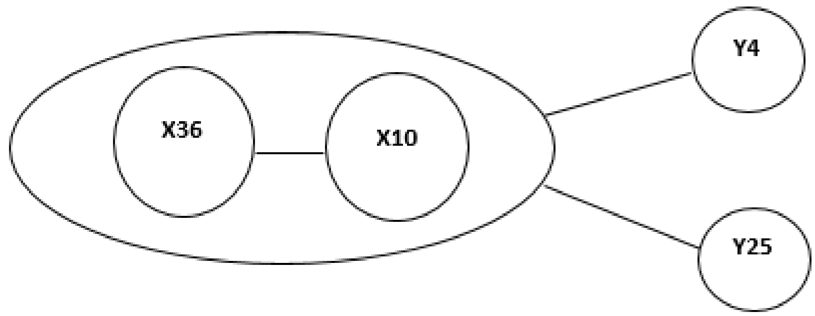
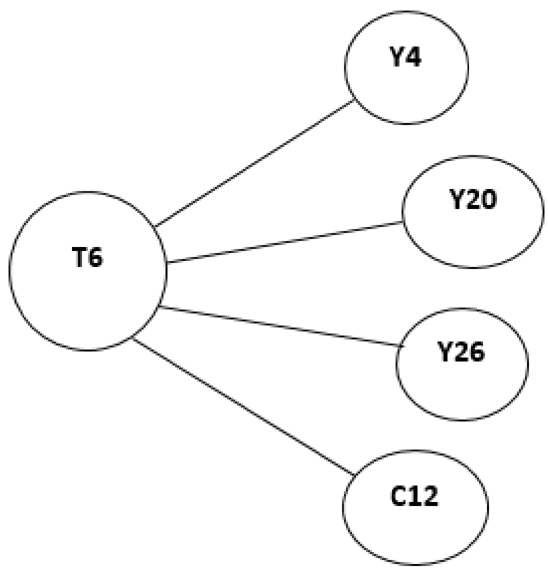
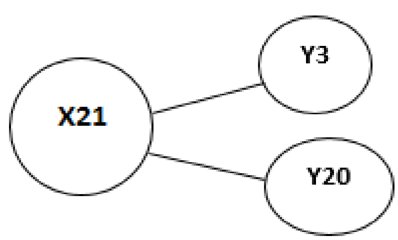
Line of Code	Diagrams	Interaction
		
52,54		8
55,97; 56,59; 57,85		34

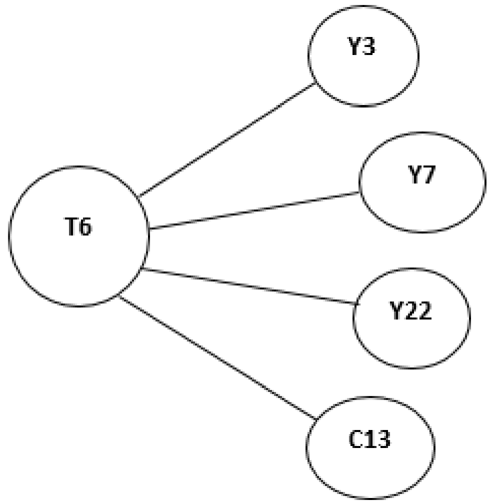
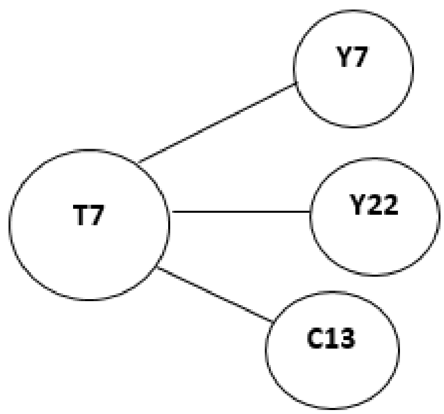
Line of Code	Diagrams	Interaction
		
60		1
61,63		2
64,66		1

Line of Code	Diagrams	Interaction
		
67		1
68,71; 69,74		12

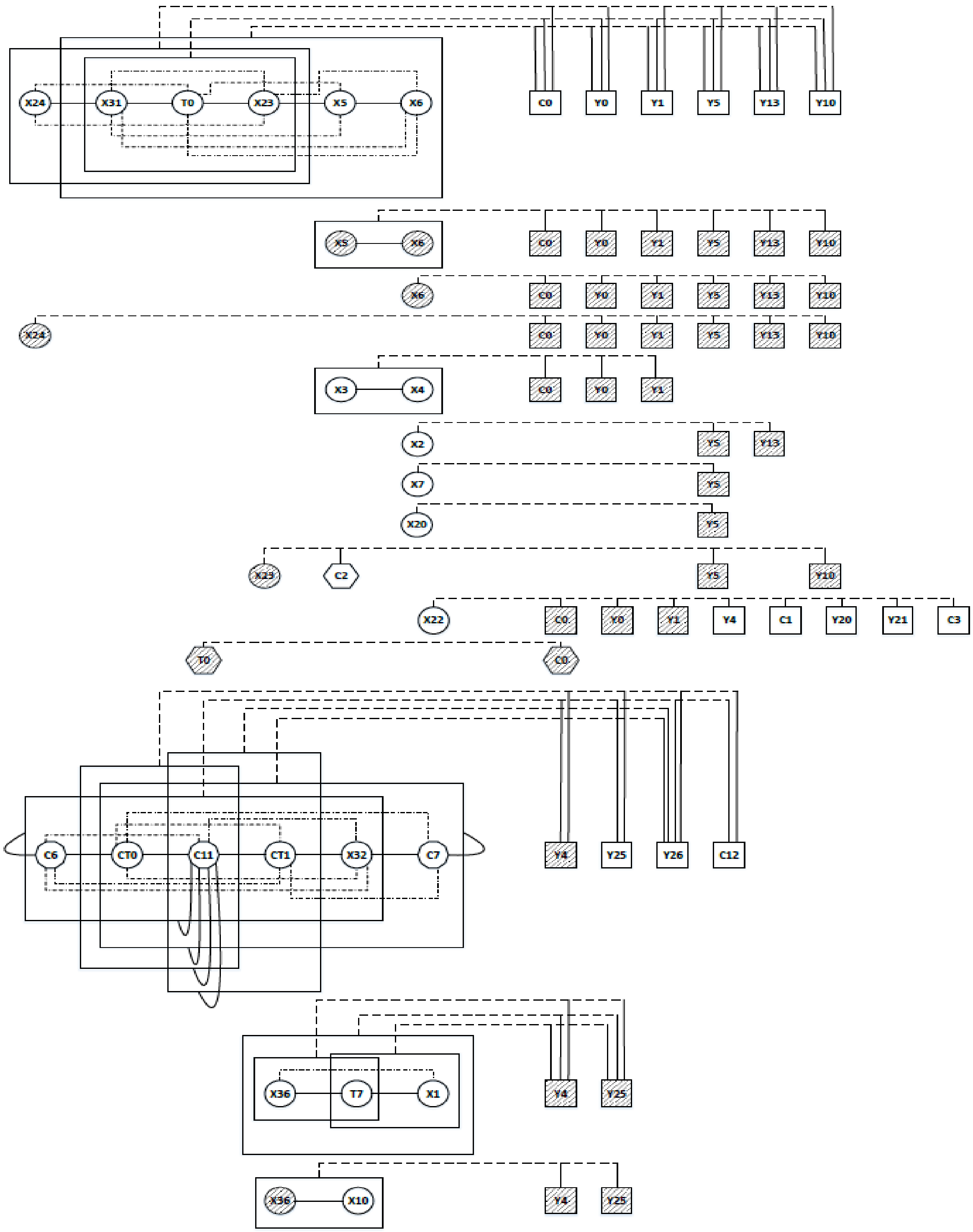
Line of Code	Diagrams	Interaction
72,97; 75,77		9
79		6

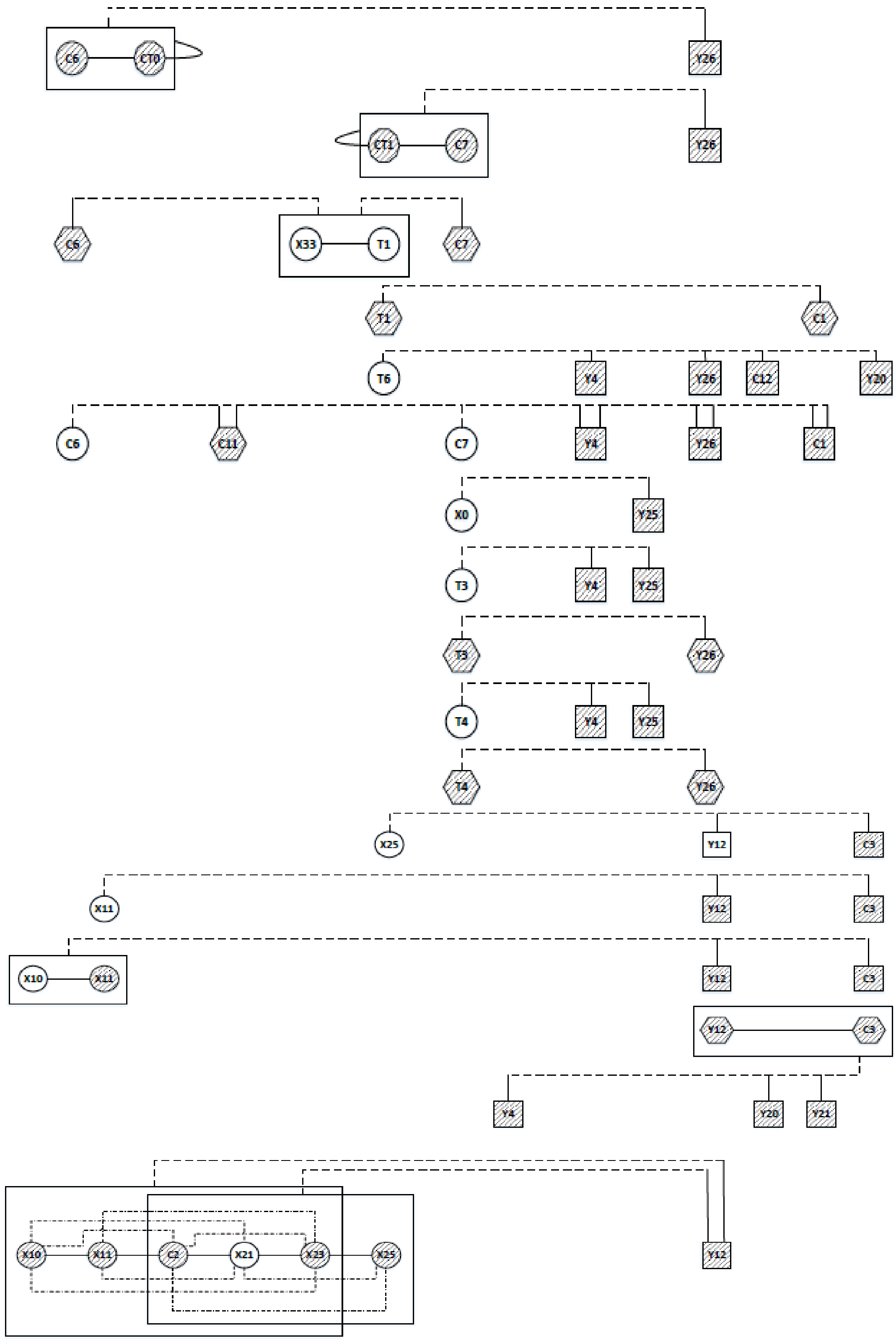
Line of Code	Diagrams	Interaction
86		1
87,89		2
93		1

Line of Code	Diagrams	Interaction
94,71		4
99		4
100,102		2

Line of Code	Diagrams	Interaction
104		4
106,108		3
Total Interactions		**211**
Result: $C = M^2 + I^2 = 58^2 + 211^2 = 47885$		

Validating the methodology through a Big Diagram


X24
X31
T0
X23
X5
X6
C0
Y0
Y1
Y5
Y13
Y10
X3
X4
X2
X7
X20
C2
X22
Y4
C1
Y20
Y21
C3
C6
CT0
C11
CT1
X32
C7
Y25
Y26
C12
X36
T7
X1
X10


C6
CT0
Y26
CT1
C7
Y26
C6
X33
T1
C7
T1
C1
T6
Y4
Y26
C12
Y20
C6
C11
C7
Y4
Y26
C1
X0
Y25
T3
Y4
Y25
T3
Y26
T4
Y4
Y25
T4
Y26
X25
Y12
C3
X11
Y12
C3
X10
X11
Y12
C3
Y12
C3
Y4
Y20
Y21
X10
X11
C2
X21
X23
X25
Y12

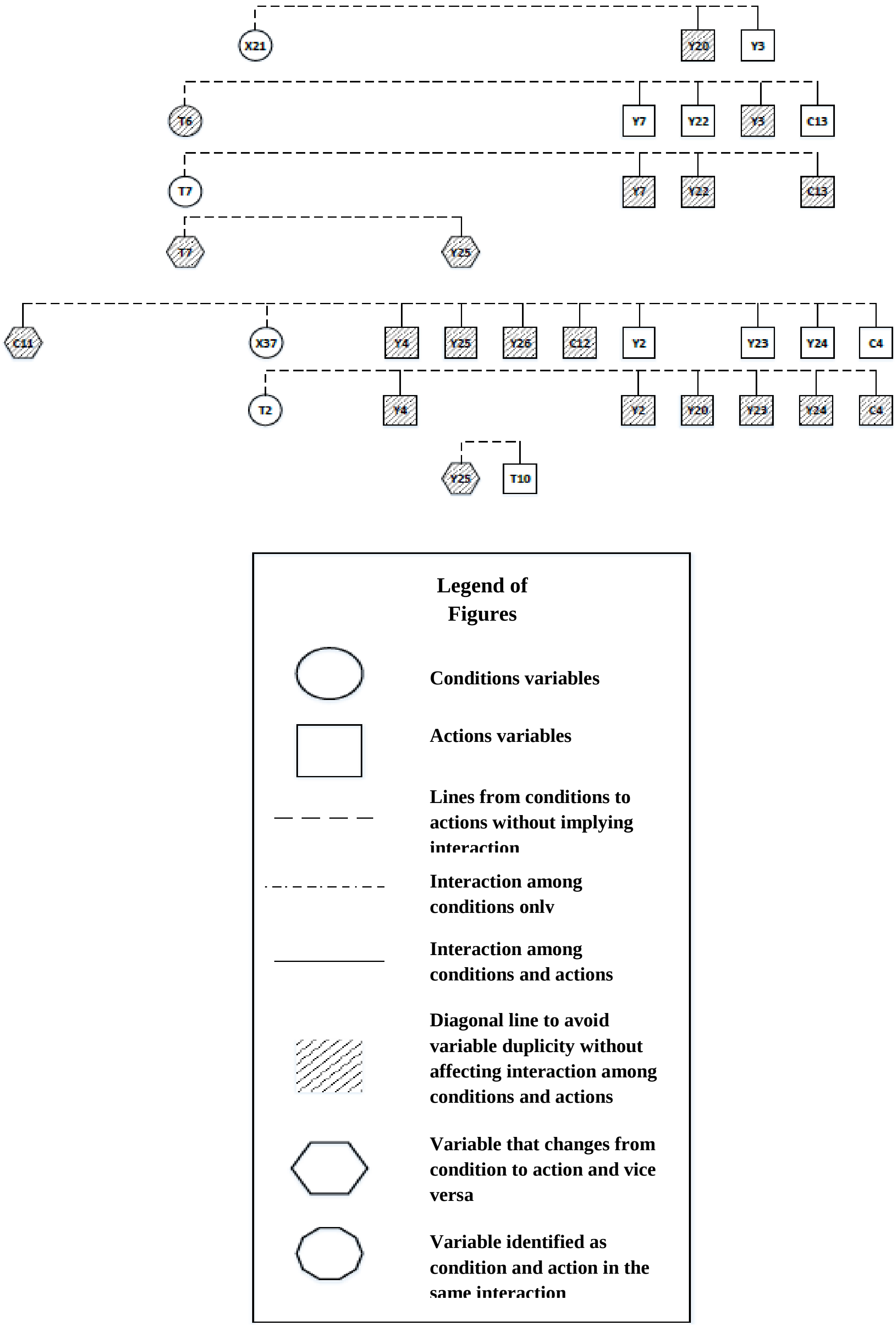


Figure B1: Big Diagram

Appendix C: CPSHI Approval

Comité para la Protección de los Seres Humanos en la Investigación
CPSHI/IRB 00002053
Universidad de Puerto Rico – Recinto Universitario de Mayagüez
Decanato de Asuntos Académicos
Call Box 9000
Mayagüez, PR 00681-9000

25 de abril de 2016

Dra. Lourdes Medina
Ingeniería Industrial
RUM

Estimado Dra. Medina:

El Comité para la Protección de los Seres Humanos en la Investigación (CPSHI) ha considerado su Solicitud de Revisión y demás documentos sometidos para el estudio titulado *A study of complexity in Project based learning for engineering undergraduate education (Protocolo 20160421A-B).*

Su proyecto cualifica para un proceso expedito de aprobación bajo la categoría 7 del 45 CFR 46.110. Luego de evaluarlo, el comité determinó que este estudio no supera el nivel mínimo de riesgo y cumple con todos los requisitos de protección de seres humanos según definidos por la reglamentación federal 45 CFR 46. Por tanto, aprobamos su investigación. La aprobación tiene vigencia de un año a partir de hoy; esto es, desde el 25 de abril de 2016 hasta el 24 de abril de 2017. Le recordamos que la aprobación emitida por nuestro comité no lo exime de cumplir con cualquier otro requisito institucional o gubernamental relacionado al tema o fuente de financiamiento de su proyecto.

La reglamentación federal exige que nuestro comité supervise toda investigación mientras continúe activa. Se consideran activos aquellos proyectos que aún estén reclutando participantes o haya terminado el reclutamiento pero aún se estén recopilando o analizando datos. Si vislumbra que su proyecto seguirá activo al momento de vencerse la fecha de aprobación, le pedimos que someta una solicitud de extensión a más tardar un mes antes del vencimiento de su vigencia.

Le adjuntamos la hoja de consentimiento con el sello de aprobación del Comité. Le agradeceremos utilice estos documentos para los trámites correspondientes de su investigación. Le recordamos que debe entregarle una copia de la hoja de consentimiento informado a todos/as los/as participantes que acepten ser parte de su estudio.

Cualquier cambio al protocolo o a la metodología deberá ser revisado y aprobado por el CPSHI antes de su implantación, excepto en casos en que el cambio sea necesario para eliminar algún riesgo inmediato para los/as participantes. El CPSHI deberá ser notificado de dichos cambios tan pronto le sea posible al/a la investigador/a. El CPSHI deberá ser informado de inmediato de cualquier efecto adverso o problema inesperado que surgiera con relación al riesgo de los seres humanos, de cualquier queja sobre esta investigación y de cualquier violación a la confidencialidad de los participantes.

Cordialmente,

Dr. Rafael A. Boglio Martínez
Presidente
CPSHI/IRB
UPR – RUM

Teléfono: (787) 832 - 4040 x 6277, 3807, 3808 – **Fax:** (787) 831-2085 – **Página Web:** www.uprm.edu/cpshi
Email: cpshi@uprm.edu

Comité para la Protección de los Seres Humanos en la Investigación
CPSHI/IRB 00002053
Universidad de Puerto Rico – Recinto Universitario de Mayagüez
Decanato de Asuntos Académicos
Call Box 9000
Mayagüez, PR 00681-9000

February 1, 2017

Dr. Lourdes Medina
Industrial Engineering
RUM

Dear Dr. Medina:

As Director of the Institutional Review Board of the University of Puerto Rico - Mayagüez Campus, I have considered your request for modification for the project titled ***A study of complexity in project-based learning for engineering undergraduate education*** (Protocol num. 20160908). After evaluating the modifications, I have determined that they do not alter the criteria that led to the original approval. The modifications do not impose any additional risk to participants nor do they alter the guarantees of anonymity and confidentiality. For these reasons, your original approval stands.

We remind you that any modifications or amendments to the approved protocol or its methodology must be reviewed and approved by the IRB before they are implemented, except in cases where the change is necessary to reduce or eliminate a potential risk for participants. The IRB must be informed immediately if an adverse event or unexpected problem arises related to the risk to human subjects. The IRB must likewise be notified immediately if any breach of confidentiality occurs.

Sincerely,

Dr. Rafael A. Boglio Martínez
President
CPSHI/IRB
UPR - RUM

Teléfono: (787) 832 - 4040 x 6277, 3807, 3808 – **Fax:** (787) 831-2085 – **Página Web:** www.uprm.edu/cpshi
Email: cpshi@uprm.edu

Appendix D: R code for Random Forest Analysis

```
setwd("C:/Users/sotoz/OneDrive - Hewlett Packard Enterprise/Data")
library(readr)
rawdata <-read_csv("C:/Users/sotoz/OneDrive - Hewlett Packard Enterprise/Data/VCIrawdata.csv")

library(randomForest)

#Cross Validation
data=rawdata

nFolds=10

MAD=matrix(nrow=nFolds,ncol=1)
MAPE=matrix(nrow=nFolds,ncol=1)#cv MAPE matrix
MSE=matrix(nrow=nFolds,ncol=1)#cv error matrix
Rsq=matrix(nrow=nFolds,ncol=1) # aVCIuracy matrix
permRows=sample(x=1:nrow(data),size=nrow(data),replace=FALSE)

# Create testing and training folds
obsFold=floor(nrow(data)/nFolds)
pending=nrow(data)-floor(nrow(data)/nFolds)*nFolds
j=0

for (i in 1:nFolds){
  if (i>=(nFolds-pending+1) & pending>0) {
    assign(paste("F",i,sep=""),data[permRows[(j+1):(j+obsFold)],]) ; j= j + obsFold + 1 } else
    { assign(paste("F",i,sep=""),data[permRows[(j+1):(j+obsFold)],]); j= j + obsFold }

  testing=get(paste("F",i,sep=""))
  trainingRows=setdiff(1:nrow(data),as.numeric(row.names(testing)))
  training=data[trainingRows,]

  #RandomForest Regression
  myRF=randomForest(VCI~.,data=training,importance=TRUE,do.trace=100,proximity=TRUE)
  pred_cv=predict(myRF,newdata=testing)
  actual=testing$VCI
  MAD[i]=sum(abs(actual-pred_cv))/length(pred_cv)
  MSE[i]=sum((actual-pred_cv)*(actual-pred_cv))/length(pred_cv)
  MAPE[i]=sum(abs(actual-pred_cv)/actual)*(1/length(pred_cv)) # the mean absolute percentage error
  Rsq[i]= 1 - sum((actual-pred_cv)^2)/sum((actual-mean(actual))^2)
}

plot(myRF)

myRF$importance
varImpPlot(myRF)

impo<-myRF$importance
write(impo,"C:/Users/sotoz/OneDrive - Hewlett Packard Enterprise/Data/ResulTs/VCIimportscoreP.cvs")
```

```
#myRF$mtry
#myRF$type
imposd<-myRF$importanceSD
write.csv(imposd,"C:/Users/sotoz/OneDrive - Hewlett Packard
Enterprise/Data/Results/VCIimportscoreP.csv")

#myRF$ntree
#myRF$oob.times
#myRF$forest #averiguar mas sobre este valor
#myRF$forest$nodepred
#myRF$proximity

#Performance metrics
MSE
mean(MSE) # error
Rsq
mean(Rsq) # aVCIurracy
MAD
mean(MAD)
MAPE
mean(MAPE)

ok<-cbind(MSE,Rsq,MAD,MAPE)
ok
```

Appendix E: R code for Decision Tree Analysis

```
setwd("C:/Users/sotoz/OneDrive - Hewlett Packard Enterprise/Data")
library(readr)
rawdata <-read_csv("C:/Users/sotoz/OneDrive - Hewlett Packard Enterprise/Data/locrawdata.csv")

library(randomForest)

#Cross Validation
data=rawdata

nFolds=10

MAD=matrix(nrow=nFolds,ncol=1)
MAPE=matrix(nrow=nFolds,ncol=1)#cv MAPE matrix
MSE=matrix(nrow=nFolds,ncol=1)#cv error matrix
Rsq=matrix(nrow=nFolds,ncol=1) # aVCIuracy matrix
permRows=sample(x=1:nrow(data),size=nrow(data),replace=FALSE)

# Create testing and training folds
obsFold=floor(nrow(data)/nFolds)
pending=nrow(data)-floor(nrow(data)/nFolds)*nFolds
j=0

for (i in 1:nFolds){
  if (i>=(nFolds-pending+1) & pending>0) {
    assign(paste("F",i,sep=""),data[permRows[(j+1):(j+obsFold)],]) ; j= j + obsFold + 1 } else
    { assign(paste("F",i,sep=""),data[permRows[(j+1):(j+obsFold)],]); j= j + obsFold }

  testing=get(paste("F",i,sep=""))
  trainingRows=setdiff(1:nrow(data),as.numeric(row.names(testing)))
  training=data[trainingRows,]

  #RandomForest Regression
  myRF=randomForest(LOC~.,data=training,importance=TRUE,do.trace=100,proximity=TRUE)
  pred_cv=predict(myRF,newdata=testing)
  actual=testing$LOC
  MAD[i]=sum(abs(actual-pred_cv))/length(pred_cv)
  MSE[i]=sum((actual-pred_cv)*(actual-pred_cv))/length(pred_cv)
  MAPE[i]=sum(abs(actual-pred_cv)/actual)*(1/length(pred_cv)) # the mean absolute percentage error
  Rsq[i]= 1 - sum((actual-pred_cv)^2)/sum((actual-mean(actual))^2)
}

plot(myRF)

myRF$importance
varImpPlot(myRF)


myRF$mtry
```

```
MSE
MMSE<-mean(MSE) # error

Rsq
MRsq<-mean(Rsq) # aVCIurracy

MAD
MMAD<-mean(MAD)

MAPE
MMAPE<-mean(MAPE)

ok<-cbind(MMSE,MRsq,MMAD,MMAPE)
ok
```

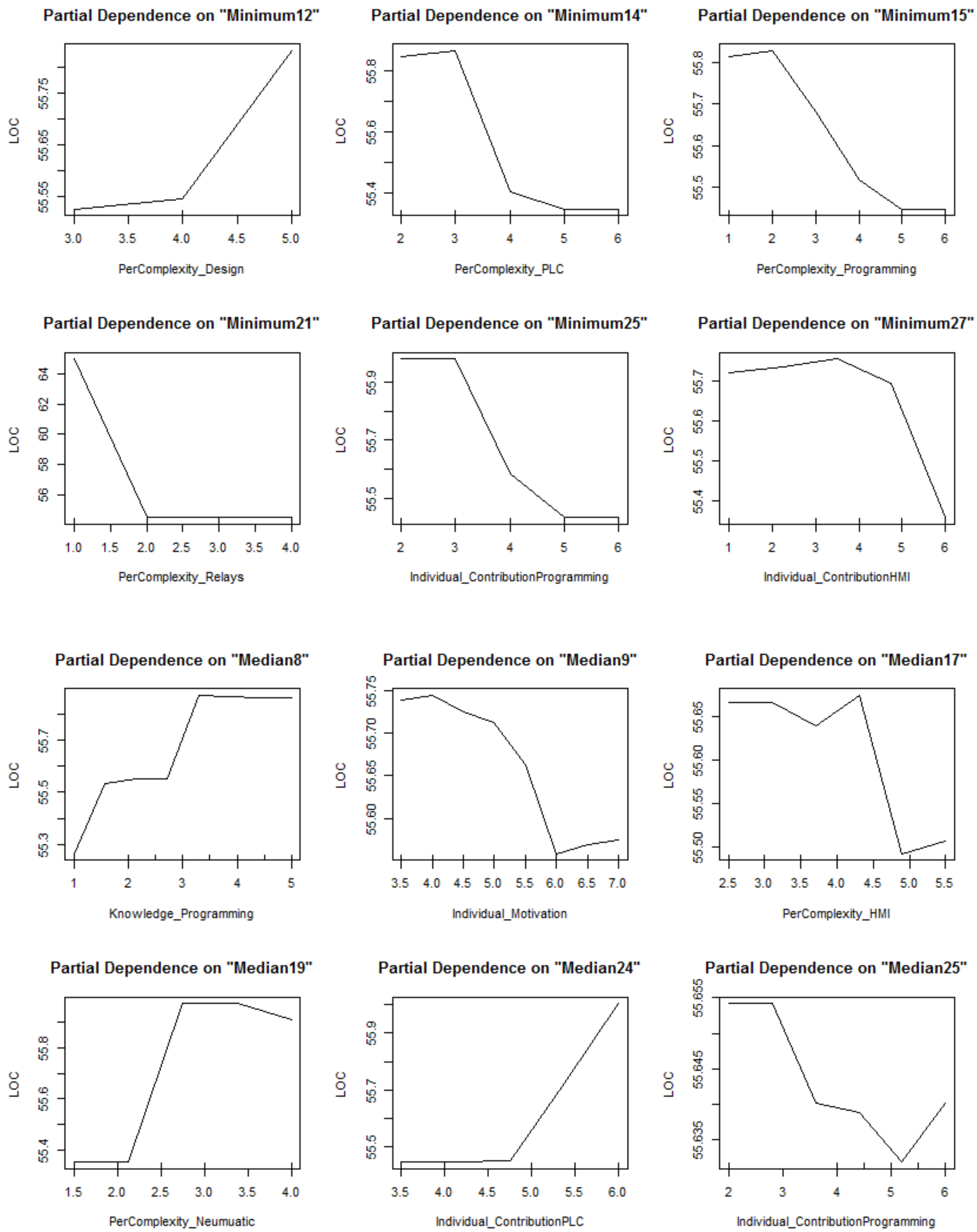
Appendix F: Friedman Test

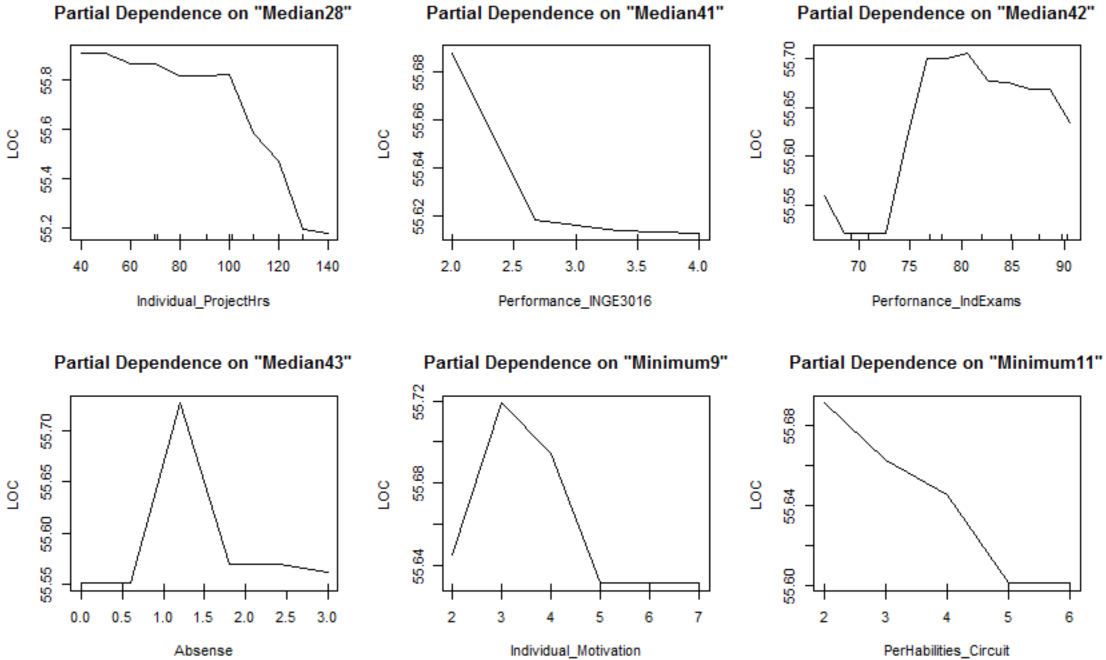
```
setwd("C:/Users/sotoz/OneDrive - Hewlett Packard Enterprise/Data")

library(readr)

rawdata <-read_csv("C:/Users/sotoz/OneDrive - Hewlett Packard
Enterprise/Data/Results/NewResults/Friedman2.csv")

data=as.matrix(rawdata)

data

Ft<-friedman.test(data)

Ft

library(PMCMR)

posthoc.friedman.conover.test(data,p.adjust.method="bonferron")
```

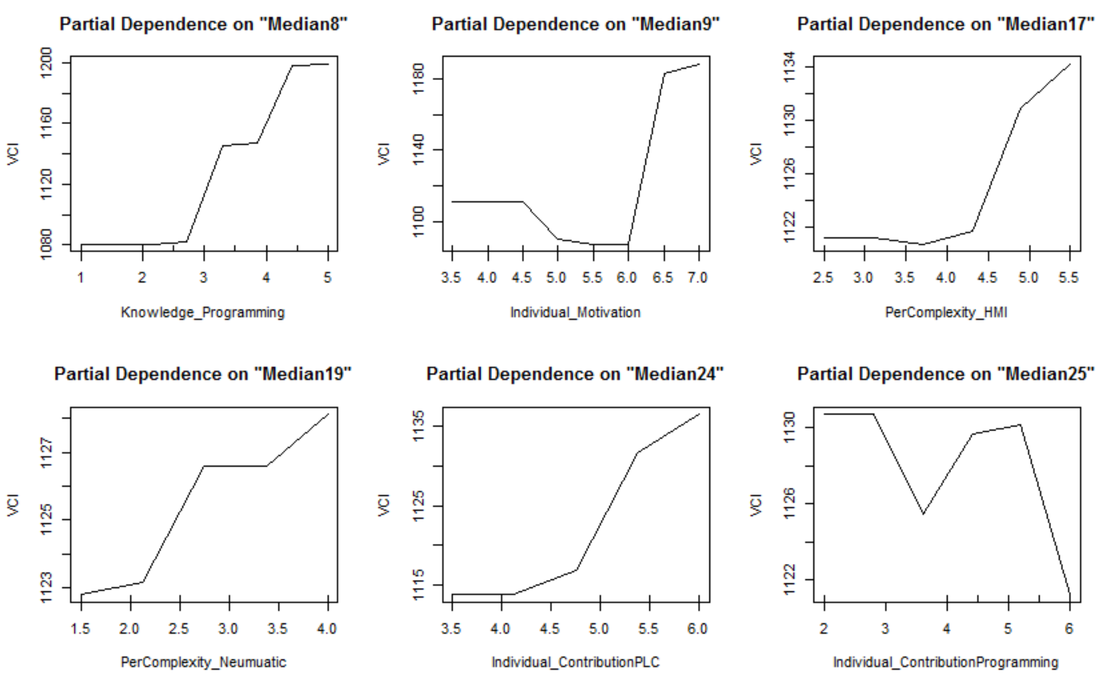
Appendix G: Student's t-Test for Variable Importance Selection

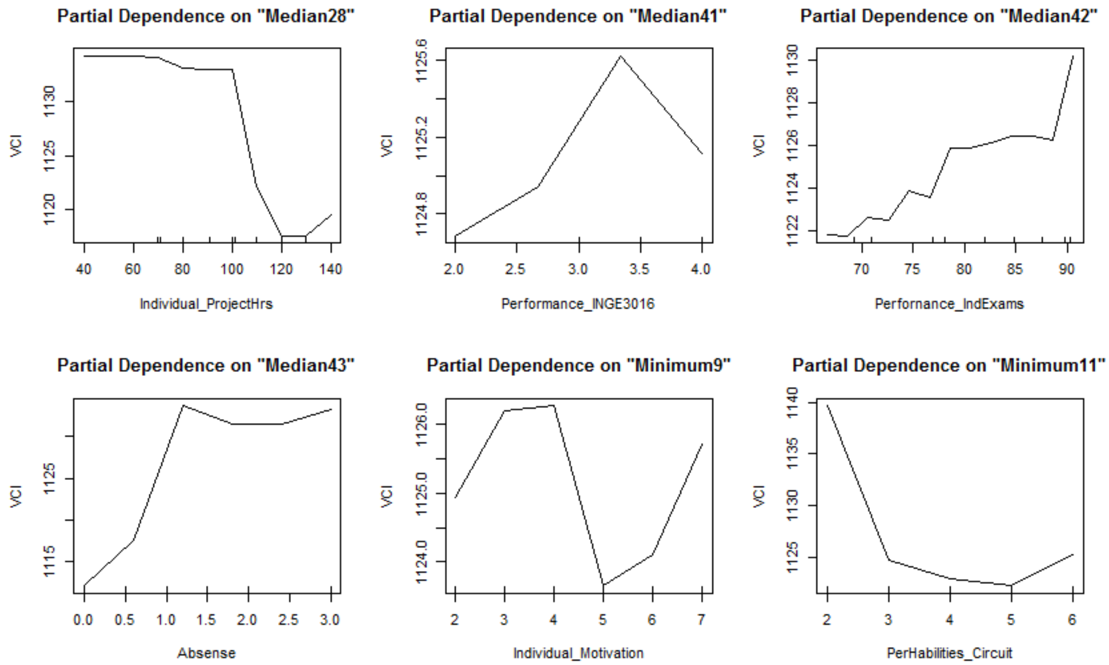
```
setwd("C:/Users/sotoz/OneDrive - Hewlett Packard Enterprise/Data")
library(readr)
rawdata <-read_csv("C:/Users/sotoz/OneDrive - Hewlett Packard Enterprise/Data/grpallSCIS.csv")

library(randomForest)
data=rawdata
names(data)[ncol(data)]="Y"
qArtificial=0.9
nPerm=30
nVar=ncol(data)-1
X=data.frame(matrix(nrow=nrow(data),ncol=nVar*2))
X[,1:nVar]=data[,1:(ncol(data)-1)]
impor=matrix(nrow=nVar*2,ncol=nPerm)
q=matrix(nrow=nPerm,ncol=1)
for (i in 1:nPerm)
{
  for (j in 1:nVar)# Artificial variables
  {
    X[,nVar+j]=sample(X[,j],length(X[,j]),replace=FALSE)
  }
  data2=cbind(X,data$Y) # New data frame with original Xs, artificial Xs, and Y at the end
  names(data2)[ncol(data2)]="Y"
  data2$Y=as.factor(data2$Y) # Depende del tipo de variable
  rF<-randomForest(Y~.,data=data2,ntree=500,importance=TRUE)
  impor[,i]<-rF$importance[,5] # Gini
  q[i]<-quantile(impor[(nVar+1):(2*nVar),i],probs=qArtificial)
}

pval<-matrix(nrow=nVar,ncol=1)
for (i in 1:nVar)
{
  test=t.test(x=cbind(impor[i,]),y=cbind(q),alternative="greater",paired=TRUE,conf.level=0.95) # no
parametrica
  pval[i,1]<-test$p.value
}
pval<-data.frame(pval)
dfPVAL=cbind(1:nVar,pval)
impVars=subset(dfPVAL,pval<0.05/nVar,1)
impVars
```

Appendix H: Partial Dependence Plot

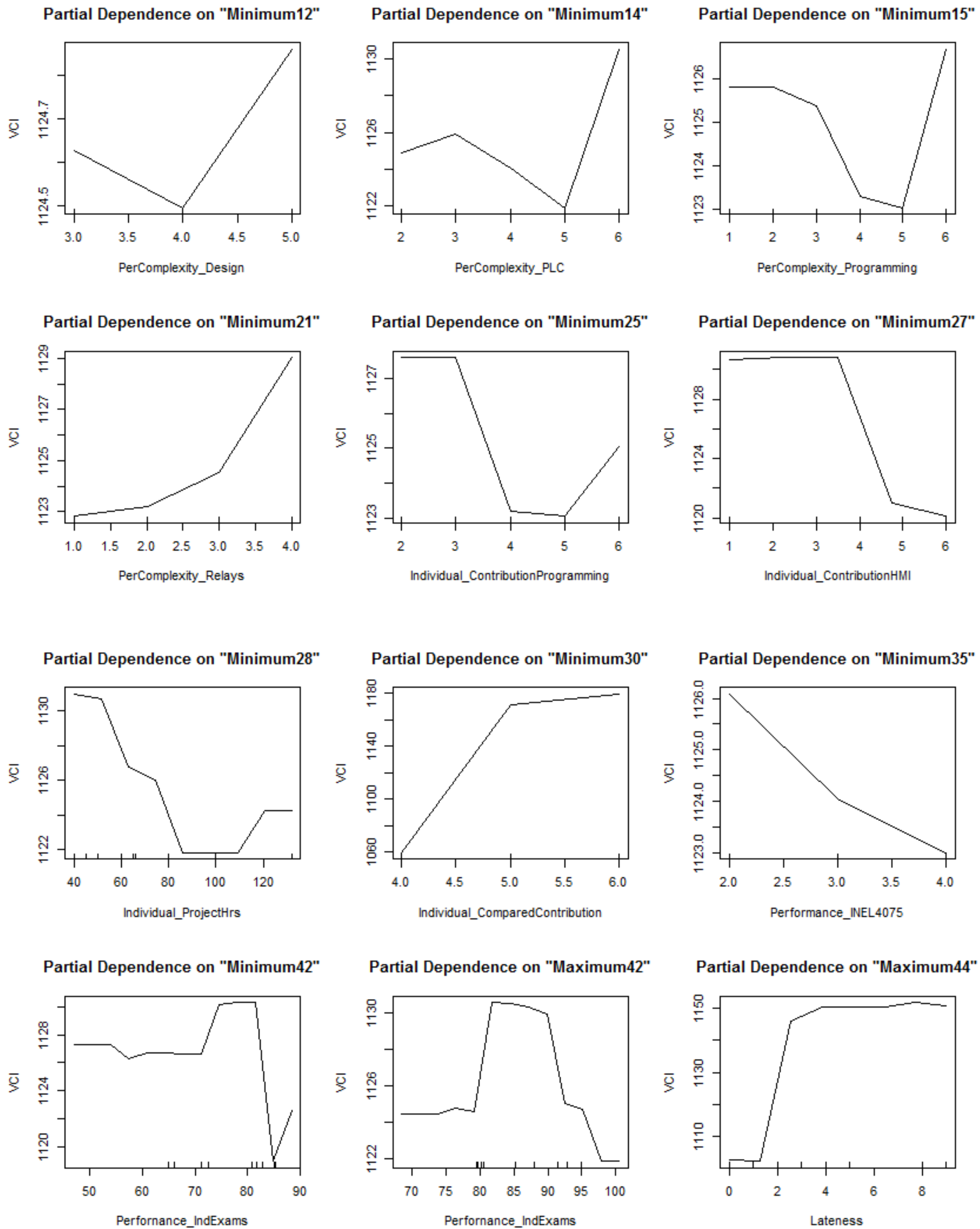


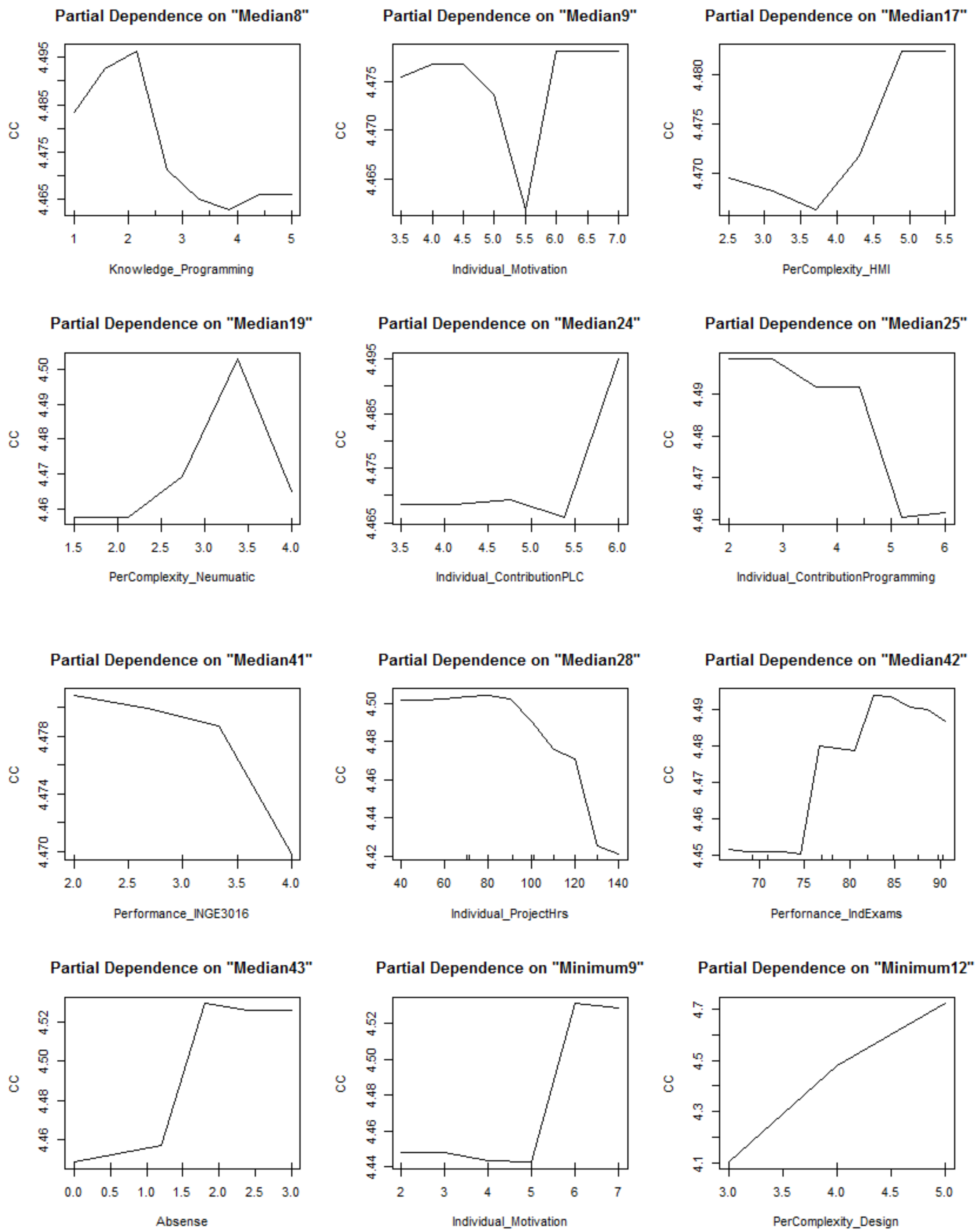

Partial Dependence on "Median28"
LOC
Individual_ProjectHrs
Partial Dependence on "Median41"
LOC
Performance_INGE3016
Partial Dependence on "Median42"
LOC
Perfornance_IndExams
Partial Dependence on "Median43"
LOC
Absense
Partial Dependence on "Minimum9"
LOC
Individual_Motivation
Partial Dependence on "Minimum11"
LOC
PerHabilities_Circuit

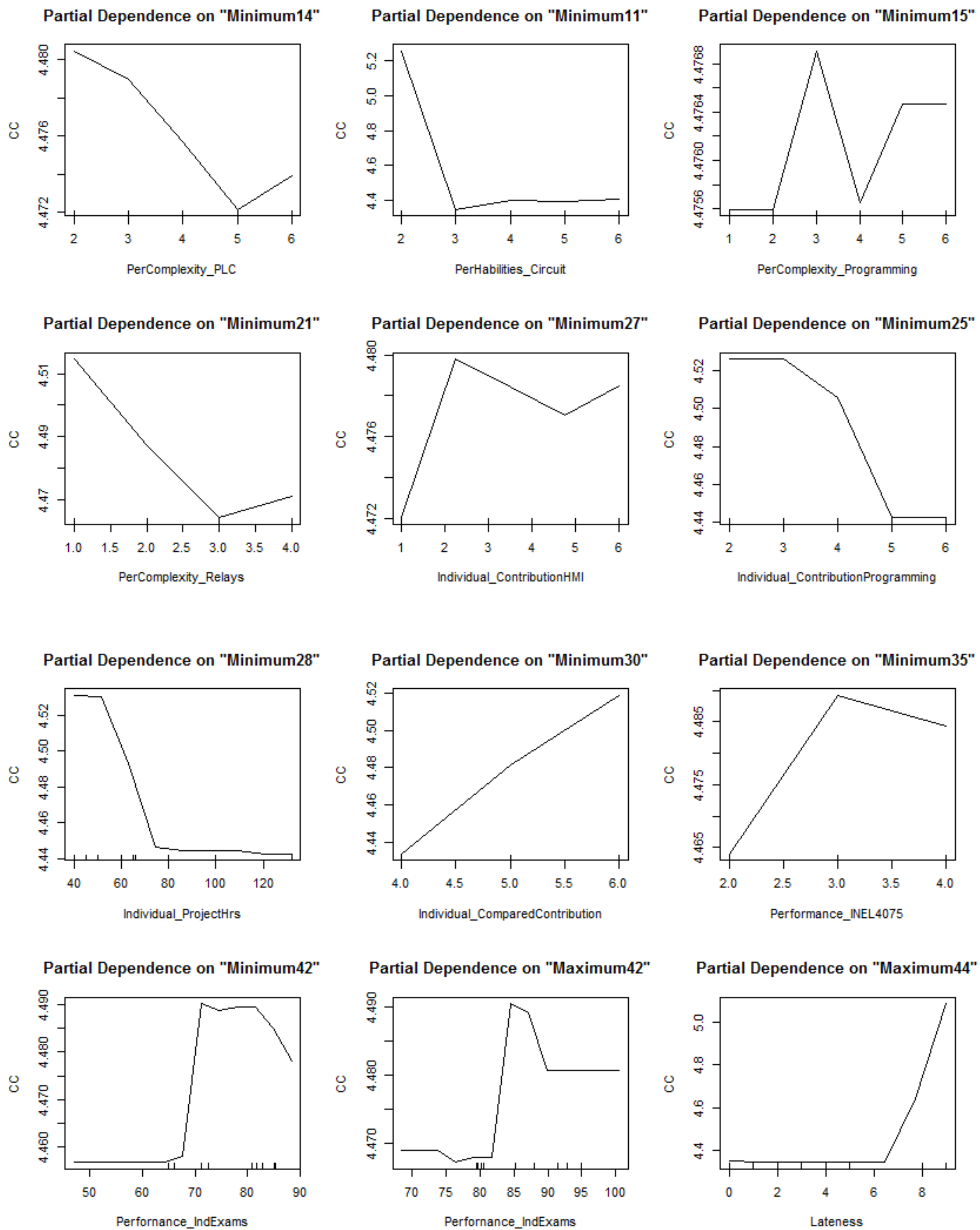

Partial Dependence on "Median8"
VCI
Knowledge_Programming
Partial Dependence on "Median9"
VCI
Individual_Motivation
Partial Dependence on "Median17"
VCI
PerComplexity_HMI
Partial Dependence on "Median19"
VCI
PerComplexity_Neumuatic
Partial Dependence on "Median24"
VCI
Individual_ContributionPLC
Partial Dependence on "Median25"
VCI
Individual_ContributionProgramming

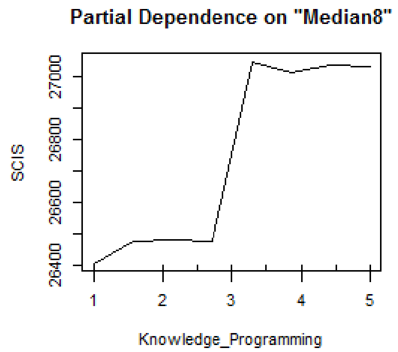


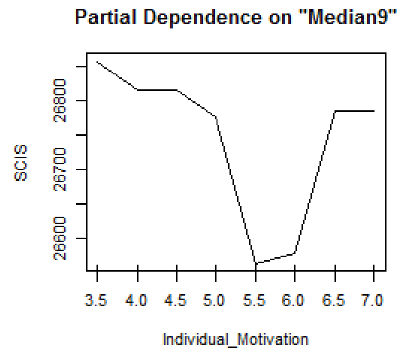
Partial Dependence on "Median28"
VCI
Individual_ProjectHrs
Partial Dependence on "Median41"
VCI
Performance_INGE3016
Partial Dependence on "Median42"
VCI
Perfornance_IndExams
Partial Dependence on "Median43"
VCI
Absense
Partial Dependence on "Minimum9"
VCI
Individual_Motivation
Partial Dependence on "Minimum11"
VCI
PerHabilities_Circuit

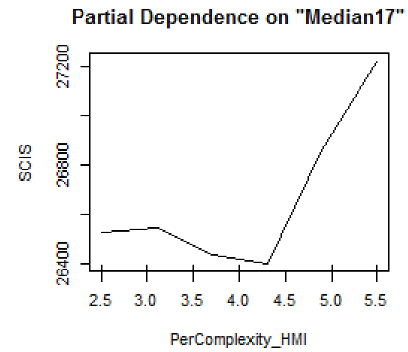

Partial Dependence on "Minimum12"
VCI
PerComplexity_Design
Partial Dependence on "Minimum14"
VCI
PerComplexity_PLC
Partial Dependence on "Minimum15"
VCI
PerComplexity_Programming
Partial Dependence on "Minimum21"
VCI
PerComplexity_Relays
Partial Dependence on "Minimum25"
VCI
Individual_ContributionProgramming
Partial Dependence on "Minimum27"
VCI
Individual_ContributionHMI
Partial Dependence on "Minimum28"
VCI
Individual_ProjectHrs
Partial Dependence on "Minimum30"
VCI
Individual_ComparedContribution
Partial Dependence on "Minimum35"
VCI
Performance_INEL4075
Partial Dependence on "Minimum42"
VCI
Perfornance_IndExams
Partial Dependence on "Maximum42"
VCI
Perfornance_IndExams
Partial Dependence on "Maximum44"
VCI
Lateness

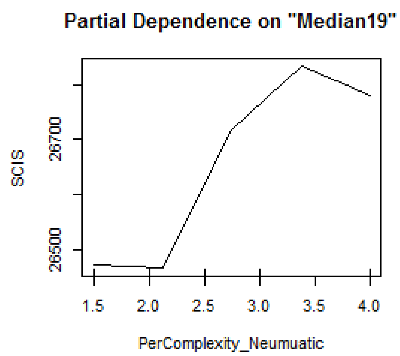

Partial Dependence on "Median8"
CC
Knowledge_Programming
Partial Dependence on "Median9"
CC
Individual_Motivation
Partial Dependence on "Median17"
CC
PerComplexity_HMI
Partial Dependence on "Median19"
CC
PerComplexity_Neumuatic
Partial Dependence on "Median24"
CC
Individual_ContributionPLC
Partial Dependence on "Median25"
CC
Individual_ContributionProgramming
Partial Dependence on "Median41"
CC
Performance_INGE3016
Partial Dependence on "Median28"
CC
Individual_ProjectHrs
Partial Dependence on "Median42"
CC
Perfornance_IndExams
Partial Dependence on "Median43"
CC
Absense
Partial Dependence on "Minimum9"
CC
Individual_Motivation
Partial Dependence on "Minimum12"
CC
PerComplexity_Design

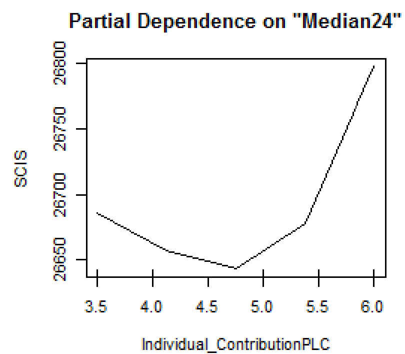

Partial Dependence on "Minimum14"
CC
PerComplexity_PLC
Partial Dependence on "Minimum11"
CC
PerHabilities_Circuit
Partial Dependence on "Minimum15"
CC
PerComplexity_Programming
Partial Dependence on "Minimum21"
CC
PerComplexity_Relays
Partial Dependence on "Minimum27"
CC
Individual_ContributionHMI
Partial Dependence on "Minimum25"
CC
Individual_ContributionProgramming
Partial Dependence on "Minimum28"
CC
Individual_ProjectHrs
Partial Dependence on "Minimum30"
CC
Individual_ComparedContribution
Partial Dependence on "Minimum35"
CC
Performance_INEL4075
Partial Dependence on "Minimum42"
CC
Performance_IndExams
Partial Dependence on "Maximum42"
CC
Performance_IndExams
Partial Dependence on "Maximum44"
CC
Lateness

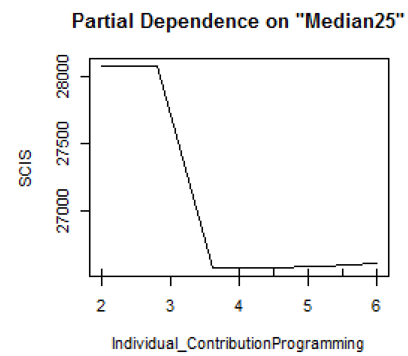

Partial Dependence on "Median8"
SCIS
Knowledge_Programming

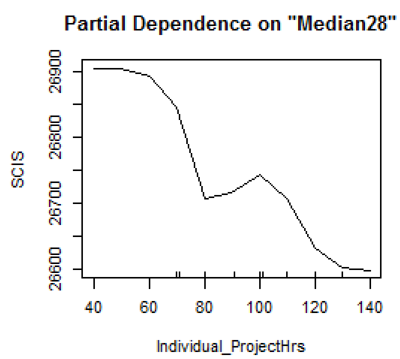

Partial Dependence on "Median9"
SCIS
Individual_Motivation

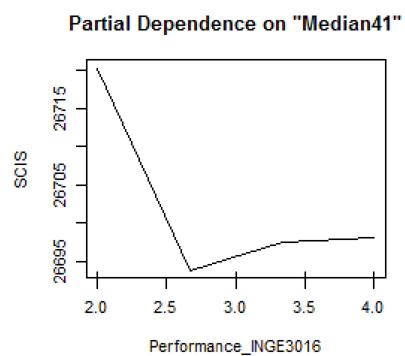

Partial Dependence on "Median17"
SCIS
PerComplexity_HMI

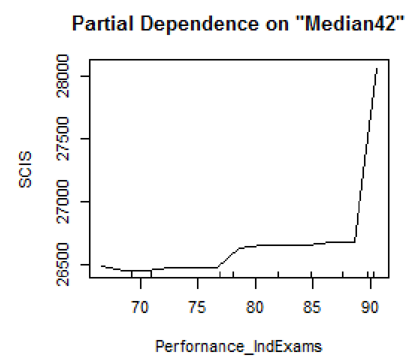

Partial Dependence on "Median19"
SCIS
PerComplexity_Neumuatic

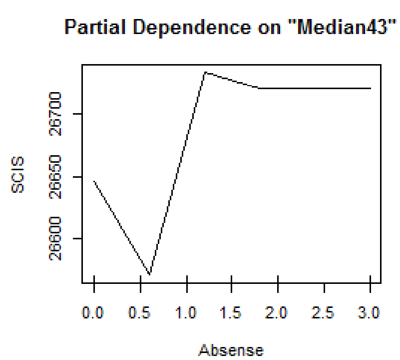

Partial Dependence on "Median24"
SCIS
Individual_ContributionPLC

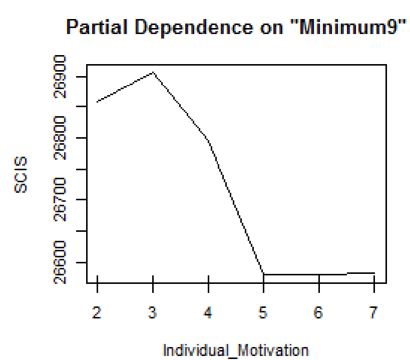

Partial Dependence on "Median25"
SCIS
Individual_ContributionProgramming

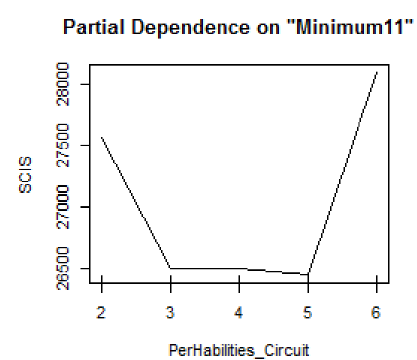

Partial Dependence on "Median28"
SCIS
Individual_ProjectHrs

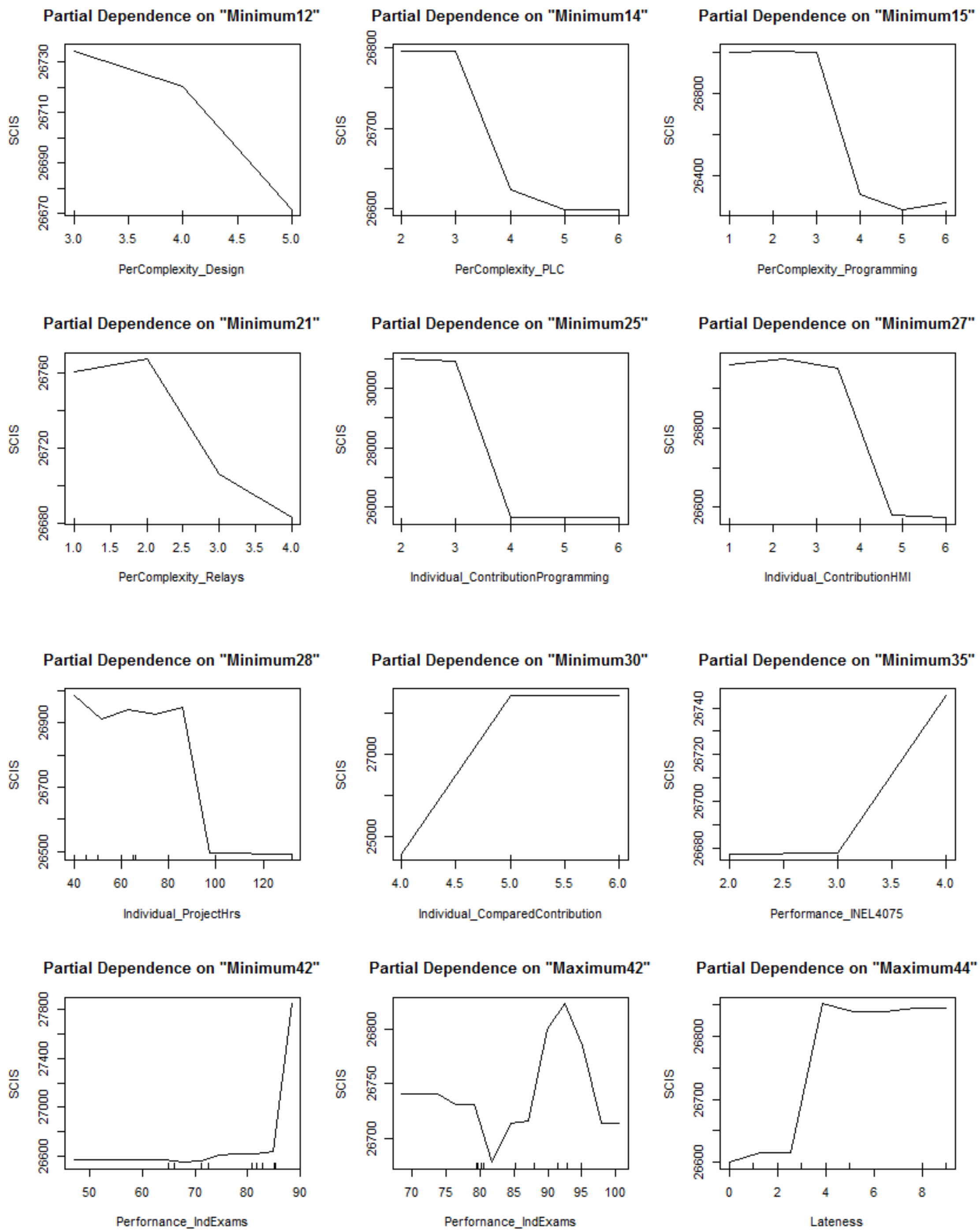

Partial Dependence on "Median41"
SCIS
Performance_INGE3016


Partial Dependence on "Median42"
SCIS
Perfornance_IndExams

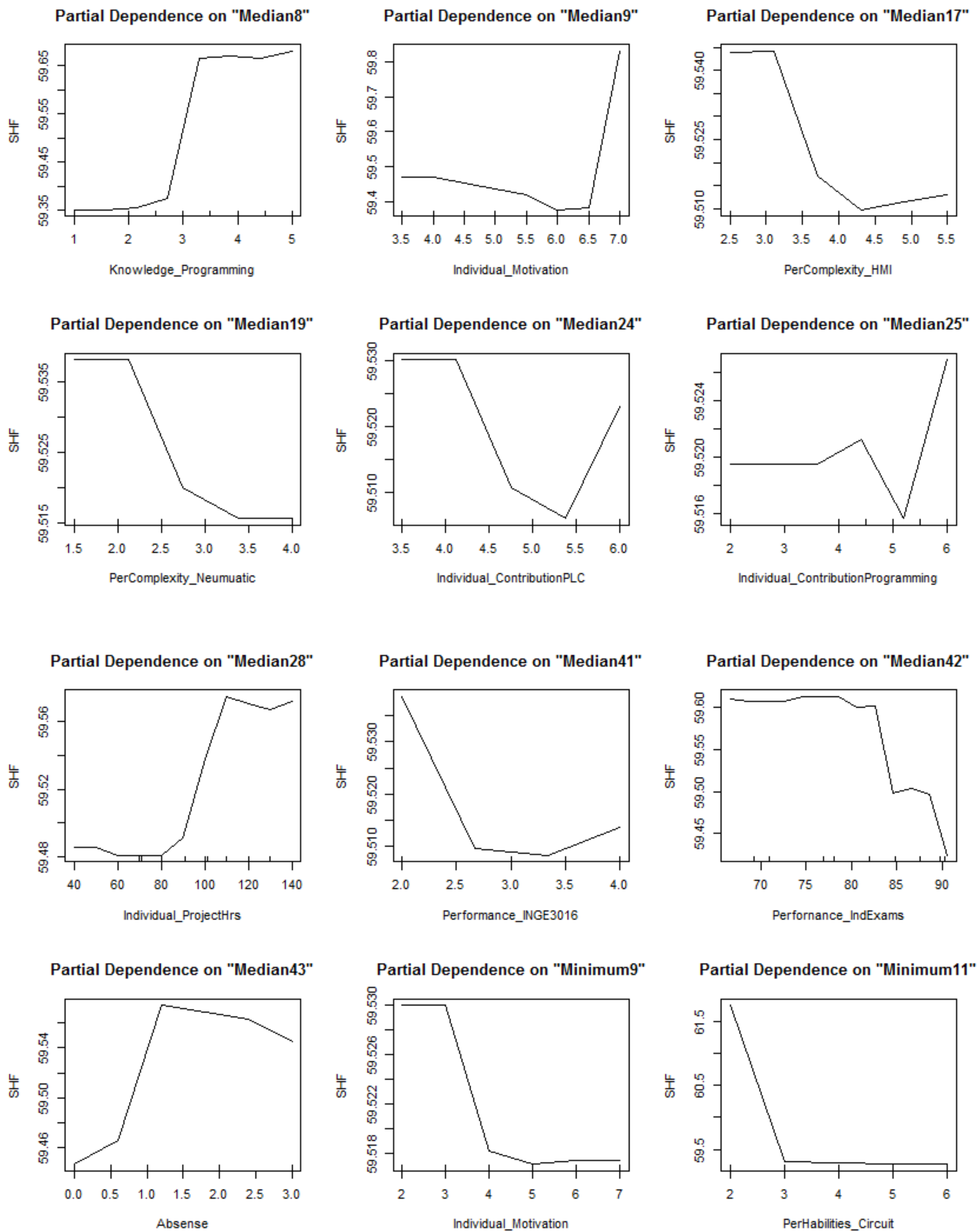

Partial Dependence on "Median43"
SCIS
Absense

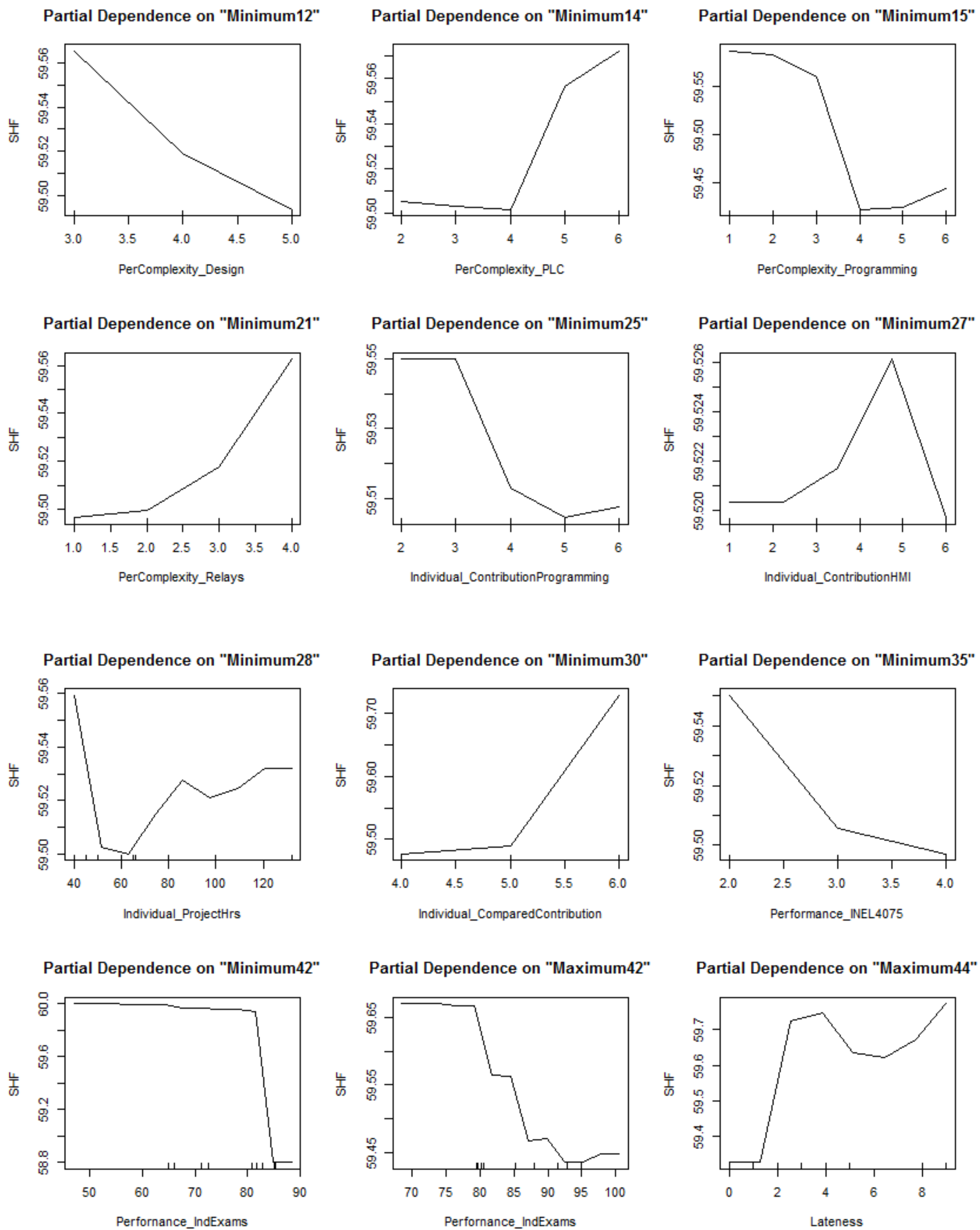

Partial Dependence on "Minimum9"
SCIS
Individual_Motivation

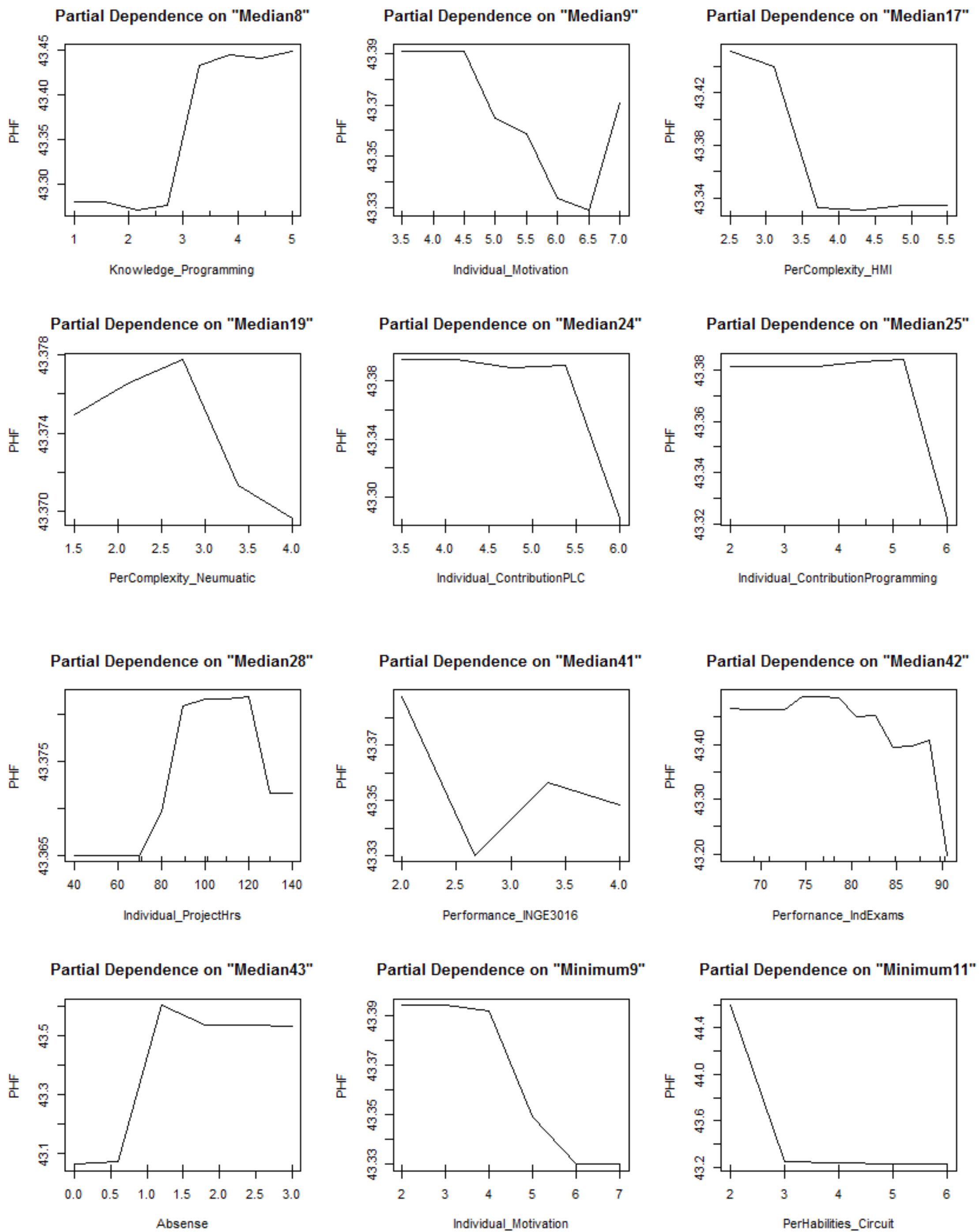

Partial Dependence on "Minimum11"
SCIS
PerHabilities_Circuit

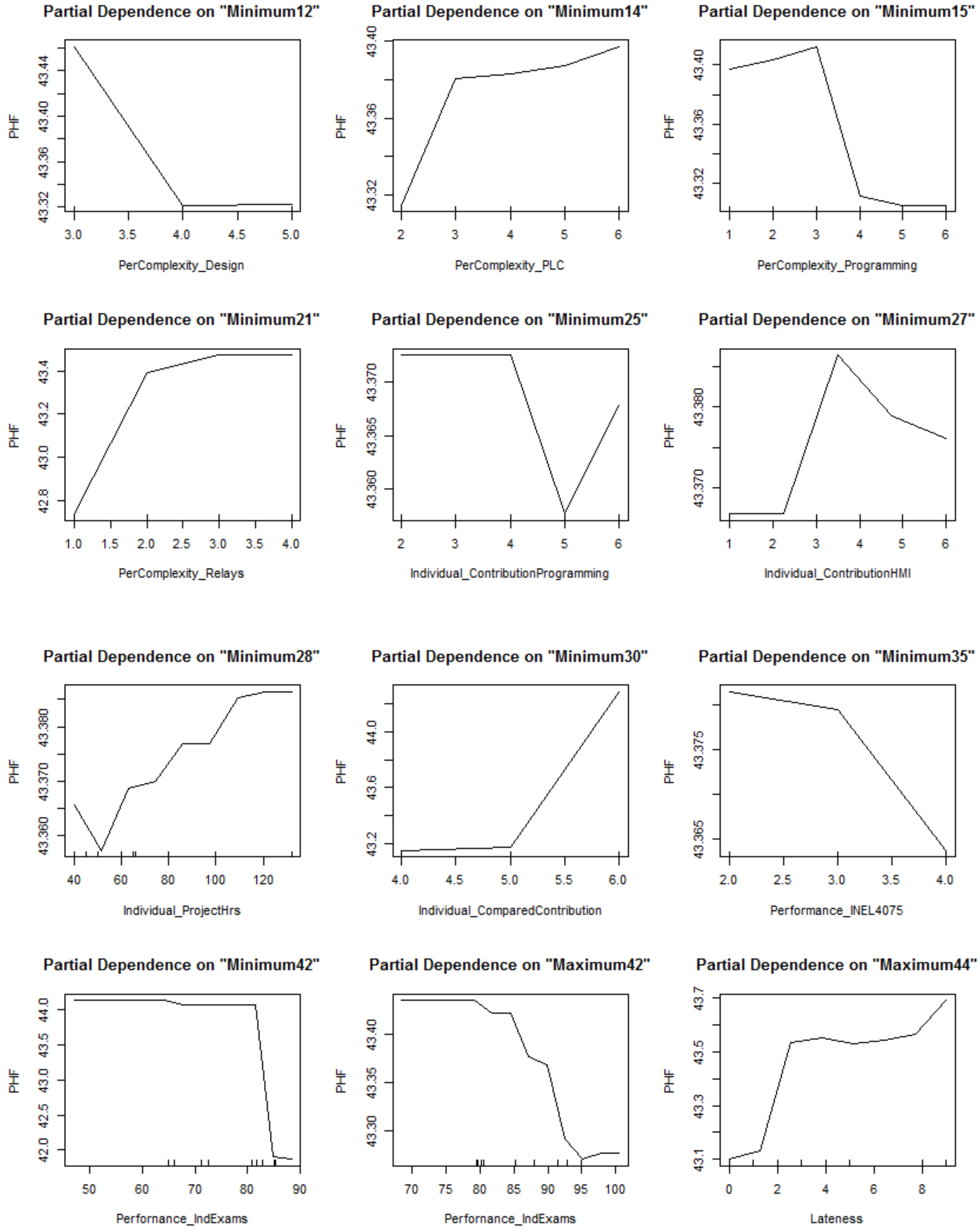


Partial Dependence on "Minimum12"
SCIS
PerComplexity_Design
Partial Dependence on "Minimum14"
SCIS
PerComplexity_PLC
Partial Dependence on "Minimum15"
SCIS
PerComplexity_Programming
Partial Dependence on "Minimum21"
SCIS
PerComplexity_Relays
Partial Dependence on "Minimum25"
SCIS
Individual_ContributionProgramming
Partial Dependence on "Minimum27"
SCIS
Individual_ContributionHMI
Partial Dependence on "Minimum28"
SCIS
Individual_ProjectHrs
Partial Dependence on "Minimum30"
SCIS
Individual_ComparedContribution
Partial Dependence on "Minimum35"
SCIS
Performance_INEL4075
Partial Dependence on "Minimum42"
SCIS
Perfornance_IndExams
Partial Dependence on "Maximum42"
SCIS
Perfornance_IndExams
Partial Dependence on "Maximum44"
SCIS
Lateness


Partial Dependence on "Median8"
SHF
Knowledge_Programming
Partial Dependence on "Median9"
SHF
Individual_Motivation
Partial Dependence on "Median17"
SHF
PerComplexity_HMI
Partial Dependence on "Median19"
SHF
PerComplexity_Neumuatic
Partial Dependence on "Median24"
SHF
Individual_ContributionPLC
Partial Dependence on "Median25"
SHF
Individual_ContributionProgramming
Partial Dependence on "Median28"
SHF
Individual_ProjectHrs
Partial Dependence on "Median41"
SHF
Performance_INGE3016
Partial Dependence on "Median42"
SHF
Perfornance_IndExams
Partial Dependence on "Median43"
SHF
Absense
Partial Dependence on "Minimum9"
SHF
Individual_Motivation
Partial Dependence on "Minimum11"
SHF
PerHabilities_Circuit


Partial Dependence on "Minimum12"
SHF
PerComplexity_Design
Partial Dependence on "Minimum14"
SHF
PerComplexity_PLC
Partial Dependence on "Minimum15"
SHF
PerComplexity_Programming
Partial Dependence on "Minimum21"
SHF
PerComplexity_Relays
Partial Dependence on "Minimum25"
SHF
Individual_ContributionProgramming
Partial Dependence on "Minimum27"
SHF
Individual_ContributionHMI
Partial Dependence on "Minimum28"
SHF
Individual_ProjectHrs
Partial Dependence on "Minimum30"
SHF
Individual_ComparedContribution
Partial Dependence on "Minimum35"
SHF
Performance_INEL4075
Partial Dependence on "Minimum42"
SHF
Perfornance_IndExams
Partial Dependence on "Maximum42"
SHF
Perfornance_IndExams
Partial Dependence on "Maximum44"
SHF
Lateness


Partial Dependence on "Median8"
PHF
Knowledge_Programming
Partial Dependence on "Median9"
PHF
Individual_Motivation
Partial Dependence on "Median17"
PHF
PerComplexity_HMI
Partial Dependence on "Median19"
PHF
PerComplexity_Neumuatic
Partial Dependence on "Median24"
PHF
Individual_ContributionPLC
Partial Dependence on "Median25"
PHF
Individual_ContributionProgramming
Partial Dependence on "Median28"
PHF
Individual_ProjectHrs
Partial Dependence on "Median41"
PHF
Performance_INGE3016
Partial Dependence on "Median42"
PHF
Perfornance_IndExams
Partial Dependence on "Median43"
PHF
Absense
Partial Dependence on "Minimum9"
PHF
Individual_Motivation
Partial Dependence on "Minimum11"
PHF
PerHabilities_Circuit


Partial Dependence on "Minimum12"
PHF
PerComplexity_Design
Partial Dependence on "Minimum14"
PHF
PerComplexity_PLC
Partial Dependence on "Minimum15"
PHF
PerComplexity_Programming
Partial Dependence on "Minimum21"
PHF
PerComplexity_Relays
Partial Dependence on "Minimum25"
PHF
Individual_ContributionProgramming
Partial Dependence on "Minimum27"
PHF
Individual_ContributionHMI
Partial Dependence on "Minimum28"
PHF
Individual_ProjectHrs
Partial Dependence on "Minimum30"
PHF
Individual_ComparedContribution
Partial Dependence on "Minimum35"
PHF
Performance_INEL4075
Partial Dependence on "Minimum42"
PHF
Perfomance_IndExams
Partial Dependence on "Maximum42"
PHF
Perfomance_IndExams
Partial Dependence on "Maximum44"
PHF
Lateness