UNIKASSEL
VERSITÄT

Embedded Systems

Tze Ying Sim

Effectiveness of the methods for engineering courses
in a large non-homogenous class setting
With regards to the specific disciplines - Computer Science
and Mechatronics in learning Embedded System

II Forschung 4

kassel
university
press

Embedded Systems

II Forschung 4

Herausgegeben von Prof. Dr.-Ing. Birgit Vogel-Heuser,
Technische Universität München

Tze Ying Sim

Effectiveness of the methods for engineering courses in a large non-homogenous class setting

With regards to the specific disciplines -
Computer Science and Mechatronics in learning Embedded System

kassel
university
press

This work has been accepted by the Faculty of Electrical Engineering and Computer Science of the University of Kassel as a thesis for acquiring the academic degree of Doktor der Ingenieurwissenschaften (Dr.-Ing.).

Supervisor: Prof. Dr.-Ing. Birgit Vogel-Heuser
Technische Universität München
Co-Supervisor: Prof. Dr. rer. nat. Albert Zündorf
Universität Kassel

Defense day: 12. Juli 2010

Bibliographic information published by Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at http://dnb.d-nb.de.

Zugl.: Kassel, Univ., Diss. 2010
ISBN print: 978-3-89958-970-2
ISBN online: 978-3-89958-971-9
URN: http://nbn-resolving.de/urn:nbn:de:0002-9711

2010, kassel university press GmbH, Kassel
www.upress.uni-kassel.de

Printed in Germany

Table of Contents

1 Introduction

1.1 Motivation

Starting from the 1950s, research had been crossing the boundaries of its own disciplines. Researchers constantly find themselves being stretched to boundaries beyond their knowledge [KA+07]. The mechanical engineers need to deal with electrical knowledge, the historians and the anthropologists ventured into each other area, so did the physician and the chemist. Examples of such boundary crossing fields are automation control, embedded systems, automotive industry, bio informatics etc. Research is not the only area that moves beyond its own discipline. With the emergence of multi-disciplinary technology, the industry is also moving toward this direction. Engineers working in the industry no longer can solve a problem without having some knowledge of inter connected discipline. An example is the window control system in the automotive industry. Before the implementation of electronic parts to control the vehicle's windows, the mechanical engineer can be certain that if the window fails to open or close, the problem is a mechanical problem. However, with the implementation of electronic control system, the problem may also lie in the electronic parts.

The changes from pure mechanical parts, to the fusion with electronic parts and the integration with control software bring new changes and challenges to the industry and research team. The team needs to have members from different disciplines or members with knowledge from different disciplines. Communications between team members from different disciplines are necessary. Communications can be a challenge as the terms by different disciplines are different, and even the same term can have different semantics. There are also research efforts to develop modelling notations to assist their communications, for example UML, SysML [We06], and SA/RT [HH+00]. Changes in organisation structure also support this new form of co-operation, for example instead of using the normal hierarchical reporting structure, another possible option is the matrix reporting structure.

Apart from changes in research and industry, the source where new engineers are shaped and produced, for example institutes of higher education, also need to make the necessary changes to meet this continuing market demands [PV09], [Bo04]. Among the suggestions are introduction of interdisciplinary courses, encouraging students to take courses from other departments, team work between different departments, etc. These proposals posed to be a challenge as the course content and methods need to be catered to groups of students with different knowledge [Bo90]. Another challenge facing institutes of higher education is the growing intake of students [Ba04]. Even though the number of students increases, the number of resources does not increase proportionally. Methods tackling these three problems need to be investigated. By understanding the cognitive mindset of different disciplines, suitable methods can be devised. These can be implemented not only in higher education institutions but also in industrial training.

1.2 Objective

The title of this dissertation is “Effectiveness of the methods implemented for engineering courses in a large non-homogenous class setting, with regards to the specific disciplines- Computer Science and Mechatronics students in learning Embedded System”. This research focuses on discovering suitable methods for Computer Science and Mechatronics students. The impact of the different methods implemented will be analysed specifically for both disciplines. Apart from this, the efforts and resources will also be recorded. Evaluation on the impacts of the methods implemented will be analysed based on the students’ exam grade. Other resources include course evaluation and group work evaluation. Feedbacks from these evaluation forms will be analysed to determine which methods are more suitable for the Computer Science and Mechatronics students. Suitable proposals can also be drawn for similar class size, group of students, or when the resources are restricted.

1.3 Structure of this Work

Chapter 2 defines the scope of this research. It first introduces the state of the art and different terms that would be used across this work. This is followed by introducing the scope where these methods will be implemented that is the area embedded system. Different challenges faced and current methods implemented in institution of higher education. Chapter 3 will further explore the area embedded system. The course content of embedded system will be presented here. This is to provide an outline on the range of subjects covered in the courses within this research.

Having the background on the subjects implemented, chapter 4 describes the different methods implemented in the courses. Concrete examples on how specific methods are implemented in specific subjects will be presented here. The fifth chapter describes the techniques used to assess the methods. The methods implemented in chapter 4 will also be evaluated using two evaluation forms. The evaluation categories will be described here. Apart from evaluation form, the methods are also accessed based on the final exam of the students. It is expected that students will score better grade when the suitable methods are implemented.

In chapter 6, different observations concerning the methods implemented will be presented. These observations are based on the techniques introduced in chapter 5. Using the data from final exam grade and the evaluation forms, hypotheses concerning the effectiveness of the methods and learning preferences of the students will be approved or disproved. The comparison of methods to the efforts required will also be conducted. Based on the analysis in chapter 6, chapter 7 will conclude on the methods implemented. The dynamics between the different methods will also be discussed.

Lastly, chapter 8 presents the summary and future works for this research.

2 State of the Art

This chapter will introduce the definitions and state of the art that are applicable to this research. As mentioned in chapter 1, research and industry are crossing the boundaries of its disciplines. Therefore, the term discipline, the different levels of co-operation between the different disciplines will be discussed in sub-chapter 2.1. The nature of this research falls in the category of interdisciplinary work. Examples of different interdisciplinary work and the courses offered will also be discussed.

The next sub-chapter 2.2 discusses an area that involves more than one discipline, and the core of this research – embedded system. The sub-chapter begins by defining an embedded system. This is then followed by the characteristics of an embedded system. The next section provides an overview on the subjects covered in the course embedded system by different institutions of higher education. A comparison with the subjects implemented in the course in this research will be conducted in section 2.2.3.

In order to prepare engineers that are able to work in teams with different disciplines, institutions of higher education are making the necessary changes. However, along with these changes come new challenges, sub-chapter 2.3 discusses the challenges faced by these institutions of higher education. The challenges discussed here are having class with different cognitive mind set, having large class size, and limited resources. Different methods under the term active learning are implemented to overcome the large class size problem. In order to have an overview on the available methods, the term active learning and the methods under active learning will be discussed in sub-chapter 2.4. As this research is in the context of teaching and learning in higher education, 3 learning pedagogies that provide guidelines in teaching and testing students will also be discussed in this sub-chapter.

The last sub-chapter zooms into the area where the methods proposed in this research takes place. This is the course embedded system. Firstly, the subjects covered in this course will be presented. Next, the students' background, namely the Computer Science students and the Mechatronics students will be presented. Finally, the class setting in relation to the three challenges presented in the previous sub-chapter will also be expounded.

2.1 Disciplines and Interdisciplinary Courses

2.1.1 Disciplines

The word discipline can mean a branch of knowledge, a specialised field of knowledge, to train by instruction and practise, to teach to obey rules, or a systematic way to obtain obedience. To a parent, the discipline of a child is determined by the way he/she behaves. This is not much different from the discipline in the education or professional field. A person is recognised to be associated in a particular discipline by his/her knowledge, way of thinking and the ability to solve problems from that particular field.

The idea of disciplines is also being strengthened by the establishment of faculties and department in colleges and university. The list of academic discipline continues to grow over time. The earliest disciplines are for example Theology, Arts, Language, Philosophy, and Chemistry. With the development of technology, Mechanical Engineering, Electrical Engineering and Computer Science were added to the list. Each different discipline prepares one for a specific profession. Further development in this discipline or profession leads to the word “specialisation”. For example, one can be a Computer Scientist who specialises in improving the quality of control software. Institutions of higher education, for example universities or colleges, teach and research on different disciplines.

With the span of time, it is no longer sufficient for one discipline to stay within its own field. An example supporting this idea is proposal of courses involving different disciplines [VDI90]. The research work has brought the researchers thus far to stretch beyond their own limits. The researchers continue to find themselves encountering connections in area of another research [KA+07], for example Engineering meets Humanities, and Chemistry meets Physics. This communication, interaction or relationship with another discipline is also known using the following few phrases “cross-disciplinary”, “multi-disciplinary”, “trans-disciplinary” and “inter-disciplinary”. The terms explored below are based on [Kl96] and [Da95]

- Cross-disciplinary: Efforts to view one discipline from the other but one is the subordinate of the other.
- Multi-disciplinary: Several disciplines working side by side, together and at equal weight, all contributing to understanding of a particular issue. However, no integration of knowledge is necessary.
- Inter-disciplinary: Work that involves two or more disciplines, creating a new entity or set of relationships. Integration of knowledge happens here.
- Trans-disciplinary: Way of thinking that transcends the current disciplines. This involves a new way of thinking, new way of organizing people, new way of communication – this will in turn spell new forms of cooperation between the disciplines.

Interdisciplinary teaching at a university involves integrated course content from different disciplines. This course can be addressed by a teaching instructor in a particular department, or teaching instructors from different backgrounds. The coupling of teaching instructors from different backgrounds to teach a course is widely known as team-teaching [Da95]. As this research focuses on interdisciplinary work, further elaboration on interdisciplinary courses will be discussed in the next section.

2.1.2 Interdisciplinary Courses

As the course in this research involves subjects that focus on the software development of hardware, and the responsibility of the different disciplines; the course in this research is an interdisciplinary course. Along with the advantages of interdisciplinary courses, it also brings along different challenges. The different challenges in an interdisciplinary will be elaborated in section 2.3.1, together with other challenges faced by institutions of higher education. This section will describe the different characteristics of an interdisciplinary course. Interdisciplinary courses are courses where more than one discipline is taken into consideration [DD07]. The emergence of interdisciplinary study is seen in students taking double majors, or major minor program from different disciplines; for example interdisciplinary courses like political science, or environmental science. Normally, interdisciplinary courses are conducted as combined efforts from other departments.

Interdisciplinary courses give a broader view in solving a problem. There are different views in solving a problem, for example the marketing personnel and the engineering team who needs to develop the product might not see eye to eye. It is more important for the marketing personnel to secure the sale and profit of the company. The promises to the customer might be done without considering the complications of engineering process, or the engineers might be interested to implement the latest technology without knowing the customers' response for this technology. In an interdisciplinary course, each discipline might get a view of what is important to the other group, creating a more holistic way in looking and solving a particular problem. With extra knowledge from another discipline, it is possible to implement a more holistic solution to the problems.

Interdisciplinary courses are not only important to gain the needed interdisciplinary knowledge but they are also useful to cultivate team work and communication among the future engineers. The purposes of interdisciplinary courses are to encourage the interaction between various disciplines, and that the students will learn to cooperate with colleagues from other disciplines. This is especially important to equip the students to work in teams in the industry. Many of the interdisciplinary group work involve students from different disciplines solving interdisciplinary problems [SV10]. The group works are for students from higher semesters.

The word interdisciplinary covers a wide range of cooperation between different disciplines. Some of the co-operation needs more co-ordination between different departments, and other co-operation needs lesser co-ordination. This research will group the different co-operations in two categories, namely closely coupled co-operation and loosely coupled co-operation. Closely coupled require more co-ordination, whereas loosely coupled require lesser co-ordination. Closely coupled interdisciplinary courses can be establishment of a graduate program that involves a few schools with selected courses designed for the students, the establishment of a course with contributions from different departments, having two or more teaching instructors working closely on the course content and conducting the class together. Group work within closely coupled co-operation context involves participation from students of different disciplines, working together to accomplish a task. They usually have two or more teaching instructors to contribute their expertise in guiding the students to complete a particular project [Ne94]. Loosely coupled interdisciplinary courses involve the mentioning or integration of another discipline in a single course, a teaching instructor with background from different fields can integrate the different knowledge in the single course and present it to the class [Da95]. Apart from that loosely coupled group work can be described as the involvement of students from different discipline in a group work that is conducted from a single department or the involvement of students from the same discipline on a group work that involves knowledge from different discipline.

The interdisciplinary courses explored here fall in the loosely coupled interdisciplinary course. The course content is a combination of electrical engineering and computer science subjects. Therefore, the course content itself is interdisciplinary. The students in the course are from different disciplines, namely Computer Science and Mechatronics. This also contributes to the interdisciplinary aspect of the course. However, the course is only being handled by one teaching instructor and one teaching assistant from the Electrical and Computer Science Department. There is no close co-operation between the Electrical Engineering and Computer Science Department, and the Mechanical Engineering Department in conducting the course. Therefore, the course explored in this research falls into the loosely coupled interdisciplinary course.

It is essential to understand and to implement effective teaching methods that will benefit the students from various disciplines. No detailed study is available for interdisciplinary course between Computer Science students and Mechatronics students in a large class environment.

2.2 Embedded System

This sub-chapter will introduce an interdisciplinary subject – embedded system. The different definitions for an embedded system, its characteristics and the subjects normally taught in this subject will be the foundation on the methods that will be implemented in this research.

2.2.1 Definition of an Embedded System

An embedded system consists of software and hardware unit, with a specific function and very often needs little intervention from its user. The term microprocessor and microcontroller are often related to the subject embedded system. Typical embedded system architecture consists of application specific integrated circuit (ASIC), digital signal processor (DSP), encoding and/or decoding devices (CODEC), microcontroller, control panel, system bus and memory [EL+97]. Among the common programming languages used to write the software for an embedded system are assembly code, VHDL, ADA, C, C++, Lisp, Pascal, and FORTRAN. Among the definitions for an embedded system includes:

- *“Embedded system is a vast field encompassing numerous disciplines…state-of-the-art trends (are) such as … ASIPs, SoC Communication …, testing of core-based integrated circuits”*[Zu06].
- Microcontroller(s) that are incorporated in a product, for example a washing machine or a television, and is firmly part of the product. The open/close-loop control should be kept as simple as possible for the user [LG99].
- A (Micro-) Computer system that is embedded in a technical system but by itself is not a computer [HV05].

These definitions are the “more common” definition. When asked to name an embedded system the answer that come across the students mind are microprocessor, microcontroller, microwave, cash dispenser, printer, robot, programmable telephone etc. However, embedded system is not limited to this definition. [BvdB+98] describes an embedded system as

- A software/hardware unit that is connected to the main system using sensors and actuators. The main task is to monitor, control and regulate the system. An embedded system often deals with a reactive system and also hybrid distributed system. These systems have real-time requirement. Typically, the interaction between user and such system are not directly apparent. The user interacts with the embedded system subconsciously.

  “*Ein eingebettetes System (abgekürzt: ES) ist eine Software-/Hardware-Einheit, die über Sensoren und Aktuatoren mit einem Gesamtsystem verbunden ist und darin Überwachungs-, Steuerungs- beziehungsweise Regelungsaufgaben übernimmt. In der Regel handelt es sich bei eingebetteten Systemen um reaktive, häufig auch um hybride verteilte Systeme mit Echtzeitanforderungen. Typischerweise sind solche Systeme dem menschlichen Benutzer nicht direkt sichtbar, er interagiert unbewusst mit dem eingebetteten System*.“ [BvdB+98]

The definition by [BvdB+98] shows that the tasks for an embedded system include open-loop control, close-loop control, monitoring and data processing. Depending on the task, the real time requirements will be determined, for example soft real time, hard real time or hybrid real time systems. The tasks are loaded on the controllers before being executed in the technical systems. If there are any changes, the programs need to be reloaded on the controller before the changes can take place. The controllers for such embedded system are for example process control system and automation control system.

From the definitions above, it is observed that there is no one “ultimate” definition for embedded system. An embedded system can be as simple as a calculator and can be as complicated as the controller in a manufacturing, process technology factory, or even a control system for a nuclear plant. According to [LG99], an embedded system can be a product or a production plant. An embedded system can range from a simple microprocessor, a product with microprocessors or microcontrollers, to the complicated control of a factory. Therefore, the term embedded system covers a large range of applications with involvement from numerous disciplines [Zu06].

During the first part of the course in this research, the students are introduced to the basic components of an embedded system, and in the second part of the course, the focus is on the development of an embedded system as defined by [BvdB+98]. An embedded system has three components, the controlling device, the controlled physical devices, and the user interface. The controlling device is the embedded hardware or software that will interact with the user interface and the physical devices. The physical devices are various actuators and sensors. The user interface allows the user to key in the necessary command and also returns specific values from the system. The physical devices and the user interface will interact with the user and the environment.

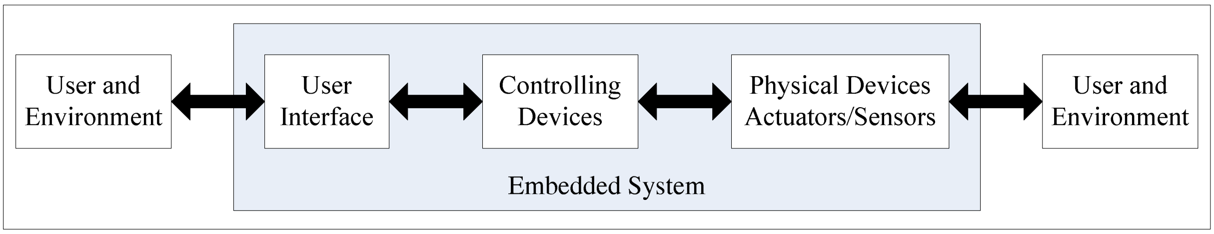


Figure 2.1 - Embedded System after [BvdB+98] definition

2.2.2 Characteristics of an Embedded System

This sub-section will further elaborate on the characteristics of an embedded system. The two main areas to be discussed here are embedded system as an interdisciplinary field and the requirements of an embedded system. It is important to discuss the characteristics of an embedded system, as this further support that embedded system is an interdisciplinary field and it also provides and overview on what are the important characteristics that needs to be covered in the course.

Embedded system as an interdisciplinary field:

With the invention of computers and microprocessors in the 1940s, many applications that were once implemented using the physical systems are now transferred to the software that controls the different hardware. "*There are many examples of embedded systems in the real world. For instance, a modern car contains tens of electronic components (control units, sensors, and actuators) that perform very different tasks. The first embedded systems that appeared in a car were related to the control of mechanical aspects, such as the control of the engine, the antilock brake system, and the control of suspension and transmission. However, nowadays cars also have a number of components that are not directly related to mechanical needs of the passengers: navigation systems, digital audio and video players, and phones are just a few examples.*" [Zu06]. Here the expertise of mechanical engineer, electrical engineer and computer science is required to solve the problem. Integration of knowledge is needed in this example of embedded system.

Another evidence of this integration is the emergence of “Mechatronics”. In the late 1960s, Japan’s Yaskawa Electric Company introduced the term Mechatronics [To02]. European universities had been involved with Mechatronics education since 1970s and they are taught in form of “embedded system” [Ac97]. In an interdisciplinary project, SFB 614 (Sonderforschungsbereich), the term Mechatronics is described as close interaction between mechanics, electronics, control engineering and software, to improve the behaviour of a technical system. Using information from sensors, data concerning the environment and the system will be collected and processed in computers. This will lead to the control or actuators and thus influencing the system.

“Der Begriff Mechatronik bringt dies zum Ausdruck. Gemeint ist hier das enge Zusammenwirken von Mechanik, Elektronik, Regelungstechnik und Softwaretechnik, um das Verhalten eines technischen Systems zu verbessern. Dafür werden mit Hilfe von Sensoren Informationen über die Umgebung, aber auch über das System selbst erfasst und in Rechnern verarbeitet. Dies führt zur Ansteuerung von Aktoren und somit zur Beeinflussung des Systems ” [GRS06].

Mechatronics (or embedded system) today involves at least four disciplines, namely mechanical engineering, electrical engineering, control engineering and computer science. Changes in one area will affect the other discipline. Communication between disciplines should be encouraged. There are also various research looking into developing a common modelling notation that can be understood by various disciplines [SL+09], [SW09], [Sc07].

The percentage of control software in the different system also increased over time [SA03]. Therefore, it had been increasingly important that the electrical engineers and mechanical engineers also understand the methodologies available in software development. It is also important that the computer science students not only learn about the development of a computer program, but it will be more relevant if the interactions with the hardware are also taken into the curriculum’s consideration [To02].

Requirements of an embedded system:

As compared to pure software, software in an embedded system has extra criteria. The software in the embedded system mentioned above, whether the software for a microprocessor or control system have the additional criteria as below:

- Time critical – many control systems are time critical. The delays or acceleration in milliseconds can cause heavy losses to the industry. Time critical requirement can further be divided into two categories – hard real time and soft real time requirement. Hard real time requirements will be useless if the time requirements are not met. In soft real time system, the usability of the system decreases as it wavers further from the time requirements.
- Distributed – the architecture of control system are often distributed over a few controllers. Consideration should also be given to the organisation of sensors,

actuators and operating interface in the architecture. With this, other aspects like secured network and real time network should also be considered.

- Interface with other hardware and system – embedded system needs to interface with hardware system and other control system. Unlike pure software application, where the interaction is at most with computer peripherals, an embedded system interacts with the sensors and actuators and other embedded system on the network.

The implementation of software in the control system is still not as mature as the software engineering itself in the classical Computer Science domain. There are various projects undertaken to implement the methods in software engineering in the control system, for example the implementation of requirements engineering [Va07]; and software modelling in the development of these control systems [FV07], [KV09], [LC06]. Different adjustment and changes are done to adopt the available methods in the control system field.

2.2.3 Subjects in Embedded System

The importance of embedded system will continue to grow [Ma06]. Traditional education focusing on mostly hardware, as in electrical engineering programs, or on mostly software, as in computer science, will not be sufficient to cater for the growing demand of embedded system. As embedded system has wide diversity and a wide span of complexity levels, it is not easy to teach embedded system as a unified topic [KC+05]. Apart from that embedded system in education is still being defined and redefined [GT05].

There are various ways to approach teachings of embedded systems. There are universities who split the course across a few semesters teaching from the fundamental level to the advance level [KC+05], [YT+05]. Students go through various courses instead of one "embedded system" course. There are universities who offer embedded system as a one time course with other pre-requisite courses before the students can attend the "embedded system" course [Ma05]. Embedded system can also be a four year course that is covered through various courses [RS+05].

ZVEI in its report mentioned that students who have bachelor degree should be ready for the working market. Therefore, it is important to also have practical work for the bachelor students [ZVEI04]. "*European Commission launched in 2001 the Artist FP5 Accompanying Measure, gathering more than twenty top academic institutions and laboratories … to look into the question on how to design an "ideal" graduate curriculum for graduate students.*"[CS+05]. Part of the graduate curriculum proposed by this group is as follows:

- Foundations of Computer Science and Engineering – basic algorithms, basic notions on gates and assembly language, element of language theory (automata, regular expression), basic operating system, software modelling. This knowledge should be acquired during under graduate studies.

- Basic Control and Signal Processing – motivation on the interaction with physical system, especially for computer science students (control theory and feedback theory). Signal modelling, state and feedback, sampling and sampled control, discrete-event control, hybrid-control theory. The training to use tools such as Matlab/Simulink is also necessary.
- Theory in Computing –detonational semantics, axiomatic semantics, and structural operational semantics.
- Real Time Computing – taxonomy for real-time applications (for example soft and hard real-time), operating systems (time- and event-triggered systems, preemptive and non-preemptive scheduling, basic scheduling algorithms), compiler, languages (asynchronous and synchronous paradigms), design and validation.
- Distributed Computing – distributed algorithms, networks, design (VHDL and FGPA design, CAN and Ethernet networks), and algorithms (safety critical).
- Evaluation and Optimisation of Extra functional Properties – performance (QoS management), dependability (Petrinets, Markov chains, and fault trees), power consumption, memory and stack usage, execution time, trade-offs.
- System Architecture and Engineering – design methodologies, modelling, verification, and group work.

From Artist Education Group working group's proposal for graduate's program proposal, the "Foundation of Computer Science and Engineering", "Basic Control and Signal Processing", "Real Time Computing", "Distributed Computing" and "System Architecture and Engineering" are covered from the various undergraduate program compiled in Table 2.1.

Table 2.1 - Subjects taught in Embedded System class

	[PR05][1]	[SP05][2]	[Se05][3]	[RS+05][4]	[YT+05][5]	[Ma05][6]	[Mu05][7]	[Ch10][8]	[Ko10][9]	[Da10][10]	Research[11]
Introduction											
Introduction to Embedded Systems	X	X		X		X	X		X	x	X
Basics of Embedded System	X	--	--	--	--	--	--	--	--		X
Combinational/Sequential Circuit Design	X	--	--	--	--	--	--	x	--		X
Embedded Systems Hardware Architecture											
Microprocessors, Microcontrollers	--	--	--	x	--	X	x	x	X		X
Processors Architecture	--	--	--	x	--	X	x	x	--		--
Hardware in Loop	--	--	--	--	--	X	--	--	--		--
AD/DA Converter	--	--	--	X	--	X	--	--	x		X
Clocks	--	--	--	X	--	--	--	--	x		X
Embedded System Software Architecture											
Operating system - RTOS, Windows CE, VxWorks etc	--	--	X	X	--	X	X	X	X		X
Application Software	--	X	--	X	--	X	X	--	--		X

[1] Dongduk Women's University, Sungbuk-gu, Seoul, Korea; Sungkonghoe University, Kuro-gu, Seoul, Korea; Sangmyung University, Jongno-gu, Seoul, Korea; Konkuk University, Gwangjin-gu, Seoul, Korea

[2] University of California at Berkeley

[3] University of Waterloo, Ontario, Canada

[4] University of Alabama, Tuscaloosa

[5] Nagoya University, Japan

[6] University of Dortmund, Germany

[7] Hong Kong University of Science and Technology, Hong Kong

[8] Technische Universität München, Germany

[9] RWTH Aachen University, Germany

[10] Department of Computer Science, University Oldenburg

[11] Department of Embedded System, University of Kassel

Table 2.1 - Subjects taught in Embedded System class (continue)

	[PR05]	[SP05]	[Se05]	[RS+05]	[YT+05]	[Ma05]	[Mu05]	[Ch10]	[Ko10]	[Da10]	Research
Middleware	--	--	--	--	--	X	--	--	--		--
Interrupt - Interrupt Handlers, ISRs etc	--	--	--	--	--	--	X	x	x		X
Scheduling - Rate Monotonic, Priority Inheritance, Priority Inversion etc	--	--	--	--	--	--	X	X	x		X
Communications											
Bus system - CAN, LIN, RS232 etc	x	--	--	X	--	--	--	X	X		X
Asynchronous vs. Synchronous	x	--	--	--	--	--	--	--	x		X
TCP/IP	x	--	--	--	--	--	--	--	--		--
Development of Embedded System											
System Requirements	X	--	X	X	--	--	x	--	--	X	X
System Design/ Modelling/ Specifications Techniques - Petrinets, UML, SysML, SDL, VHDL, State Charts etc	X	X	X		--	X	X	X	X	X	X
System H/W Design	X	--	X		--	--	x	--	--	X	--
Hardware/Software Co-design	--	--	X	X	--	X	--	--	--	x	--
System Programming - Assembly Language, ANSI C, PEARL, IEC 61131-3 etc	--	--	X	--	--		--	--	X	x	X
Hardware/Software Co-verification	--	--	X	--	--		--	--	--		--
Verification, Validation and Testing	--	--	X	--	--	X	X	--	--	X	--

Legends: X – Clearly mentioned in the references, x – author's deduction based on the description in the references

From Table 2.1, it is observed that the subjects for embedded systems can be divided into two categories, the fundamentals of embedded system and the development of an embedded system. The fundamentals of embedded systems include the introduction to embedded system, the hardware architecture, the software architecture and the specific means of communication. The other category is the development of an embedded system. This covers the different lifecycle phases in the development of an embedded system. The phases are system requirements gathering, system designs, system implementation, system integration, and validation and test. Two subjects that specifically look into the balance and trade off between hardware and software systems are hardware/software co-design and co-verification [LP06].

2.3 Challenges for Institutions of Higher Education

In order to accommodate to the demands of industry and research, institutions of higher education had been offering the students, interdisciplinary course work, team work opportunities etc. However, institution of higher educations faces different challenges in the process of changing and accommodating to these new interdisciplinary needs. Three challenges, which are relevant to the institution of higher education and this research, will be presented in this sub-chapter, namely different cognitive mindset, large class size and limited resources.

2.3.1 Different Cognitive Mindset

People learn by connecting different ideas together. Cognitive scientist demonstrated that learning is a process of drawing connections on what people have already known. Therefore, students with different backgrounds will associate the new knowledge differently. As students from different disciplines have different ground courses, they would have different cognitive mind set [DD07]. [BS+08] mentioned in their research that the main problem areas of the course technique programming module are the different previous knowledge of the participating students. The module is boring for one and overstrains the other group.

Students from different disciplines might have different description for the same term. For example the word “model” for Computer Science students can mean a software model, whereas the Mechatronics students will mean a hardware model.

2.3.2 Large Class Size

The second challenge is the large class size. The class size in the universities starts to grow when higher education ceases to be “only for the elites” but also “for the masses” [Bo90]. There are different definitions for “large” and “small” class size. According to [GL+96], small class is defined as 30 or fewer students, whereas large class is defined as 70 or more students. [Ku07] defined small as less than 21 students in a class, medium as between 21 and 75 students, and large as more than 75 students. [DD+06] classified that small class has less than 35 students, small-medium class has between 36 and 50 students, moderately large has between 50 and 70 students, large is between 70 and110 students and very large is more than 110 students. There is also class size with more than 500 students. This is possible for foundation classes in bigger universities, for example the class “Information Technology” in Technische Universität München. No specific research on methods implemented for more than 150 students as compared to 500 or more students is found.

The methods implemented in this research will focus on large class size. For further references of class size, the classification by [DD+06] will be used (Table 2.2). This definition is more detailed and yet did not contradict the definition from [GL+96] and [Ku07].

Table 2.2 – Categorisation of class size [DD+06]

Size	Category
<= 35	small
30 < x <= 50	medium
50 < x <= 70	moderately large
70 < x <= 110	large
> 110	very large

It is important to identify the different category of class size, as the different class size has different impact on teaching and learning. According to different research, the numbers of students do influence the learning behaviour, exam result and feedback to the evaluation [Ku07], [AW04], [CR00], [Fo94]. The "Kindergarten to 12th Grade program" (K-12) also recognises this and different studies had been conducted to make the classes more effective [BS+04].

Large class students are more difficult to handle. Large class environment is more impersonal as compared to small class. According to [Ge92], 12 feet is the comfortable distance for social distance and 25 feet is reserved for public speaking. In today's large class environment, the distance between the teaching instructor and teaching assistant is more than the social distance. On top of that, the students are seated in rows that are inconvenient for movement. Neither can the student move easily to the front, nor can the teaching instructor move easily between the students. The sitting arrangement in this type of lecture hall is very similar to a cinema, and can also influence the students to only be an "audience" in the class. There are reported incidents where students behave rudely in a class, being disrespectful and appear uninterested [Ca99]. There are students who are working on other courses assignments, reading news, talking, and falling asleep. These behaviours are discouraging not only to the teaching instructor but also to other students who are interested in the class.

Smaller class size will have positive correlation with the exam grades [Ku07], and [AW04]. [KD+08]. The study on K-12 also shows a negative correlation between the class size and the students' achievement. [GL+96] analysed the relations between class size and student performance over a period of 10 years. One of the reasons for poorer grades is due to the poor attendance. [CR00] mentioned that the class attendance for a large class usually dwindles to 40% or 30% at the end of the course. The research from [DF96] shows that the attendance has a positive regression with the students' grade. The more the students attend the course, the more they are going to or are able to appreciate the subject, and thus will also normally do better in the exam.

There is a negative regression between students' evaluation of teachings (SETs) and the class size. [Ku07] mentioned that the course ratings are better if the class size is 20 or less, and the SETs within this small group decreases as the class size increases. However, according to the same study, the class size does not influence SETs when the class size is bigger than 20 students. This is because there is no significant difference between the value of medium and large size class SETs. However, when compared to the small class, the medium class and large class have poorer SETs' value. [Ha86] reported that there is no difference in SETs value for the teaching instructor who handles both small and large class. However, the grading for the teaching assistant decreases as the class size grew. This is attributed to the lack of experience by the teaching assistant in handling large classes. This shows that it is important to have experienced teaching assistants to handle large classes.

The most common pedagogy form for large class setting is through lecture. A traditional lecture takes place when a lecturer recites or passes on the knowledge in verbal form while the students take notes [Mi99]. Among the challenges in a large class settings are it is more difficult to control the class behaviour [Ca99], to follow up on students' understanding, and to encourage students' participation in the class. In order to overcome these problems, various methods under the term "Active Learning" will be introduced. This will be further elaborated in sub-chapter 2.4.

There are several methods to implemented exercises in large class size environment. The exercises can be conducted in the same setting as the lecture, in a large class size environment or be broken into smaller groups. Exercises that are conducted in the large class size environment face the same challenges as the lecture. If the institute have more resources, the students can be broken into smaller groups and the exercise session can be conducted separately. Meaning the lecture is conducted in a large class environment but the exercises are conducted in smaller groups. This opens up discussion opportunity concerning the subjects that are still unclear in the exercises session. Another method is utilising the internet [GG+07]. This is more than just uploading the content onto the internet and providing the answers. According to Schulmeister, collaboration and co-operation are important for virtual learning [Sc05]. Students can log into the portal to solve the exercises or discuss the problem with other students or the teaching assistant. However, this requires resources to maintain architecture and the content of the web page.

2.3.3 Limited Resources

The third challenge for interdisciplinary class is the extra coordination that is required. Table 2.3 presents a compilation of interdisciplinary courses that implemented different methods.

Table 2.3 - Compilation of interdisciplinary courses in terms of number of participating disciplines, contributing departments and number of students with examples of methods implemented

Disciplines	Source	Number of Students	Disciplines	Contributing Dept.	Lab work	Remote Lab/ LMS	Traditional Lecture	Group work	Project with External	Active Learning Lecture	Exercise	Muddy card	Flash Card
General	[Ja07]	M	-	-	-	X	-	X	-	-	X	-	-
	[SS+05]	S	-	-	-	-	-	X	-	-	-	-	-
Religion, English	[CE+95]	M	2	2	X	-	X	-	-	-	-	-	-
English, Philosophy, Sociology	[Ma00]	L	2	3	-	-	-	X	-	-	-	-	-
English, US History	[Ma00]	L	2	2	-	-	-	X	-	-	-	-	-
Sociology	[AB05]	-	5	5	-	-	-	-	X	-	-	-	-
Medical	[CC02]	-	1	3	-	-	-	X	X	X	S	-	-
Engineering	[SB02]	L	4	4	-	-	-	X	-	-	-	-	-
	[HW+02]	M	1	1	-	-	-	-	-	X	S	X	X
	[GP06]	M	1	2	-	-	-	X	-	X	-	-	-
	[PD07]	S	1	1	-	-	-		-	-	-	-	-
	[SE07]	S	1	1	-	-	-	X	-	-	-	-	-
	[Mi08]	S	1	1	X	-	-	X	-	-	-	-	-
Computer Science, Business	[Mi99]	L	2	3	-	-	X	X	X	X	S	-	-
Mechatronics	[GH01]	-	1	1	X	-	-	-	-	-	-	-	-
	[WT+01]	-	1	1	-	-	-	X	-	-	-	-	-
	[To02]	-	1	1	X	-	-	-	-	-	-	-	-
	[SM02]	-	1	2	X	-	-	X	-	X	-	-	-
	[EB03]	-	1	2	X	X	-	-	-	-	-	-	-
	[MS93]	M	1	1	X	X	-	-	-	-	-	-	-
Mechatronics, Computer Science	Research	L	2	1	X	-	X	X	-	X	M	X	-
Total for Implemented Methods					8	3	3	13	3	6	5	2	1

Legends: L = Large Size Class; M = Medium Size Class; S = Small Size Class; - = No Information Available

From the compilation of different research as shown in Table 2.3, most of the interdisciplinary classes involve more than one discipline that is represented by different departments. In the case of team-teaching, two teaching instructors will teach a class together. The advantage of involving different departments is the load is distributed across the different departments. With these available resources, it is also possible to conduct group work and involves the student more actively in the learning process. The last entry is the research conducted in this dissertation, unlike most large interdisciplinary classes, the number of contributing department is only one; whereas the number of participating disciplines is two.

2.4 Active Learning

Different methods under the term active learning are implemented to overcome large class size problem. Apart from this active learning also encourages interaction between teaching instructor/assistant and students, as well as interaction among students. This would help students to understand fellow colleagues from another discipline better. This sub-chapter is going to present the different active learning methods, and the challenges faced in active learning.

2.4.1 Active Learning's Methods

Active learning means the students not only listen and take notes in the class but they also have the opportunity to participate actively in the class [BE91]. Among the methods implemented in active learning include informal group learning, formal group learning, group work, problem based learning, team-teaching, cold calling, in-class demonstration, muddy card, flash card, concept test, evaluation form, pause method, laboratory work, and learn management system [HW+02], [Mi99], [BS96], [SS+05], [Da95]. The various methods mentioned above can be divided into two categories, those that can be implemented in class and those to be implemented outside the class (Table 2.4).

Table 2.4 - In Class Implementation and Out of Class Implementation

Methods	In Class	Out of Class
Informal / Formal group learning	X	--
Group work	X	X
Problem based learning	X	X
Team-teaching	X	--
Cold calling	X	--
In class demonstration	X	--
Muddiest card	X	--
Flash card	X	--
Concept test	X	--
Pause method	X	--
Laboratory work	--	X
Learn management system / Internet	--	X

Informal Group Learning: This is group formed informally during lecture or exercise. The teaching instructor will spontaneously break the class into small groups, normally in groups of 2 or 3, as this is the limitation of lecture style sitting. The group will only be valid for this discussion. Each group will discuss a certain topic or work on a problem. It is also possible that a representative will need to present the findings in the class.

Group Work: The students work in groups to solve a particular problem, assignments or projects. Group work can be conducted in the class or outside the class. Group work requires co-ordination among the students. This presents an opportunity for the students to develop socially as well.

Problem Based Learning (PBL): The teaching instructor provides an open ended problem and the students will need to solve the problem according to the experience and knowledge. The question should be a problem that the students are interested in solving. Problem based learning is usually conducted in groups. The students will discuss ideas or hypothesis that can be used to solve this problem. The teaching instructor acts more like a facilitator through out the discussion. As in group work, PBL also involves the social skills. The students learn to solve problem individually.

Team Teaching: At least two instructors from different fields work together to conduct a class at the same time to a single group of students. Discussion between the instructors can take place “live” before the students. Students are able to experience the view from the different fields.

Cold Calling: The teaching instructor simply calls a student in the class to answer a question or give an opinion. No name is necessary for this method, description or appointing by position, for example the third student from the right in the last row, can be used. Another method of cold calling is round robin calling. This is implemented by calling the student sitting next to the current student, and again the student next to him/her, and so on.

In Class Demonstration: The teaching instructor uses different objects or even the students themselves to explain the subject or to draw students’ interest to the subject. The objects may be hardware, charts, tools, or acting the process.

Muddy Card: Students are requested to use 2 to 5 minutes to write down the areas that they did not understand (muddiest part of the class) on a small card. The teaching instructor would then answer the questions in the next class. Another variation is to post the answers on the internet or provide answer sheets. Typically, students will be given a small card to write their feedback. This is normally done at the end of each lecture session.

Flash Card: The students are give a few cards in different colours, for example a red card and a green card. The teaching instructor will pose a question, the students will then show one of the cards to represent their answer, for example red for disagree and green for agree. This can also be done electronically by providing a computer system (personal response system) at every seat in the lecture hall. [KC05] mentioned that students who use of personal response system performed better in the class base assessment. However, these students are also the students who are more motivated and attend class more frequently.

Concept Test: The teaching instructor will pose a question that will encompass the important concepts for the particular class at the end of the class. The students will try to answer this question and hand up the answer. Through the students' answer, the teaching instructor will be able to access the students' understanding. Concept test can be coupled with flash card method to acquire students' feedback.

Pause Method: The teaching instructor will pause for a few seconds between sentences, glancing across the students, and giving room for students to ask questions if any. Pause method can also be use to give emphasis. Change of tone and speed in presentation makes the presentation more interesting.

Laboratory Work: The students receive an assignment to be completed in the laboratory. The whole class can also be conducted in the laboratory. What is important is that the students get to do practical work.

Learn Management System: Course content can be uploaded on a learn management system, there might be an online "laboratory", or tutors who can chat and assist students. Students have the opportunity to revise the course content and to track their progress.

2.4.2 Active Learning's Challenges

Implementation of active learning methodologies may require more effort and resources than traditional lecture [BE91]. The researches undertaken to implement active learning in large class environment showed that much resources are required. For example more than *"50% of the college's 150+ faculty member provided input"* to enrich the undergraduate program with hands-on, project based learning [CS99], personal response system is installed for concept test, and extra 1 to 1.5 hours are needed to respond to muddy cards [HW+02]. Bonwell and Sutherland mentioned four possible obstacles for implementing active learning, namely limited class time; a possible increase in preparation time; the potential difficulty of using active learning in large classes; and a lack of needed materials, equipment, or resources [BS96]. [HW+02] described that active learning will increase the preparation effort as there might be lack of materials and resources. The teaching instructor needs to spend time designing ideas that are suitable for the course content.

Secondly, the students might not be familiar with this method and be less than willing to cooperate. Students, who are used to only listening in the class, might feel uncomfortable and not willing to participate and voice their opinion in the class. The change of role and responsibility would need some getting used to. This is compounded with the large class environment, where students sit in rows and it is not conducive for group discussion. Instead of doing group discussion, the students can also discuss in pairs, as in the informal group learning presented in sub-section 2.4.1.

Thirdly, students are not willing to participate because they do not want to be “embarrassed”. Students might not participate because they are not sure of the answer and fear of giving the wrong answer. They would then be “embarrassed” in front of the class. Another situation is the student does not want to be referred as “teacher’s pet”, as he/she actively participate and fulfil the teaching instructor’s request. It is important not to only incite participation from a few students. General methods, like cold-calling can be implemented. However, more important is to provide a “safe environment”, where no ideas will be ridicule. Instead, students should be praised for their participations. This will encourage students to participate more. This aligns with Skinner’s (1904-1990) behaviour theory. Skinner emphasized that the desired behaviour can be enforced by positive reinforcement like praising and award.

2.5 Pedagogic Theories

As this research concerns discovering suitable teaching methods, 3 pedagogic theories will be introduced in this sub-chapter. The methods implemented will be compared to the pedagogic theories introduced here. This will provide a comparison to see if the methods implemented covers the learning behaviour of different students as well as the level of involvement by each method.

2.5.1 Bloom’s Taxonomy vs. Anderson’s and Krathwohl’s Taxonomy

The Bloom’s taxonomy was defined in 1956 by Benjamin Bloom [Bl56]. This taxonomy categorises the different levels of thinking and learning. This taxonomy has since then been popular among educationist to evaluate the level of student’s ability. Bloom’s taxonomy is divided into six categories. The six categories are knowledge, comprehension, application, analysis, synthesis, and evaluation. The six categories are often represented in a pyramid form, from the simplest and most implemented category to the most challenging and least implemented category (Figure 2.2). Knowledge is at the most bottom level, according to Bloom almost 95% from the exam questions he surveyed are in this category. Students are expected to recall previously learnt material from memory. This can be achieved simply by rote learning. The challenge to test the different categories gets more difficult with each level. The last category is evaluation. Students are expected to judge, criticise, decide the suitability of a certain material.

Evaluation	The ability to assess/compare different information
Synthesis	The ability to use the information to form something new
Analysis	The ability to organise the information as to understand it better
Application	The ability to use the information in a new setting.
Comprehension	The ability to summarise, explain the information in own's word
Knowledge	The ability to recall information learnt

Figure 2.2 - Bloom's taxonomy

In 2000, [AK+00] proposed a modified version of Bloom's taxonomy. This new taxonomy is also known as the "Anderson and Krathwohl's Taxonomy" (AK's taxonomy). The difference between this taxonomy and Bloom's are:

- Instead of having synthesis at the fifth level, evaluation is moved to the fifth level and synthesis in at the sixth level.
- Instead of using noun, verb is used. The new levels are remembering, understanding, applying, analysing, evaluating and creating.

The reason for [AK+00] to switch the evaluating category with synthesis category is because putting the ideas to create something novel or new is more difficult. This change is significant as one needs to first evaluate the pro and cons, and to understand the strength and weaknesses of an idea before being able to come up with something new. This dissertation will use the new categories provided by [AK+00].

2.5.2 Gardner's Theory of Multiple Intelligence

Everyone has a different learning behaviour. Howard Gardner mentioned that everyone has different intelligences at varying degree. The intelligences are linguistic intelligence, logical-mathematical intelligence, spatial intelligence, bodily-kinaesthetic intelligence, naturalistic intelligence, musical intelligence, interpersonal intelligence, and intrapersonal intelligence. Each person will have stronger tendency in one or a few intelligences. Different methods implemented in the class will be able to benefit the students with this particular intelligence. Students with different multiple intelligence usually end up in different fields. For example, students who have stronger logical-mathematical intelligence might end up doing work related to scientific thinking, whereas a student with interpersonal intelligence will further work in area where skills to work effectively with others is required [Ga93]. This does not mean that multiple intelligences are only important for secondary school students who are determining which career steps they are going to choose. As each person has more than one multiple intelligences, teaching instructors in higher education can also make sure that the methods implemented cover at least a range of multiple intelligences. This would be especially useful if the students are from a diverse range of background [Ke01].

2.5.3 Edgar Dale’s Cone of Learning

The cone of learning theory is introduced by Edgar Dale [Da69]. Edgar Dale mentioned that people will remember better when they are actively involved in the learning process. According to the “cone of learning” theory people remember 10% of what they read, 20% of what they hear, 30% of what they see, and 50% of what they see and hear. These first 50% can be acquired by passive learning. The students do not have to be involved. The next 50% can be acquired using active learning. When the students participate by writing or verbally expressing the ideas, they will remember 70% of what they learn. By doing what they learnt, for example performing the simulation, doing the design work themselves, the students are able to remember 90% of what they learn.

When comparing to the AK’s taxonomy, the passive learning only covers the first level (section 2.5.1). The active learning activities require the students not only to receive the information but also to participate in it. This covers then the other 5 levels of AK’s taxonomy, for example by participating in a discussion or giving a presentation, the student needs to summarise and explain the information in his/her own words.

2.6 Embedded System 1 (ES1) and Embedded System 2 (ES2)

Having presented the foundation needed for this research in the previous sub-chapters, this sub-chapter will introduce the two courses in this research. Firstly, the context of the course will be presented. This is followed by the background of the students who participate in both these courses.

2.6.1 Course Introduction

Embedded System 1 with course code FB16-6951 (ES1) and Embedded System 2 with course code FB16-6952 (ES2) were introduced in summer semester 2006 (SS2006)[12] and winter semester 2006/07 (WS0607)[13] respectively. Figure 2.3 shows the semesters where ES1 and ES2 are conducted.

SS2006	WS0607		SS2007	WS0708	SS2008	WS0809	SS2009
ES1	ES2	ES1	ES2	ES1	ES2	ES1	ES2

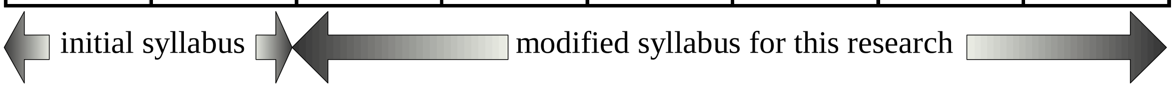


Figure 2.3 – Semesters where ES1 and ES2 are conducted

[12] Summer semester is represented by SS, followed by the year (e.g. 2006, 2007, 2008, 2009)

[13] Winter semester is represented by WS, followed by the short form of the years (e.g. 0607 for 2006/2007)

The course content of ES1 WS0607 differs from that of ES1 SS2006. The same applies to ES2 SS2007; the course content also differs to that of ES2 WS0607. The course content is modified to suit the student’s pre-knowledge and learning behaviour after the first implementation ES1 SS2006 and ES2 WS0607. Except for WS0607 where ES1 is being offered parallel to ES2, ES1 and ES2 are only offered once a year. ES1 is conducted in the winter semester and ES2 in the summer semester. Both these courses are conducted in German by teaching instructor who is a professor, and a teaching assistant who is a doctorate candidate. Both these courses are 3 credits course. 1 credit can be equivalent to 30 minutes of class a week. Each class session lasts for 90 minutes, starting from 8:15 am to 9:45am. The lecture and exercise are conducted in almost alternate weeks. The exercise session will take place after the completion of a chapter. The winter semesters are normally between 16 and 18 weeks. However, the winter semester holidays are between 2 and 3 weeks. The first week is normally lecture-free week. During this week introduction activities for the first semester students will be conducted. Therefore, there are 14 weeks available for lectures and exercises. Summer semesters are shorter and there is no semester holiday in summer semesters. There are only 14 weeks in summer semester and all the 14 weeks are available for lectures and exercises. Generally, there are 14 weeks available for lectures and exercises for both winter and summer semesters. The average number of participants for ES1 and ES2 are 80 students. As there is no official course registration list, the number of the students is based on the exam registration. Averagely, 66% of the students are Computer Science and the other 34% are Mechatronics students. The course handbooks for both Computer Science and Mechatronics students propose the students to take ES1 in the third semester and ES2 in the fourth semester [FB16+06], [SK07].

2.6.2 Subjects Taught in ES1 and ES2

Even though a major part of ES1 and ES2 concerns the computer science discipline but a fraction of physical systems and the basic of electronics are also included. The subjects covered in ES1 and ES2 are listed in Table 2.5.

Table 2.5 - Subjects taught in ES1 and ES2

Embedded System 1 (WS08099)	Embedded System 2 (SS2009)
1) Definitions of terms	1) Requirements Engineering for Embedded System
2) Basics – Logic and Gates	2) Modelling for Embedded System – SA/RT, SysML
3) Typical Architecture	3) Automation Technology
4) Scheduling	4) Programming in Embedded System – IEC 61131-3
5) Programming Languages – PEARL, Assembly Language, and VHDL	5) Verification, Validation and Test
6) Bus System	

Not all the subjects in Table 2.1 are covered, but to the best possible the foundations required to understand embedded system are being covered. As ES1 and ES2 are only 3 credits courses, it is important to only teach the fundamentals of embedded system in ES1 and ES2. There are optional subjects on embedded system that the students can select in higher semesters, if they are interested. The objectives of ES1 and ES2 are to train the students to provide the foundation to think independently as well as to work in team when developing and implementing an embedded system.

ES1 provides the foundation needed for the students to understand the basics of an embedded system. This includes learning about the basic architecture of an embedded system, its development history, example of applications, bus systems, and different programming languages. Microprocessors and microcontrollers are also introduced. Except for logics and gates, the subjects taught in ES1 and ES2 can be identified in the categories as defined in Table 2.1. Logics and gates, which are the basics of all computer systems, are taught to the Computer Science students in the second semester through the course Digital Technology. However, as the Mechatronics students, who also participate in ES1, only take the class Digital Technology in the same semester. This causes the difference of basic knowledge between Computer Science students and Mechatronics students. To solve this problem, ES1 briefly touches this subject as to provide the basic knowledge required to the Mechatronics students. At the end of ES1, students should understand

- the basic components of an embedded system,
- the important requirements like real-time and multitasking ability; different methods to schedule the processes, and different operating systems that can meet these requirements;
- the whole process of reading an input signal, having the signal transfer to a bus system, the processes in the controller, sending the signal back through the bus system and the signal will activate and action in the actuator.

ES2 focuses on the second category of an embedded system course that is the development lifecycle of an embedded system (section 2.2.3). ES2 begins with requirements engineering, then modelling methods, the available automation technology (more to hardware and architecture), the programming language, and finally, verification, validation and test. At the end of the course, the students are expected to know the different lifecycle phases, what happens in each phase, the tools available to assist them especially in the design and implementation phase and the stake holders involved.

2.6.3 Students' Background

ES1 and ES2 are compulsory courses for Computer Science and Mechatronics students who are in their third and fourth semester. There are also a small percentage of Electrical Engineering students but the number is very small (between 0% and 4% for each semester).

This research will only cover the Computer Science and Mechatronics students, as the number of electrical engineering students is not sufficient to make meaningful comparison.

Intake Requirements:

The pre-requisite for Computer Science degree in the University of Kassel is the general diploma for secondary school, technical college or vocational school. The yearly intake is in winter semester. The course will last seven semesters and the students will graduate with a Bachelor of Science degree. The students are required to participate in a 12 weeks industrial practical work. There is no restriction of students' intake.

The Mechatronics students also have the same pre-requisite as the Computer Science semester. The students are expected to graduate within seven semesters and they will be awarded the title Bachelor of Science degree for the Computer Science students and Diplom I for the Mechatronics students. The Mechatronics students have still yet to convert to the Bachelor/Master curriculum. The enrolment is in winter semester. Unlike the Computer Science students, the Mechatronics students are expected to do 15 weeks of industrial practical work. There is also no restriction of students' intake.

From the intake requirements, it is observed that the requirements for the Computer Science and the Mechatronics students are the same. The only difference between the two bachelor degree programs is the duration of industrial practical work. Any student who qualifies for Computer Science also qualifies for Mechatronics and vice versa. The qualifications of enrolled students for Computer Science and Mechatronics courses for winter semester 2006/2007 to winter semester 2008/2009 are acquired from the University Kassel's Students' Centre. The main qualifications for both groups of students are the secondary school (Gymnasium – allgemein Hochschulreife) and technical school (Fachoberschule – Fachoberschulreife). 25% and 48% of Computer Sciences students has secondary school qualification and technical school qualification respectively. More Mechatronics students are with secondary school qualification at 40% and 33% are with technical school qualification.

The Computer Science degree is more mature and better offered as compared to the Mechatronics degree. According to Centre for Higher Education Development (CHE) [CHE] report in "Zeit" [StZeit], the search result for Information Technology / Computer Science (Informatik) in University returns 550 results, whereas Mechanical Engineering returns 102 results. Note that the search did not use Mechatronics, as it is not a search category by itself. From the fact that Mechatronics courses are not commonly offered, it is possible to deduce that the students who choose Mechatronics degree are really interested to develop in this field. Most of the Mechatronics students have secondary school qualification. They may not have practical experiences on Mechatronics, as compared to those who came from technical school or vocational school, but are determined to choose this field. This in return might influence the attitudes the students have towards their study.

Graduation Requirements:

The Computer Science students are expected to complete at least 180 credits and maximum 240 credits for the Bachelor degree. The students are expected to complete 129 credits in the foundation years and another 57 credits in the advance years. Additionally, the students are required to complete another 12 credits for practical and another 12 credits for bachelor project [Le04].

The Mechatronics students need to complete 210 credits for their Diplom I, 30 credits are allocated for each semester. The 210 credits also involve 15 credits for practical work and another 15 credits for bachelor project [SK07]. As Mechatronics Diplom is an interdisciplinary degree, the students will be visiting courses in the field of Computer Science, Mechanical Engineering and Electrical Engineering. The proposed semesters to finish the Bachelor and Diplom I degree are 7 semesters. However, the students can extend the semesters as long as they do not fail any compulsory exams more than 3 times.

The grading for Computer Science students and the Mechatronics students are the same. The grades are categorized into five categories – very good, good, satisfactory, sufficient, and fail (Table 2.6). In order to graduate, the students should at least score sufficient, which means not good but sufficient to pass, for all the courses in the foundation level.

Table 2.6 - Grades for the Computer Science and Mechatronics students

Grade Range	Possible Grades	Remarks
1.0 < x <= 1.5	1.0, 1.3	very good
1.5 < x <= 2.5	1.7, 2.0, 2.3	good
2.5 < x <= 3.5	2.7, 3.0, 3.3	satisfactory
3.5 < x <= 4.0	3.7, 4.0	sufficient
x > 4.0	5.0	failed

Courses Visited:

The Mechatronics degree's syllabus in University of Kassel consists of courses from Faculty of Mechanical Engineering (FB15) and the Faculty of Electrical Engineering and Computer Science (FB16). Starting from the first year, the students will be taking compulsory courses from the Mechanical Engineering, Computer Science and the Mechanical Engineering disciplines [SK07].

The Computer Science students do not need to visit courses from Faculty of Mechanical Engineering (FB15). The compulsory courses for both disciplines are as in Table 2.7. The Software Tools course that will be further elaborated in chapter 4.4 is compulsory for the Mechatronics students in the fourth semester but optional for the Computer Science students. Computer Science students and Mechatronics students do share a few same courses, for example The Basics of Electronics 1 and 2, Mathematics 1 and 2 and Embedded System 1 and 2.

Table 2.7 – Courses covered by Computer Science and Mechatronics students in the first four semesters

Sem.	Courses for Computer Science	Credits	Courses for Mechatronics	Credits
Semester 1	Digital Technology	4.0	Construction Technology 1	6.0
	Introduction to C	3.0	Introduction to C++	6.0
	Introduction to Programming for Computer Science	6.0	Introduction to Mechatronics	1.0
	Law and Management	3.0	Mathematics 1	9.0
	Mathematics 1	9.0	The Basics of Electrotechnics 1	4.0
	The Basics of Electrotechnics 1	4.0		
	Total Credits	***29.0***	***Total Credits***	***26.0***
Semester 2	Algorithm and Data Structure	6.0	Constructions Technology 2	6.0
	Computer Architecture	6.0	Mathematics 2	9.0
	Discrete Structure 1	3.0	Physic	8.0
	Law and Management	3.0	Practical in Electrotechnics	2.0
	Mathematics 2	9.0	Technical Mechanics 1	5.0
	The Basics of Electrotechnics 2	4.0	The Basics of Electrotechnics 2	4.0
	Total Credits	***31.0***	***Total Credits***	***34.0***
Semester 3	Discrete Structure 2	3.0	Digital Technology	4.0
	The Fundamentals of Electrotechnics	3.0	Electrical Engineering's Material	4.0
	Embedded System 1	3.0	Electrical Measuring Technology	6.0
	Operating System	6.0	Embedded System 1	3.0
	Computer Networks	6.0	Manufacturing Technology	3.0
	Theoretical Computer Science - Logic	6.0	Mathematics 3	4.0
			Presentation	2.0
			Technical Mechanics 2	4.0
	Total Credits	***27.0***	***Total Credits***	***30.0***
Semester 4	Embedded System 2	3.0	Dynamics	4.0
	*Application Course	6.0	Embedded System 2	3.0
	Database 1	6.0	Management for Engineers	3.0
	Methodology to Programming	6.0	Mechanical Engineering's Material	3.0
	System Programming	4.0	Sensorics 1	4.0
	Theoretical Computer Science (Formal Language)	6.0	Software Tools	3.0
			System's Model Development	4.0
			The Basics of Control Technology	6.0
	Total Credits	***31.0***	***Total Credits***	***30.0***

Different Cognitive Mind Set between Computer Science and Mechatronics students

According to [GH01], Mechatronics students prefer to have interactive mode of communication. This means that students prefer to be part of the action instead of sitting and listening as compared to the traditional teaching and learning methods. According to the research by [EB03] many Mechatronics courses are based on course works. The students not only develop the control software but they also make sure that the software is compatible with the physical system. Computer Science courses require less practical works with physical systems. Most of the assignments for the computer science students on the other hand can be accomplished using particular software installed in a normal personal computer.

2.7 Overview on Scope of Research

This research focuses on discovering the suitable methods for a large interdisciplinary class. The courses in this research are ES1 and ES2. Both these courses have interdisciplinary course content. Sub-chapter 2.3 presented the three challenges faced by institution of higher education. This research covers the intersection of all the three challenges (Figure 2.4).

Area of Research: The participants in ES1 and ES2 are Computer Science students and Mechatronics students. From the courses visited by Computer Science and Mechatronics students in Table 2.7, it can be observed that the focus is different and thus developing a different set of cognitive mind set. The Computer Science students are at advantage because the Computer Science students covered part of the subjects in ES1 and ES2, which are Digital Technology, Computer Architecture, Operating System and Computer Network, in the first three semesters. Every semester there are about 80 participants for both these courses. From the class size definitions in Table 2.2, ES1 and ES2 fall into the large class category. Different methods to involve students in a course are grouped under the term active learning. Active learning lecture is popularly implemented for large class size. One of the challenges in implementing active learning is the limitation of resources.

Methods implemented: Section 2.4.1 categorises active learning methods to “in class implementation” and “out of class implementation”. The methods that are purely “in class implementation” and are connected to presentation methods will be grouped under the term “motivating lecture”. Another four active learning methods are group work, exercise session, pop quizzes, and course coupling. Using bench lab equipment as a consistent example is the sixth method implemented in this research. Each of this method can tackle one or more challenges as shown in Figure 2.4. However, the implementation needs to consider all the three challenges at the same time.

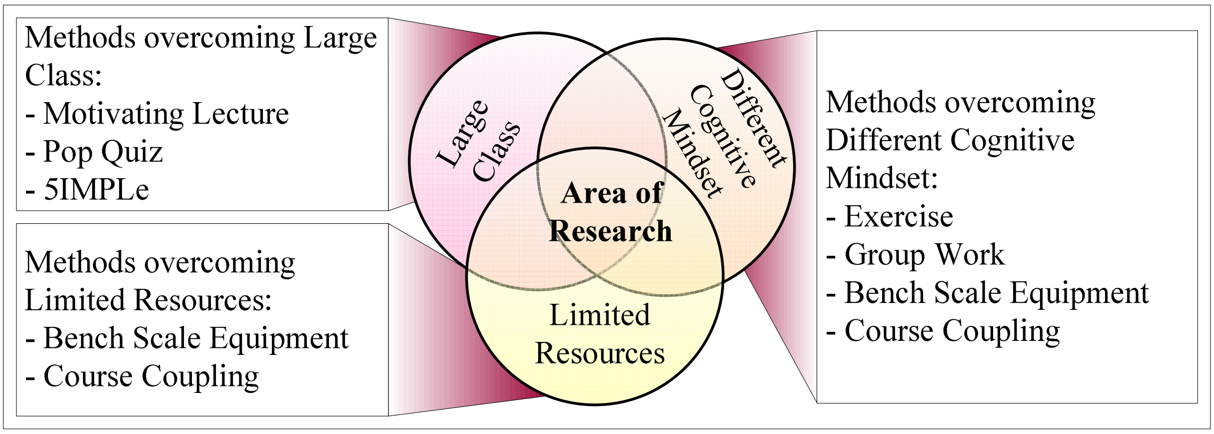


Figure 2.4 – Area of research and the methods implemented

The resources required for implementation, the comparison of methods to pedagogic theories in sub-chapter 2.5, as well as the effectiveness of the various methods to the different disciplines will also be discussed.

3 Course Content for Embedded System 1 and 2

As being introduced in sub-chapter 2.2 Embedded System is an interdisciplinary course. There are many subjects that can be covered in this course. Through out the years, the course's contents for ES1 and ES2 have been slightly modified. This chapter presents the concepts taught to the students. Only the subjects that are consistently taught through out the semesters will be discussed here. These are also the subjects taught in ES1 WS0809 and ES2 SS2009. The first sub-chapter will present the course content for ES1 and the second sub-chapter the course content for ES2. Each chapter in ES1 and ES2 will further be divided to subjects.

3.1 Course Content for Embedded System 1 (ES1)

ES1 is conducted in the winter semester. As mentioned in chapter 2.6.1, there are 14 weeks for lectures and exercises in the semester. This is followed by almost two months free lecture period, where the students need to prepare for the exam. The following sections will describe the subjects covered in ES1, namely definition of terms; basics – logics and gates; typical architecture; scheduling; programming languages – PEARL, assembly language, VHDL; and bus system.

3.1.1 Definition of Terms

During the first class, the teaching instructor first introduced the different terms in embedded system. This is important to lay the same understanding for all the students before further developing the subject.

A few definitions are discussed here. The first term is a system, the second an embedded system (eingebettete Systeme), the third a system incorporated with embedded system (einbettende Systeme), the fourth system incorporated with embedded system as a product, and the fifth term a system incorporated with embedded system as a production system.

- The definition for system is based on DIN 66201 [DIN66201]. A system is a structure that consists of different objects. These different objects have unique characteristics and specific relationship with one another. Each system comprises of hardware, software and user. Each has its own specific interfaces.
- The term embedded system is as defined in sub-chapter 2.1. An embedded system is an embedded hardware/software system that regulates a physical device by sending control signals to actuators in reaction to input signals provided by its users and by sensors capturing the relevant state parameters of the systems. An embedded system needs to meet its real time requirements. Real time system needs to execute its processes within the predefined time limit [DIN44300].
- The third term, system incorporated with embedded system is defined as a system that requires and embedded system to function. The real time requirements for the

embedded systems depend on the system that incorporates them. An example is the automobile. A car can have different embedded system, for example a navigation system, auto braking system etc. System incorporated with embedded system can be further categorised to a product or a production system.

- The fourth term is system incorporated with embedded as a product. Examples of this system are health care system, automobile with electronic system, aviation electronics etc. These products are incorporated with embedded system to perform the control functions. The users have limited influence on the system. They are only able to use the functions as provided by the system. These systems normally run on programmable hardware, microcontroller, and microprocessor. The common programming languages used here are C, C++ and Assembly language.
- The fifth term is system incorporated with embedded system as a production system. Examples of this system are manufacturing system, process technology system, logistic system etc. These systems are used to produce a certain product. These systems normally run on Programmable Logic Controllers (PLCs), Distributed Control System (DCS) or Industrial-PCs (IPCs). The programming language used here are normally based on IEC-61131-3 standard, namely Ladder Diagram (LD), Functional Block Diagram (FBD), Sequential Function Chart (SFC), Instruction List (IL), and Structured Text (ST).

The course ES1 focuses on the fundamentals of an embedded system with a view on the product incorporated with embedded system. The subjects covered here include the common architecture, programming languages, and bus system for microprocessor and micro controllers. ES2 on the other hand focuses on production system with incorporated embedded system. The focus here is the development process of such a system, starting from requirements engineering to test and validation of such system.

3.1.2 Basics – Logic and Gates

The next chapter is logics and gates. Logics and gates are the basics of a microprocessor and microcontroller. The Artist Education Group commissioned by the European Commission mentioned that *“basic notions on logic gates, combinational and sequential circuits“* should be part of the foundations of computer science and engineering for a graduate students [CS+05].

Firstly, the students are briefed on the generation of digital signals. The students are introduced with Shannon-Nyquist sampling theory. Shannon-Nyquist sampling theory states that in order to reconstruct an analogue signal in digital format, the equally spaced sampling rate should be twice the highest frequency of the analogue signal.

This then followed by the logical functions of AND, OR, XOR, NAND, NOT etc. Different Boolean algebra axioms for conjunction and disjunction functions are also introduced. Examples of the axioms are associative, commutative, distributive, and De Morgen

Theorem. Next, students are taught to simplify the Boolean expression. The algebraic as well as the Karnaugh-Veith diagram method are introduced.

The students are also introduced to sequential circuit and combinatorial circuit. Examples of sequential circuit are flip-flops and register, whereas examples for combinatorial circuit are full adder and half adder. A combinatorial circuit is a sequential circuit with delay and feedback circuit. Among the common flip-flops introduced to the students are SR flip-flop, JK flip-flop, T flip-flop, D flip-flop and master-slave flip flop. Finally, examples of flip flops' implementation using a traffic light example is presented. Using this example, the students start with the truth table based on JK flip-flop, followed by the simplification using Karnaugh-Veith diagram and the drawing of the circuit diagram.

3.1.3 Typical Architecture

The first subject in the chapter is modelling using Petrinet [Gr88]. Petrinet is being introduced because it can be used to model the behaviour of an operating system, a communication system, the reading and writing process using one channel, and various other concepts. Only the basics of Petrinet, namely the different elements in a Petrinet, the vectors and the matrixes, and the different connections between place and transitions are introduced to the students. Using the places and transition, the conditions and the flow of an operating system is modelled.

The second subject introduced is the Flynn Notation. There are different ways of handling instruction and data stream. Flynn notation is introduced to describe how a processor handles the instruction and the data stream. Flynn classifies it in four categories, namely single instruction single data (SISD), single instruction multiple data (SIMD), multiple instruction multiple data (MIMD), and multiple instruction single data (MISD).

The third subject covered is the distinction between a microprocessor, microcomputer and microcomputer system [WB05]. A microprocessor incorporates an arithmetic logical unit (ALU) and a control unit. A microcomputer has one or more microprocessors as its central processing unit, memory and input-output interface. A microcomputer system contains a microcomputer and the peripheral devices. This is followed by introducing architecture for the microcomputer system, the von-Neumann architecture. The von-Neumann architecture has a data bus for both the instruction and data. The students are taught the different component of a microprocessor. This provides a basic understanding on how a microprocessor or microcontroller works. This knowledge is necessary if the students are interested to study deeper on the development of microprocessor or microcontroller.

The fourth area covered is timer. The watch dog time is introduced. Executing software will reset the timer to its initial value. If the timer is not reset and is decremented to zero, then an alarm will be released.

The last subject discussed is interrupt and polling. There are two main categories of interrupts, namely hardware interrupts and software interrupts. Interrupt process in microprocessor are presented, for example Motorola 680xx [Ha03] has 7 levels of interrupt priorities whereas Intel 8259 on the other hand has 8 interrupt request lines. Interrupt will stop the current task to complete the higher priority request. When the higher priority request is completed, then the previous task will be continued. Polling on the other hand will continuously ask the different devices if there is any task to be done. It is difficult to decide when to implement interrupt or polling. However, typically interrupt can be implemented if it happens infrequently, the change is time critical, whereas polling should be implemented when no precise timing is necessary and the impulses are long.

3.1.4 Scheduling

Two subjects are introduced in this chapter. The first is real-time operating system and the second is the different scheduling policy [Bu97]. Real-time computing is the core for embedded system. Unlike normal computer software, software for embedded system has real time requirements. The real-time requirements are satisfied using various scheduling policies that will be introduced here.

Firstly, the requirements for real-time operating system are presented. A real-time operating system needs to correctly fulfil the logical requirements and the time requirements (timeliness).

The properties of a real-time operating system and a normal operating system were also presented. The definition of an operating system is based on DIN 44300 [DIN44300]. The operating system is defined as "The program of a digital computing that controls and monitors the execution of programs. The properties and the computing system, which is the basic of possible computing modes, are taken into consideration." A real-time operating system needs to fulfil both timeliness (Rechtzeitigkeit) and multi tasking (Gleichzeitigkeit).

"Die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften dieser Rechenanlage die Basis der möglichen Betriebsarten des digitalen Rechensystems bilden und die insbesondere die Abwicklung von Programmen steuern und überwachen." [DIN44300]

The second subject covered is the different scheduling methods. Here pre-emptive scheduling and non pre-emptive scheduling are introduced. Using a non pre-emptive scheduler, the system needs to complete the tasks before starting a new one, whereas a pre-emptive scheduler can be pre-empted any time by a higher priority task. Next, semaphore is introduced as a synchronisation's method. Semaphore is a protected variable to access shared resources. The concept is demonstrated using Petrinet that is being introduced in chapter "Typical Architecture". An example where two trains wanting to access the same railway track is presented using Petrinet. The second train can only use the railway track after the first train "leave" the track.

3.1.5 Programming Languages

Many embedded system is real-time critical. Real-time operating system is needed that ensure resource access, scheduling and sharing is introduced in 3.1.4. Here, the programming languages that are suitable for embedded system will be introduced. The chapter starts off with introducing the evolution of programming languages. There are real-time programming languages, object-oriented programming languages, and programming languages with real time abilities when implemented on a real time operating system. The three languages introduced here are languages with real-time feature, namely PEARL, and language to program a microprocessor or microcontroller, namely Assembly Language and VHDL.

PEARL:

The first programming language is PEARL [HV05]. PEARL stands for "Process Experiment Automation Real-time Language". It is based on Pascal programming language. A PEARL program can be divided into modules. Each module can be further divided to system part and problem part. System part covers the declaration with input-output interface, whereas the problem part includes data, tasks and procedures. Among the areas covered here includes the structure of a module, declaring the input and output interface, the different data types, and commands. PEARL provides different commands that enable real-time operation like ACTIVATE, SUSPEND, CONTINUE, AFTER, DURING UNTIL, ALL etc. These commands are used ensure the real-time operations are fulfilled.

The students also implemented semaphore in PEARL. The example here is built upon the railway track problem discussed in the previous chapter. In order to provide better understanding, this example is implemented using an emulator. The students can download RTOS-UH emulator [RTOS-UH] to program in PEARL. The few important commands here are PRESET to preset the variable's value, REQUEST to request for the semaphore, RELEASE to release the semaphore when the task completes and TRY to ask if the semaphore variable is available.

Finally, the students implemented the BOLT-variable. BOLT is introduced after semaphore because the BOLT-variable also allows only a certain number of processes to access the variable. This helps the student to build their knowledge on BOLT upon what they learnt for semaphore. Just like semaphore, this is determined by the keyword PRESET. Similar to the REQUEST and RELEASE commands for semaphore, BOLT uses ENTER and LEAVE. However, BOLT has to extra command and that is RESERVE and FREE. RESERVE command has higher priority than ENTER. RESERVE command will block other processes from accessing this variable. This variable is used in real-time process to write block other processes from using a value when it is updated. When the writing process completes, then the FREE command will be used to free this variable.

Assembly Language:

The second programming language in ES1 is the assembly language. Even though other high level languages, for example C, are being used to program microcontroller, assembly language is the earliest language used. As different microcontroller will have its own instruction sets, the purpose here is to give students a “feeling” on how to use assembly language. Therefore, only the basics of assembly language are being introduced here. The commands are based on Motorola 68000 architecture [Ha03]. The first area covered is register addressing, namely direct addressing, constant addressing, absolute addressing, and indirect addressing. These operations can be conducted for Byte, Word or Long Word. Examples for the different addressing methods are as below:

- direct addressing: MOVE.W D1, D3 (Move 2 Bytes content from data register D1 to D3)
- constant addressing: MOVE.W #$1234, D2 (The constant hexadecimal number 1234 should be moved to D2)
- absolute addressing: MOVE.W $1234, D2 (Move the content from the address 1234H to D2)
- indirect addressing: MOVE.W (A0),D0 (move the content from the address in the address register to D0)
- displacement addressing : MOVE.W D3, $1234(A2) (The data in the address 1234H will be added to the data in A2, and will be stored in D3)
- indirect addressing post increment: MOVE.B (A0)+, D0 (After the operation, in this case data in memory address in A0 is moved to D0, the address in A0 will be increment 1)
- indirect addressing pre decrement: MOVE.B D0, -(A0) (Before the operation, the content in A0 will be decremented by 1)

Lastly, different types of status register are introduced. Examples of status register are carry flag (C), overflow flag (V), zero flag (Z), negative flag (N), and extend flag (X).

VHDL:

The third programming language is Very high speed integrated circuit Hardware Description Language (VHDL) [Re09]. VHDL is influenced by ADA. VHDL allows one to describe a digital system at different abstract level, for example the structural and the behavioural level. This is then followed by synthesising where synthesis tools will translate the design into real hardware. This topic is introduced to provide the basics needed for hardware programming. Another aspect is to let students know that the logics and gates learnt in the beginning of the course are useful for various applications. The students did not implement synthesising with VHDL. Only the basic of VHDL code is introduced in ES1.

The VHDL code consists of minimum an entity and architecture. An entity describes the structure by defining the interfaces. Through the keyword port, the input, output and input-output variables are declared. The architecture describes behaviour through the implementation of the program. Architecture always belongs to an entity. The architecture consists of declaration part and definition part. Declaration part declares the data types, constants, components, signals etc. A constant can be assigned a value but this value cannot be changed during the program run time. A variable is also assigned a value but this value can be modified during run-time without any delay. Signal is a type of variable where the actualisation of value change can be delayed. Components are declared entity that is being implemented in an entity. Components have the same interfaces as its entity. Definition part starts after the word "begin", examples of operations are signal assignments, processes, and component instantiations.

3.1.6 Bus System

Two important characteristics for data transmission of an embedded system are correctness and deterministic behaviour of data transmission. Through the areas covered in this chapter, the students are given an idea on how these two characteristics can be fulfilled. Among the subjects are the error identification and error correction; synchronous and asynchronous transmission; medium access control, and field bus system [Ta03].

In order to ensure the correctness of data transmission, error identification and correction are introduced. Two error identification methods are introduced here, namely the checking using parity bits and polynomial code checksum (or also known as cyclic redundancy check). Parity bit is a bit that is added to a string of 7 bits. This eight bit determined if the string is an even parity bit or odd parity bit. In order to check the correctness of data transferred, the number of 1s will be compared at the recipient side. However, it is also possible for one error to cancel the other and no errors will be identified. Therefore, cyclic redundancy check is introduced. In cyclic redundancy check, the string is divided by a polynomial. The balance of the division will be joined to the string and be sent to the recipient. At the recipient, the received string will be divided with the same polynomial and if the balance is 0, then the received string is correct. One error correction method is introduced. It is the Hamming-Code. Using the Hamming-Code, the bits with error can be identified and thus corrected. The students are presented with examples on how the correction takes place and the calculation of hamming distance for number of bits to be identified and corrected.

Embedded systems have deterministic behaviour. Therefore, the communication between the devices should also be timely and deterministic. The students are taught two types of real-time communication mechanism. They are carrier sense multiple access (CSMA), collision detection (CD), and collision avoidance (CA). CSMA/CA avoids collision, whereas CSMA/CD detects the collision and then retransmits the data. Therefore, CSMA/CA is more suitable for hard real-time applications, for example the data transmission in a car. CSMA/CA is implemented in the Controller Area Network (CAN), whereas CSMA/CD is implemented in Ethernet.

As mentioned in chapter 2.2.1, an embedded system can be a product or a production system. Fieldbuses are implemented in factory automation (a production system) to minimise cabling effort. Therefore, the students are also given a short introduction in this area. Examples on how fieldbus can be connected to controllers, groups of input and outputs terminals, or directly to the sensors and actuators, and the organisation of cables using backplanes (Klemmkasten) are presented. Next, different standards for fieldbus communications are introduced. They are the RS-485 for PROFIBUS, Interbus etc; RS-232 as interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE); and the IEEE-488 general purpose inter face bus (GPIB) for short range (within 20 meters) communication.

3.2 Course Content for Embedded System 2 (ES2)

The subjects in ES2 focus on the production system incorporated with embedded system. The chapters in ES2 are requirements engineering for embedded system, modelling for embedded system, automation technology, programming in embedded system, and verification, validation and test.

3.2.1 Requirements Engineering for Embedded System

There are a few areas covered in this chapter, for example methods to collect requirements, lifecycle model of a system, and agile development.

Firstly, there are different methods that can be implemented to collect ideas. Two methods are introduced here, namely, the group and individual method. Students are highlighted with the strength and weakness for each method and the impacts of wrong requirements. Poorly defined requirements will give the system developers room to develop a system that does not fulfil the user’s requirements.

Next, lifecycle phases according to V-Model [Be05] are introduced to the students. Through this V-Model the Computer Science students and the Mechatronics students are able to see the interaction between the disciplines (Figure 3.1).

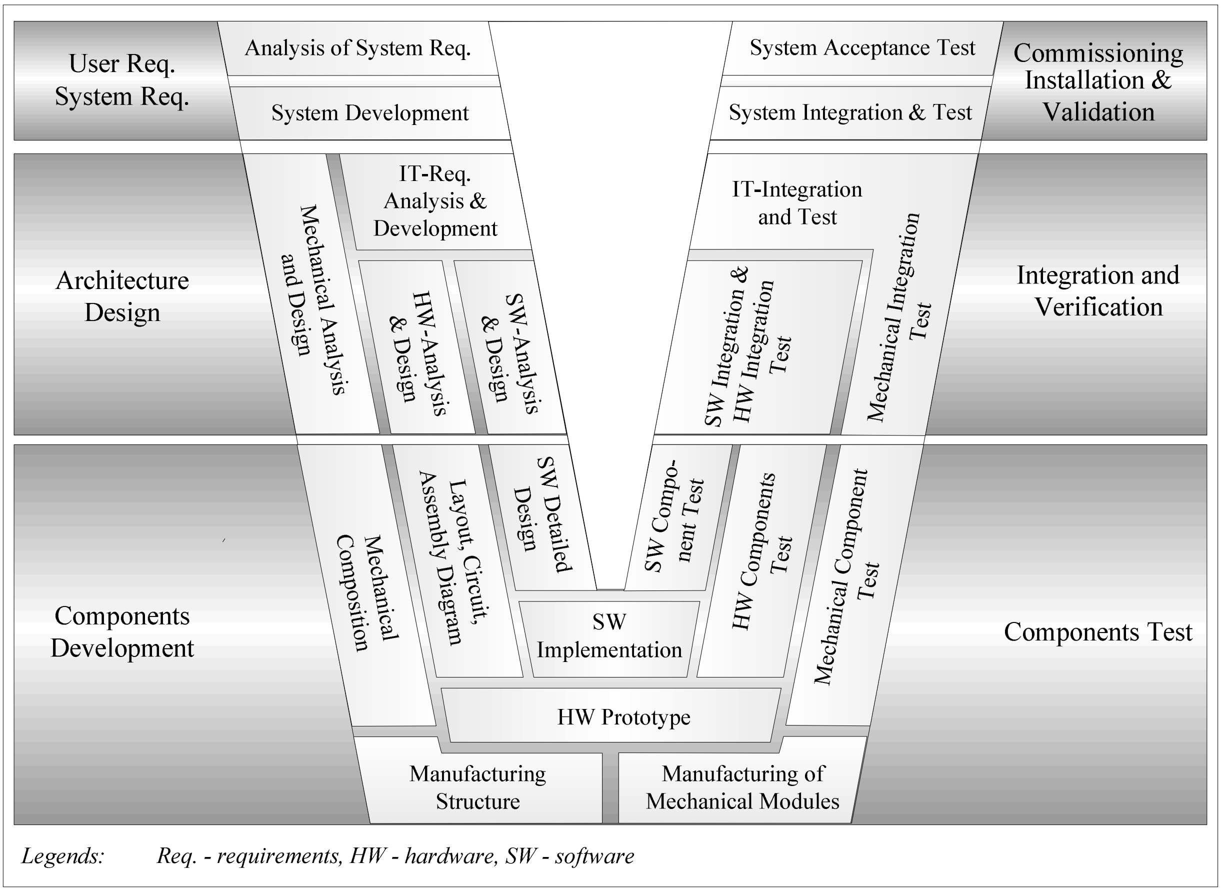


Figure 3.1 – V-model showing interaction between 3 disciplines – mechanical, electrical and software engineering (modified from [Be05])

The course content for ES2 does not cover the development of hardware and electrical components, but the students are taught concerning the interface between these disciplines. In the chapter requirements engineering for embedded system, the areas concerning users' requirements and system requirements are discussed. Architecture design is covered in the chapter modelling for embedded system. Components development in the chapter programming for embedded system using IEC 61131-3. Finally, the chapter verification, validation and test cover components test, verification with the design and validation against the users' requirements.

The first area covered in gathering user requirements is to identify the stakeholders. Examples of stakeholders of a system include management personnel, user, system developer, maintenance personnel, etc. The source, priority, urgency, ownership, requirements, stability of each requirement needs to be documented. Requirements are normally documented in text form. System requirements include identifying the solution systems, the technology that can be implemented, and the information flow of the system. Prototyping is one of the methods to collect both user and system requirements. Using prototyping the users are able to identify if the developer and the user have the same view on the system. Quick changes and additional requirements can be made on the prototype.

Different examples on requirements documents are presented using a hydraulic press example.

3.2.2 Modelling for Embedded System – SA/RT

Structured Analysis with Real-Time requirements (SA/RT) is an evolution from Structured Analysis and Design Techniques, Structured Analysis, and Structured Analysis and System Specification. Petrinet as introduced in ES1 (see 3.1.3) is able to model the behaviour of a system. With this foundation SA/RT is introduced. SA/RT not only can describe the process flow, it is also able to describe the architecture of the systems. SA/RT can be divided to system model and architecture model [HH+00]. System flows consist of process model and control model. The first level or level zero of both models is known as the context diagram. The following levels are known as data flow diagram (DFD) and control flow diagram (CFD) respectively. The architecture flow diagram presents the data and control flow in the system, whereas the architecture interconnect diagram shows the physical connection between the different parts of the system (Figure 3.2).

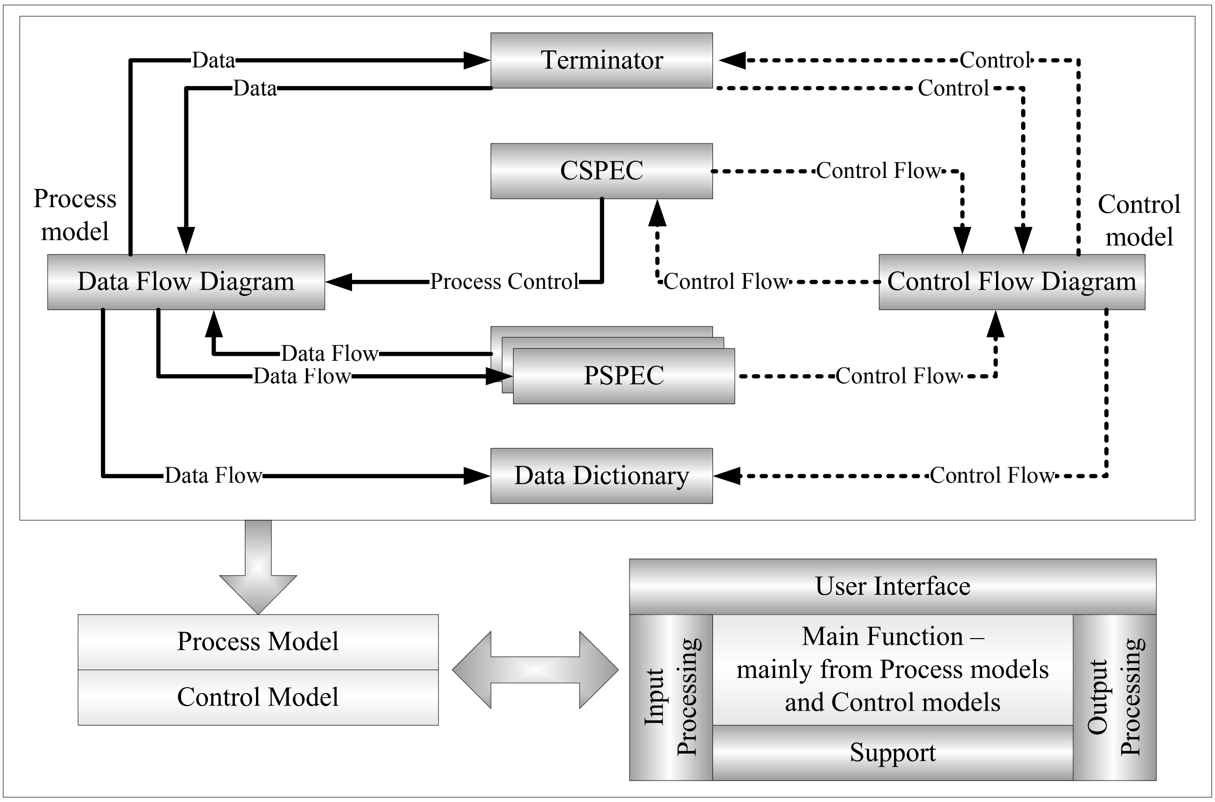


Figure 3.2 – System model and architecture model according to Hatley and Pirbhai [Vo03]

Both process model and control model include terminator, process, storage and (data or control) flow. Terminator is an external element to the system. SA/RT takes note of the external elements that interact with the system through terminators. A terminator can be a user, organisation, a physical machine, another system etc. A process will change the input to another output. Each process has a name and number. Storage stores the information, energy or material. The same storage can be used repeatedly through out the model. (Data- or Control) Flow shows the direction, an information, material or energy flow. Each flow has a name. Both process specifications (PSPEC) and control specifications (CSPEC) are the final refinement for the process model and control model respectively. PSEC and CSPEC can be represented in mathematical formula, text, table, graph, automata etc. The real-time requirements can also be described in this specification. Data dictionary describe all the data that are implemented in the process model and control model.

An architecture flow diagram comprises of architecture module(s), architecture flows and terminator(s). Architecture flow diagram shows the exchange of data between the different modules. It can also be represented in a table from. An architecture interconnect diagram describes the physical connection between the module. Issues like time requirements, compatibility requirements and security need to be considered here. Both architecture models consist of user interface, input processing, output processing, main functions and support. Finally, these concepts are presented using the same hydraulic press that is elaborated in the requirements engineering chapter.

3.2.3 Modelling for Embedded System – SysML

SysML is a general purpose modelling language for system engineering application [SysML]. SysML has 9 different diagrams. They can be divided into three categories, namely requirements diagram, structural diagram and behavioural diagram. There are 4 types of structural diagram and 4 types of behavioural diagram. There is only one diagram to represent the requirements and it is the requirement diagram (Figure 3.3). SysML is selected over UML as SysML caters for the description of a system. It also has two new diagrams, namely “Requirement Diagram” and “Parametric Diagram”.

The structural diagram describes the physical system. The 4 types of diagram are package diagram, block definition diagram, internal block diagram, and parametric diagram. Package diagram is used to organise the elements in the system. Using block diagram and internal block diagram, ES2 introduces the concept of inheritance to the students. This is necessary when considering the possibility of implementing object oriented in embedded system [Vo08]. As an embedded system comprises of both hardware and software, the students are introduced to the concept that a block can represent both hardware and software. Internal block diagram describes the internal structure of an element that can be represented by a block diagram. Through the different ports in the internal block diagram, information or energy that flows from one block to the other can be represented. The parametric diagram can further describe the relation between the different properties in the system block.

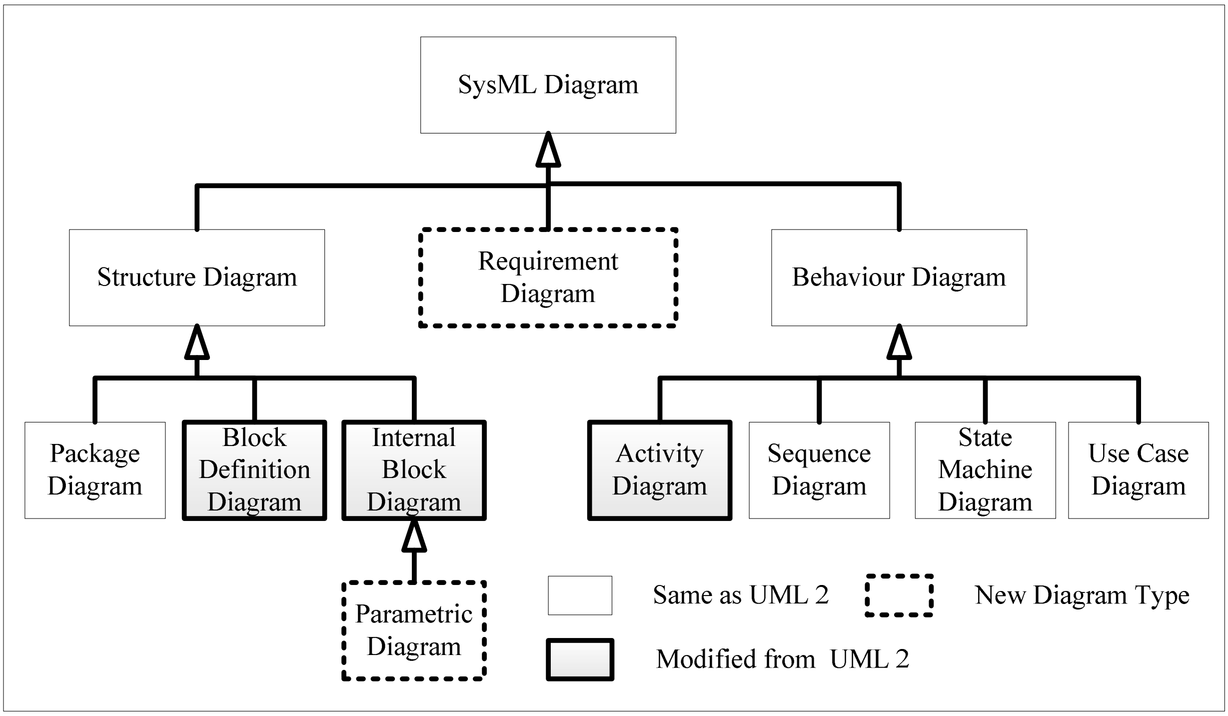


Figure 3.3 – Diagrams in SysML [SysML]

The behavioural diagram describes the behaviour or the processes in a system. They are activity diagram, sequence diagram, state machine diagram, and use case diagram. During the SA/RT classes, the students are taught process and control flow diagrams. Activity diagram has similar function as the process and control flow diagrams as it describes the turning of an activity. Activity diagrams have swim lanes that can allocate the activity to the responsible module. The responsible module can be the system or external elements (terminators in SA/RT). The sequence diagram describes the communication between two or more modules in a sequential manner. The lifelines represent the communication's partners and the messages will be exchanged between these lifelines. Using state machine diagram, one can describe the conditions for a particular event to happen, this is also similar to the state chart as control specification in SA/RT. Use case diagram is an abstraction of the communications between the different modules in a system.

3.2.4 Automation Technology

Having introduced the different methodology to model a system, the students are introduced to the architecture of an embedded system (automation system). In this chapter the students are first introduced with the components of a process automation system. Among the different components are the input-output interfaces, the communication system and the automation computer systems (Figure 3.4). A process signal is captured by a sensor. An analogue signal needs to be converted to a digital signal. It will then be transported across the communication system, for example a bus system. Following, it will be processed in the automation computer system. The automation computer system includes both the hardware of the system and the software that runs on the hardware. After process, a value will be returned to the actuator for the next action. This will go through a bus system and then the digital value will be converted to analogue value. The actuator will then perform the action based on the analogue value.

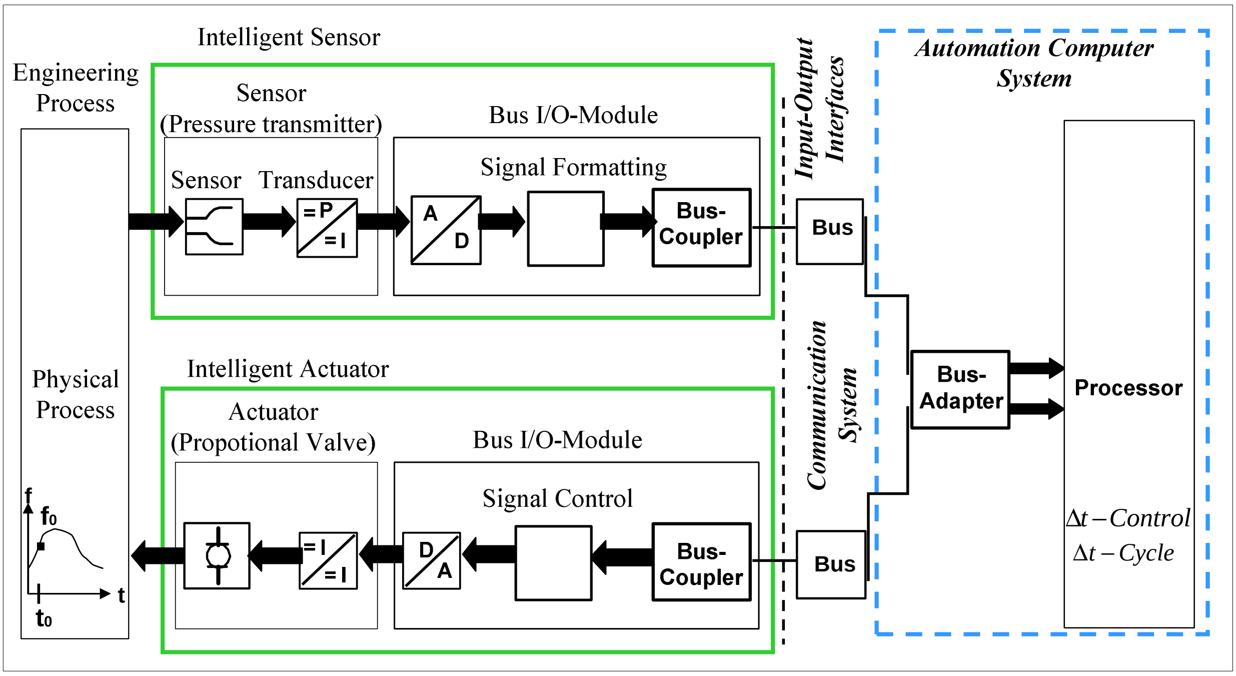


Figure 3.4 – Transmission of process signal between a physical process and an automation computer system [Vo09]

Next, 3 types of automation computer system are introduced. The objective is to enable the students “roughly decide” which automation systems is suitable for the defined requirements. They are distributed control system (DCS), programmable logic controller (PLC) and soft programmable logic controller (soft-PLC).

The control of the DCS system is distributed across different controllers. DCS is typically implemented in process engineering that involves continuous/batch processes. Examples of process engineering industry are refinery, pharmaceutical manufacturing, power grid etc. The architecture and the relevant bus system for DCS are also introduced. The students are also get to know the different DCS systems that are in the market, for example SIMATIC PCS7 from Siemens [PCS7], Delta-V from Emerson Management [DeltaV] and CENTUM from Yokogawa [CENTUM].

The definition of a PLC as according to [DIN61131-1] is “a digital system … with internal memory … is able to execute logical, process sequence, timing, counter, and arithmetical functions and … control different types of machines and processes”

“ein digital arbeitendes elektronisches System … internen Speicherung … Funktionen wie Logik, Ablauf, Zeit, Zählen, und Arithmetik auszuführen und …verschiedenen Typen von Maschinen oder Prozessen zu steuern …“ [DIN61131-1]

PLC is normally implemented for the control of discrete processes. Examples of such processes are control in logistic warehouse, assembly lines etc. The focus in this section is to highlight the cyclic operation of a PLC. A PLC will first read the input, process the signal, and then write to the output. As compared to a real-time operating system, PLC does not “react directly” to event triggered problem. It needs to reach the “input phase” before it can react to the event. The respond time for a PLC is shorter as compared to the DCS. The typical communication system are Ethernet connection [IEEE802.3], PROFIBUS [PROFIBUS], or Coaxial Cable. One can expand the PLC system by adding new rack.

Soft-PLC can be implemented on applications that does not have “hard real-time” requirements. Soft-PLC is usually executed using Industry Personal Computer (Industry-PC) or Embedded Personal Computer (Embedded-PC). The real time requirements are met by using a normal operating system with real time extension. Percentage of CPU resources can be allocated according to the control activities. The advantage of Soft-PLC is being able to run PLC without hardware controller. However, the real time requirements might not be fulfilled all the time.

3.2.5 Programming in Embedded System – IEC 61131-3

Having introduced the sequence processing of a PLC in the previous chapter, the influence of this sequence processing towards the programming language is introduced here. Automation control software is developed using the programming languages mentioned in IEC 61131-3 [DIN61131-3]. The programs are organised in different Program Organisation Units (POUs). The POUs can be programs, function blocks and functions. A program can consist of function block(s) and/or function(s). A function block can further contain other function block(s) and/or function(s), whereas a function can only consist of other function(s). Both function block and program can have many inputs and many outputs. It can be instantiated many times and in between values can be saved. A function cannot save any in-between value and may not use any global variables. A function can have many inputs but it will only return one output.

There are 5 types of programming languages under IEC 61131-3 (Figure 3.5). The programming languages can be grouped into two categories, graphical and textual. The textual languages are Instruction List (IL) and Structured Text (ST). The graphical languages are ladder diagram (LD), and Function Block Diagram (FBD). Sequential function chart (SFC) can be presented in both textual and graphical form. By briefly introducing the 5 languages, the students are able to select which language they are most comfortable with when implementing a PLC, which is introduced in the previous chapter.

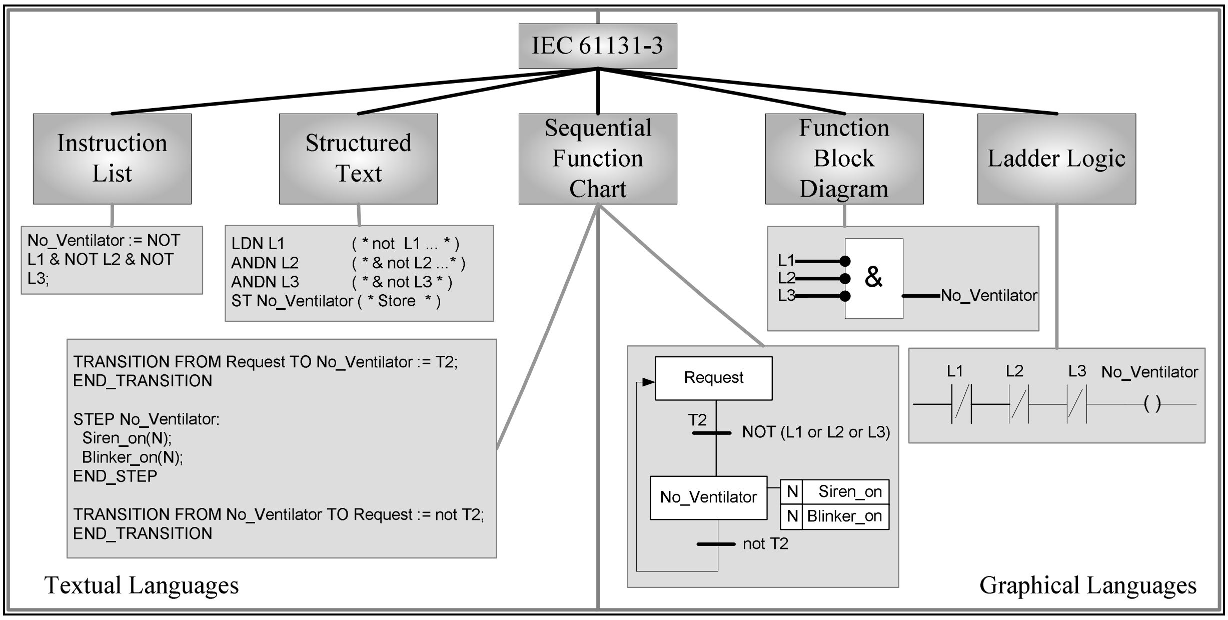


Figure 3.5 – Different programming languages in IEC-61131-3

Instruction List (IL) is similar to assembly language. Every command begins with a new line and is executed one after another. Identification-label can be labelled using a colon “:”. Commands can jump to the different identification-label. Structured Text (ST) is a high level language and is similar to PASCAL. ST is also executed line by line with conditional statements, mathematical calculation, and iteration. ST is more readable as compared to IL. Ladder diagram (LD) looks like an electrical plan. It is read from left to right. Using the “ON” and “OFF” sign, the “TRUE” and “FALSE” value are being assigned. Function blocks can also be assigned to a LD. Function Block Diagram (FBD) is a graphical programming language based on logical blocks and other function blocks. Logic gates taught in ES1 is similar to the implementation of FBD. Sequential Function Chart (SFC) consists of steps and transitions. The concept of state charts in “modelling of embedded system” can be mapped to SFC, as the transition conditions need to be fulfilled before steps are executed. SFC is able to describe the dynamic behaviour of a system. The instructions in SFC can also be represented using text. The subjects covered in previous chapter became the foundation to implement IEC 61131-3.

3.2.6 Verification, Validation and Test

Having covered the development lifecycle of an embedded system, the last chapter covers verification, validation and test. Firstly, the different types of errors are highlighted so that students will know “where to look” for the errors. Errors that may happen in an embedded system can be categorised to physical errors due to the electrical or mechanical part; inherited error due to the errors made in the development or design stage; and non-inherited error for example operational error, maintenance error or vandalism [LG99]. There are different standards for embedded software, for example in the automotive industry the standards are based on the state of the art for this specific product, whereas in areas like medical engineering, aviation, and other dangerous industry, they are controlled by standards like ISO 9000, ISO/IEC 12207 and IEC 51608. Different standards according to IEC 51608 are also introduced [Li02].

Next, the distinction between verification and validation is highlighted. From Balzert’s [Ba09] definition, verification checks if the problem is correctly solved, whereas validation checks if the problem is correctly formulated according to the users’ requirements. It is possible that a problem is correctly solved, meaning the program runs correctly; but it is not according to the users’ requirement. This again highlight the importance of “requirements engineering” covered in the beginning of ES2. Next, two types of tests to verify the systems are introduced. They are static test and dynamic test. Static test are mainly based on analysing the codes, going through the data flow and verifying if the problem are solved correctly using automata techniques. Dynamic test are done by running the software. Two types of tests are introduced here, white box testing and black box testing. White box testing is a structured oriented test, whereas black box testing is function oriented test. For very expensive application, back to back test can be conducted. The specifications are given to two programmers and their program will be compared to see if it is the same.

Embedded system needs to meet its real-time requirements and consists of electrical, mechanical and software parts. Therefore, apart from functionality test, timing test and interface test are also necessary. Timing test evaluates the timing behaviour and synchronisation of data access. Interface test can be divided to human machine interface, input-output interface, system to system interface etc. Didactically, the subjects for one chapter build upon the chapters before. This helps to build the different layer of knowledge required to tackle a new subject.

4 Implementation of Teaching and Learning Methods

Chapter 3 presented the subjects covered in ES1 and ES2. This chapter is going to present the methods implemented to teach these subjects. The methods implemented in ES1 and ES2 can be divided into two categories. They are methods implemented in the class and methods implemented outside the class. Methods implemented in the class are methods conducted during the class session, be it during the lecture or the exercise session. Methods implemented outside the class, are methods that require students to participate apart from class hours. Methods implemented outside the class require the students to meet apart from the class session to complete the assignments given, or to attend another class to help them with the subjects. Apart from methods implemented in the class and methods outside the class, another step taken was to use the same bench scale equipment as example for all the learning pertaining to the course.

Firstly, the implementation of bench scale equipment as consistent example for ES2 will be presented. Next, in the class methods for active learning will be described. This is then followed by active learning implemented partially outside the class, namely group work. The fourth sub chapter will discuss on the method course coupling. These different methods will be reviewed against AK's taxonomy and Gardner's theory of intelligence (sub-chapter 2.5), whenever appropriate. The last sub-chapter will provide a systematic overview on all the methods implemented to the learning pedagogies.

4.1 Bench Scale Equipment as Example

Throughout ES2 exercise, bench scale equipment is introduced. This bench scale equipment belongs to the Department of Embedded System. It is a discrete system. The main function of this system is to stamp the work pieces and sort it accordingly. This bench scale equipment can be divided to 4 different modules. This will be elaborated in section 4.1.2.

The bench scale equipment is selected to be implemented in ES2 as it relates to the characteristics of an embedded system. It is more complicated than a common embedded system like calculator, coffee machine etc, but it is not too complicated to be covered within the course. This discrete system is a real-time system where the co-ordination between the different modules is necessary. The actual bus system plans are also available for the class. For organisation purposes, the cabling for the sensors and actuators are grouped according the module. As there are different modules in this system, interactions and interfaces between the different modules can be modelled. Apart from that, there are actuators that plays different roles, or software that can be implemented for the different hardware. This characteristic is useful to introduce modularity and re-usability [VW07]. This bench scale equipment is programmed using IEC-61131-3 that is covered in ES2.

The same bench scale equipment is used for both exercise and coupled course to help students further develop the understanding they learnt in the exercise, in the application using a tool. Apart from that, this also reduces the preparation time for both exercise and coupled course. Students do not need to get to know a new application for the different exercises. Instead they can spend more time learning and implementing the theory. Using a consistent example, the students can also follow through the different chapters and connections with each other. Hopefully, this will help the students to understand the course content better. The details on how this bench scale equipment is implemented across ES2 will be elaborated in section 4.1.3.

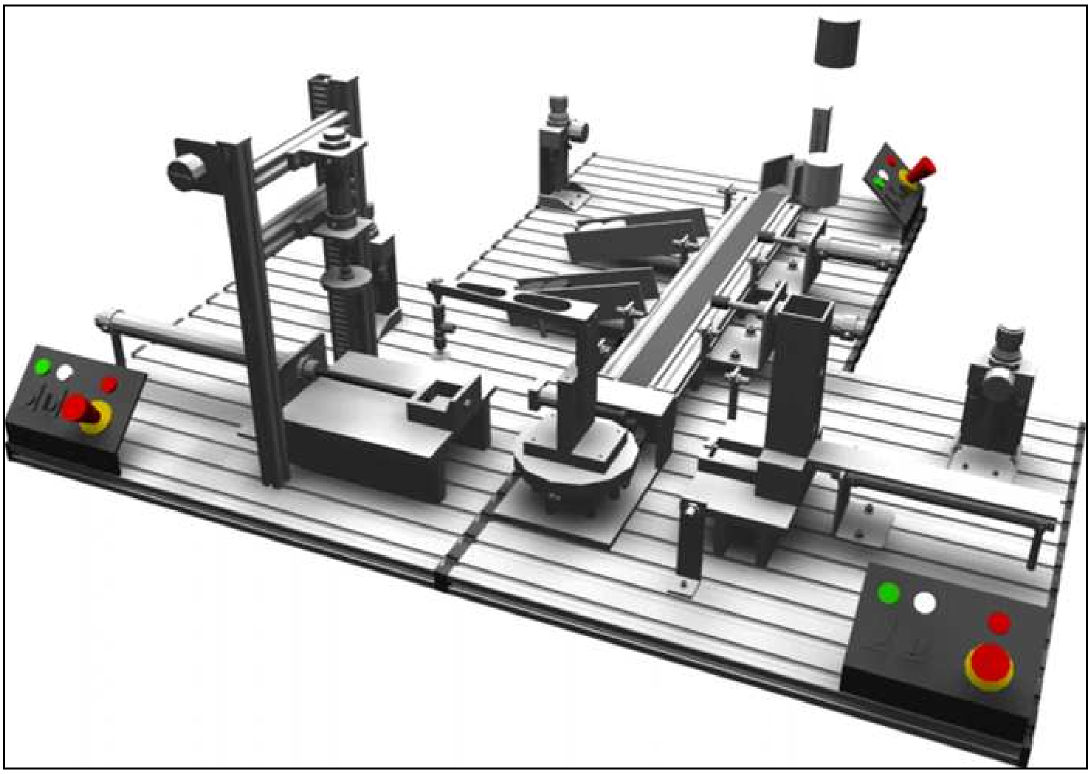


Figure 4.1 – Bench scale equipment: Stamping and sorting machine from Department of Embedded System

4.1.1 Reviews on Available Implementations

From the conducted literature review, there are projects where equipment is introduced to help students understand the course content better. There are also examples where the same resources are being shared to overcome limited resources. However, each implementation is different from the implementation in ES1 and ES2.

University of Stuttgart and Technical University of Berlin implemented Lego Mindstorms Roboters [Lego] as a teaching medium in their work [JK+08]. The purpose is to encourage secondary school students, as well as female students to take interest in engineering courses. The program is known as Roberta®. University of Stuttgart introduced new project oriented courses base on the Roberta® program. The courses are related to medical engineering, natural and science engineering, and humanities and social sciences. University of Duisburg-Essen also implemented Lego Mindstorms Roboters in the software engineering courses [HL+09]. The courses introduced in this research do not have this luxury. Apart from this, the graphical programming language for the Roboters is not the standards implemented in the industry.

4.1.2 Modules for the Bench Scale Equipment

The bench scale equipment can be divided to four modules, namely the storage module, the crane module, the stamping module, and the sorting module. Each module has a list of sensors and actuators. The functions for each module are accomplished through combination of these sensors and actuators. A list of sensors and actuators can be referred in Table 4.1.

Table 4.1 - List of sensors and actuators for bench scale equipment

Hardware	Sensor	Actuator
Inductive Sensor	Inductive Sensor	--
Optical Sensor	Optical Sensor	--
Capacitive Sensor	Capacitive Sensor	--
Pressure Sensor	Pressure Sensor	--
Spring Cylinder	Cylinder retract Cylinder extend	Cylinder extender
Pneumatic Cylinder	Cylinder retract Cylinder extend	Cylinder retractor Cylinder extender
Motor	Step Counter	Rotation Motor
Vacuum	Vacuum Status	Vacuum

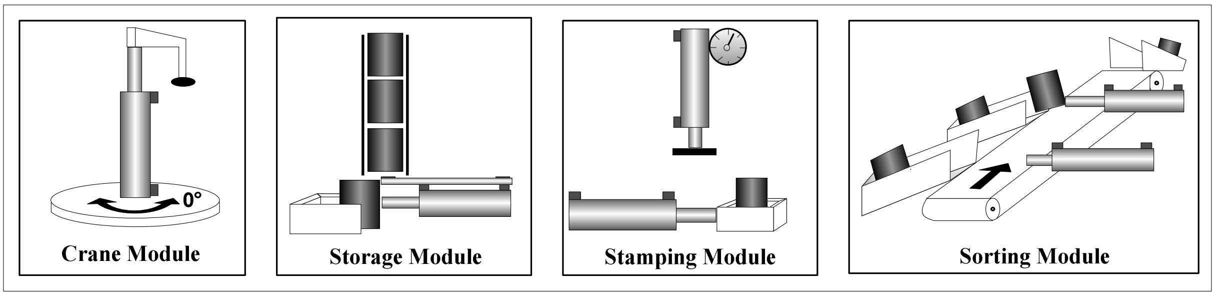


Figure 4.2 – Modules in the bench scale equipment

Crane Module:

The crane module co-ordinates the work pieces between the storage module, the stamping module and the sorting module. When the capacitive sensor senses a work piece in the storage module's container, the crane will pick up this work piece and place it in the stamping module's container for stamping. After the work piece is stamped, the crane will pick it up and place it on the conveyer belt of the sorting module.

Storage Module:

The storage module is the simplest module. The work pieces are put into the shaft manually. A capacitive sensor will sense if there is a work piece to be pushed out. The cylinder is a spring cylinder. A command will be given to the actuator, which is the cylinder extender, to extend the cylinder and thus pushing the work piece out from the shaft into a container. Sensing a work piece in the container, the crane will be commanded to pick up the work piece. As soon as the work piece leaves the container, another work piece will be pushed out from the shaft.

Stamping Module:

The stamping module stamps the work pieces according to their type. The metallic work piece will be stamped with 10 bar pressure; the black plastic work piece with 5 bar pressure and the white plastic work piece will not be stamped. When the capacitive sensor of the stamping module's container senses a work piece, the container will be retracted using the pneumatic cylinder. The amount of pressure to be applied is controlled by the stamp's pneumatic cylinder. After the stamping, the container's cylinder will extend and place the work piece for the crane to pick up.

Sorting Module:

The sorting module will sort the work pieces accordingly. The capacitive sensor will sense a work piece on the conveyer belt. The first sorting station has an inductive sensor that will sense if the work piece is a metallic work piece. If it is a metallic work piece, then the spring cylinder will be extended to push out the work piece. If this is not a metallic work piece, the work piece will follow along the conveyer belt. The next sorting station, sorts the white plastic work pieces. An optical sensor, which returns a true value when identifying a light coloured object, is stationed here to identify the white plastic work pieces. The spring cylinder will be extended to push out the white plastic work piece. The black plastic work piece that was not pushed out by the first or the second sorting station will continue to move to the end of the conveyer belt and land in the container at the end of the conveyer belt.

4.1.3 Bench Scale Equipment in ES2 Course Content

ES2 course content revolves around the development of an embedded system. The purpose is to guide the students through the development of an embedded system. It begins with the requirements engineering where the systems requirement will be gathered, prioritised and confirmed. Next, the requirements are modelled using SA/RT and SysML. After the system design, the system will be developed using IEC 61131-3. The students are taught to program using “Instruction List”, “Function Block Diagram”, “Sequential Function Chart”, “Structured Text” and “Ladder Diagram”. Next, methods to develop test cases are introduced.

During the first class, the students are introduced to the different modules and how the equipments work together. Next, a list of hardware implemented in the bench scale is tabulated. It is also highlighted that the same hardware can have different functions. The students need to always pay attention to the required functions before proceeding to solve the problem. Therefore, the same hardware can be implemented in ES2 exercise, ES2 group work and Software Tools that will be further elaborate in sub-chapter 4.4.

Bench Scale Equipment in Requirements Engineering:

During the group work for ES2 SS2009, the students are requested to develop a new system using the hardware listed in the bench scale equipment hardware list. The students are free to add any hardware. During this group work, the students learn to list down the requirements and functions for this new system. The requirements need to be recorded clearly as another group will develop this system using the specified requirements. Apart from that, the students also get an opportunity to implement the resource planning and timeline planning learnt in the class. They need to plan the timeline for the necessary task and also to assign these tasks to different members of the group according to their skills and availability.

Bench Scale Equipment in System Design:

Two modelling methods, which are SA/RT and SysML, are introduced in the chapter system design. Both these modelling methods cover a wide range of area. The focus area selected for SA/RT is the hierarchical development of a system. SysML is not only used to describe the process of a system, but reusability and modularity are also covered here. Reusability and modularity are covered here as this is a possible way to improve the programming for embedded systems.

SA/RT:

There are two major areas in SA/RT, namely the flow diagram and the architecture diagram. The flow diagrams are then refined to subsequent level until no more refinement is necessary. Refining the process by hierarchy enables the students to solve the problem “little by little”. The last level will be described using process specifications (PSPEC) or control specifications (CSPEC). Architecture diagram on the hand describes the architecture of the system, the flow of the data from one physical system to the other through specific bus systems.

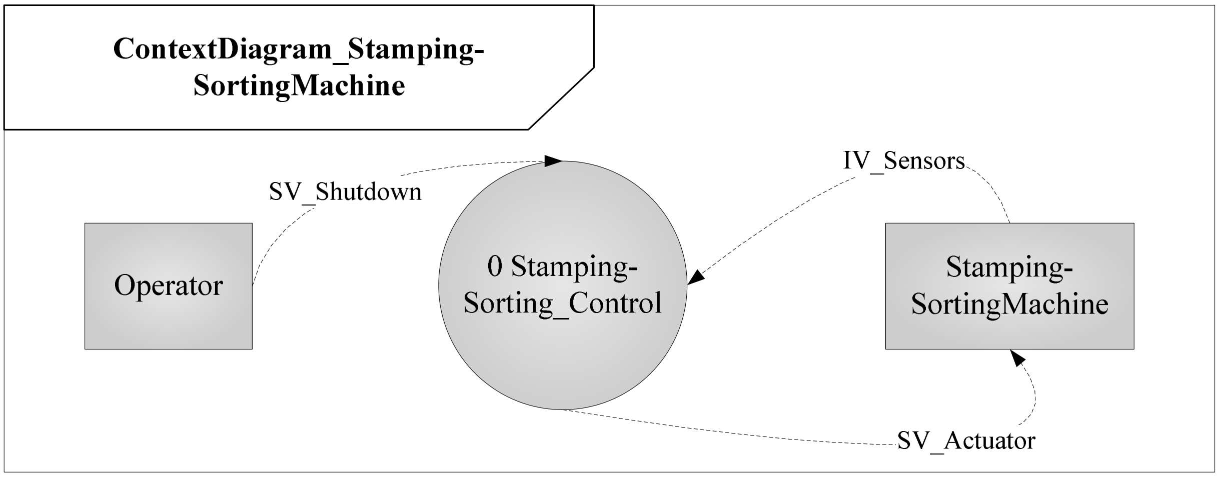


Figure 4.3 – Context diagram for the bench scale equipment's control software.

In the exercise session, the students are first requested to first draw a context diagram. As observed in Figure 4.3 there are two external elements. One is the operator, who will put the work pieces into the shaft or press the emergency button if anything goes wrong. The other external element is the physical system of the bench scale equipment. The control software is downloaded to a controller. The communication between the controller and the bench scale equipment happens through a bus system. In the beginning, the students have difficulty separating the control software from the physical system. However, using the context diagram this idea can be clearly presented.

The next exercise requests the students to refine "Level 0 Stamping-Sorting_Control". The "Level 1 Stamping-Sorting_Control" is as shown in Figure 4.4.

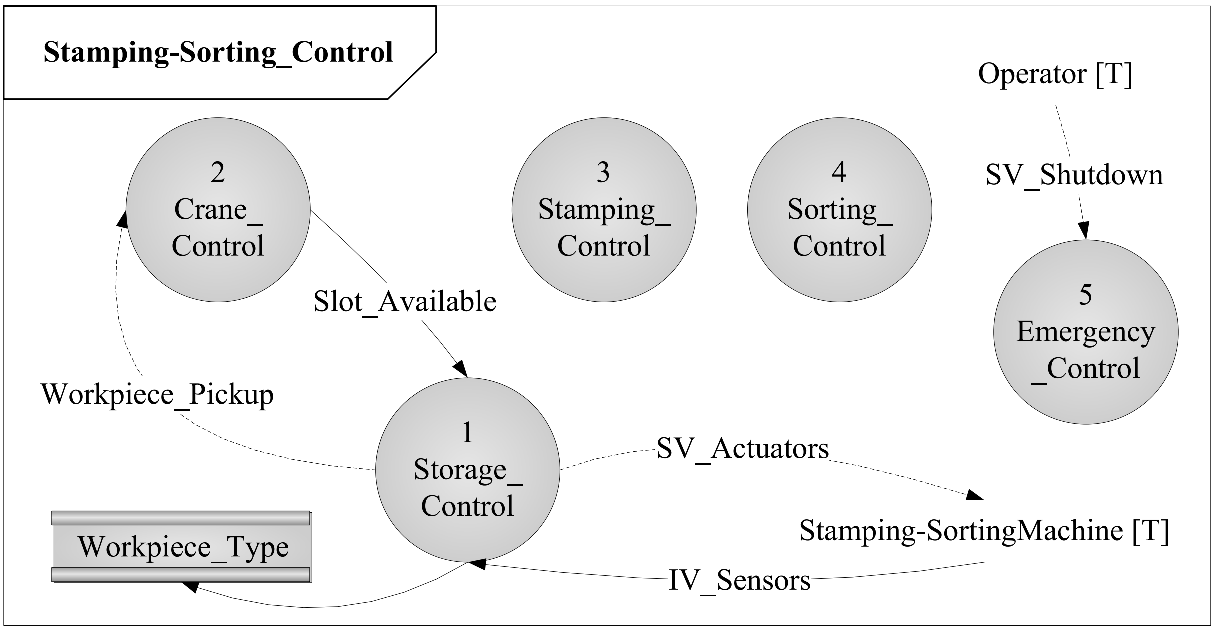


Figure 4.4 – Level 1 for the bench scale equipment

As this is the first refinement process, “Level 1 Stamping-Sorting_Control” is completed on the board with the students. According to the description of the bench scale equipment, there would be at least 4 modules. Figure 4.4 shows that there are 5 modules; the extra module is the emergency stop module. The data and control flows for the storage module are discussed. The storage module will send a signal to the crane module informing that there is a work piece to be picked up. After the crane picked up the work piece, the crane module will send a message to indicate that the container is available for new work piece. Through the capacitive sensor from the physical system, the storage module will identify if there is a work piece that needs to be pushed out. If the container is available and there is a work piece to be pushed out, a command will be sent to the physical system requesting the cylinder to extend and push the work piece to the container. As the subject SA/RT is explained using the hydraulic press example in the lecture, it is a new context to apply the knowledge in this bench scale equipment. This covers level 3 in AK’s taxonomy (section 2.5.1).

The fourth level of AK’s taxonomy is the ability to reorganise the knowledge in order to understand it better. To enforce this point, another two levels of refinements are conducted with the students. This also order to provide a complete view on the implementation of SA/RT, the students are guided. The level 2 refinement and PSPEC for module “1 Storage_Control” is further developed with the students. The students are supposed to complete the refinement for the other 4 modules. The emphasis on different levels of refinements are the abstraction level in each level is different, the consistencies of input and output for the different processes are necessary, and the modules in the SA/RT diagram can be mapped to a module or function in the program.

Next, the students were given a text description and they are required to draw the architecture diagram for the bench scale equipment. There are two types of architecture diagram, namely architecture flow diagram and architecture interconnect diagram. As the topic on bus system architecture design is not specifically covered in ES2, the students are given the bus structure in text description. Using the specifications, the architecture diagram is drawn. In the previous exercise, the students identified the information that is being sent to/received by specific actuators/sensors. Using this information they need to connect the data flow in the software, namely the system, to the hardware involved. This presents an important aspect of embedded system that is the coupling between software and hardware.

SysML:

Object oriented programming is slowly getting the attention of control software developers [Vo08]. The benefits of object oriented programming are inheritance and reusability. As the design of process had been covered in the SA/RT exercise, the SysML exercise will focus on introducing the inheritance and reusability concept to the students.

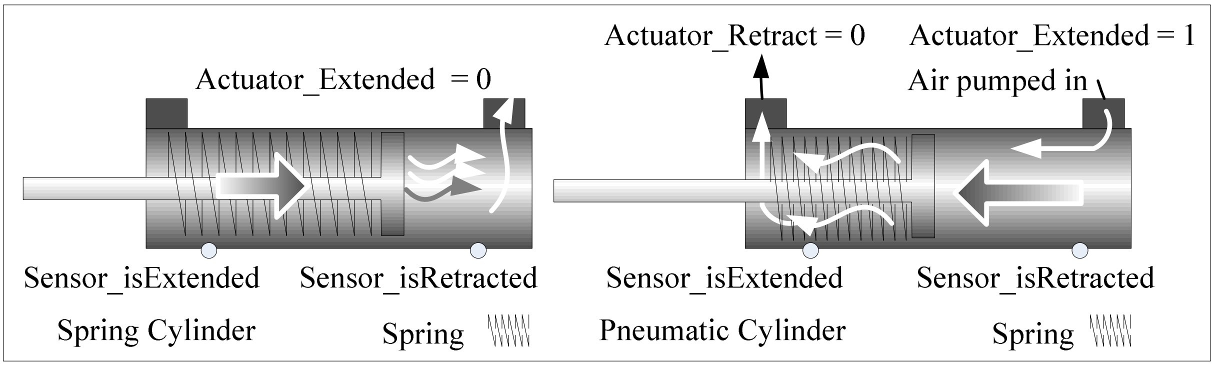


Figure 4.5 – How a spring cylinder and a pneumatic cylinder works

Before reusability can be achieved, the students need to understand on how to design modules. Therefore, the concept of modularity is first introduced to the students. The students are requested to list down the similarities in terms of physical construction of the different actuators and sensors. The list of possible software functions to achieve the different actions is also listed. The purpose here is to help the students to develop a list of basic modules. The basic modules are modules that would be re-use by other modules. Further modules can be developed using basic modules. Examples of basic modules are motor, cylinder, and sensors.

The concept inheritance, which is known in object oriented programming, is also introduced here. There are two types of cylinders (Figure 4.5) in the bench scale equipment, the spring cylinder and the pneumatic cylinder [VW07]. The spring cylinder only has an actuator that is the cylinder extender. The pneumatic cylinder on the other hand has two types of actuators, the cylinder extender and the cylinder retractor. The spring cylinders only requires the function to dispose the extend command. Here a basic module for the spring cylinder constructed. The pneumatic cylinder can inherit all the feature of a spring cylinder and on top of that has another function to dispose the retract command. Another option is to construct two types of basic module, one for the spring cylinder and one for the pneumatic cylinder. There are different issues involved in designing a module; among them are software maintenance and company's organisation. Another concept introduced is generalisation. A basic module can represent hardware, for example the inductive sensor, optical sensor and capacitive sensor are categorised as binary sensor. Figure 4.6 shows the representation of these basic modules in block definition diagram. The box in the middle represents the functions, whereas the lower box with the flow ports shows the incoming command and the out going values from the sensors.

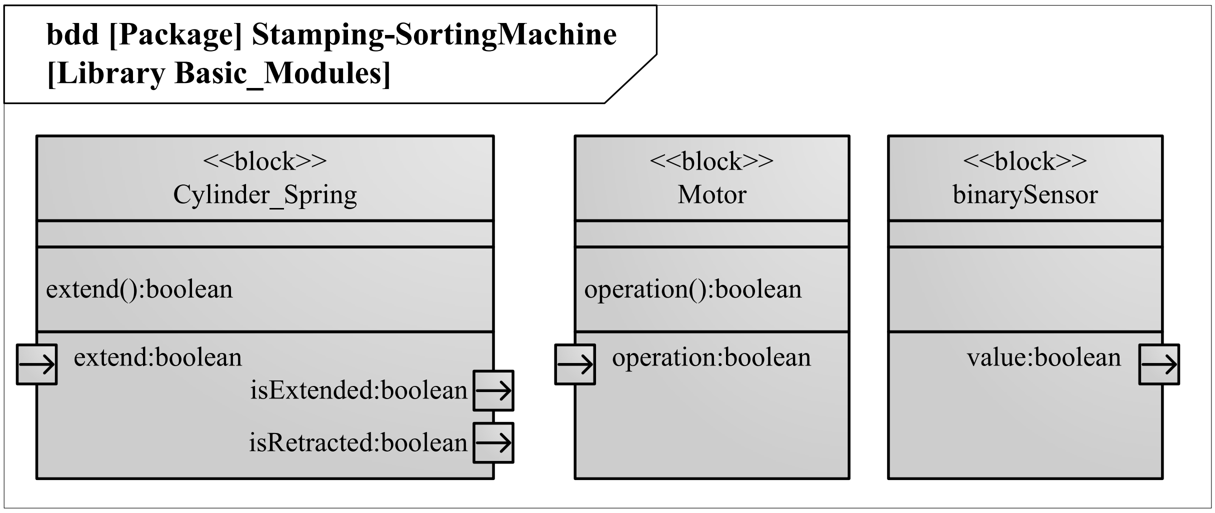


Figure 4.6 – Block definition diagram representing the basic modules for the bench scale equipment

Next, the students implemented the “reusability” concept by assembling the sorting module using the existing basic modules. The sorting module can be refined to three sub-modules “the conveyer belt control”, “sorting station for metallic work piece”, and “sorting station for white plastic work piece”. Firstly, the students are requested to design these sub-modules. For example the “conveyer belt control” module consists of a motor and a capacitive sensor. As the basic module “motor” contains the function “run”, the “conveyer belt control” module also inherits this function. It receives the command to run or to stay, and returns the capacitive sensor value to the control software. Through designing different modules using ready basic modules, the students have an experience on how to practise the concept modularity and reusability.

Having practised a few modules with the students, they are then requested to further develop the sub-modules for the “storage module”, “crane module” and “stamping module”. This allows the third level of AK’s taxonomy to be achieved, namely applying the information in a new setting. The students need almost the whole exercise session to work out the problems. This is the benefit of using the same bench scale equipment for all the exercises, as there will not be enough time for the students to implement the theory if a new application system is introduced instead.

Bench Scale Equipment in System Development:

As mentioned in section 3.2.5, IEC 61131-3 is used to develop the application programs in the automation system. There are five programming languages in this standard. The focus languages during the exercise session are function block diagram, ladder diagram and structured text. The basic of the exercise is the modelling diagrams designed in the previous exercise session. The objectives in using modelling diagrams from previous exercise are save time as introduction to new application is not necessary, and the students are able to see the importance of designing “implementable” models.

In the first question, the students are requested to implement the "1.1 Work_Piece_Identification" module using function block diagram. The information used is the PSPEC table. The function blocks are easy to implement. The students implemented the sub-module using "and" block. Different possibilities as in Figure 4.7 are discussed. The students have done digital technology. Therefore, this implementation is not new to them. Next, the implementation was shown using TrySim [TrySim]. Using different led lights to represent the different sensor value and the results, the students compared the Function Block Diagram in TrySim with the answer worked out on the board.

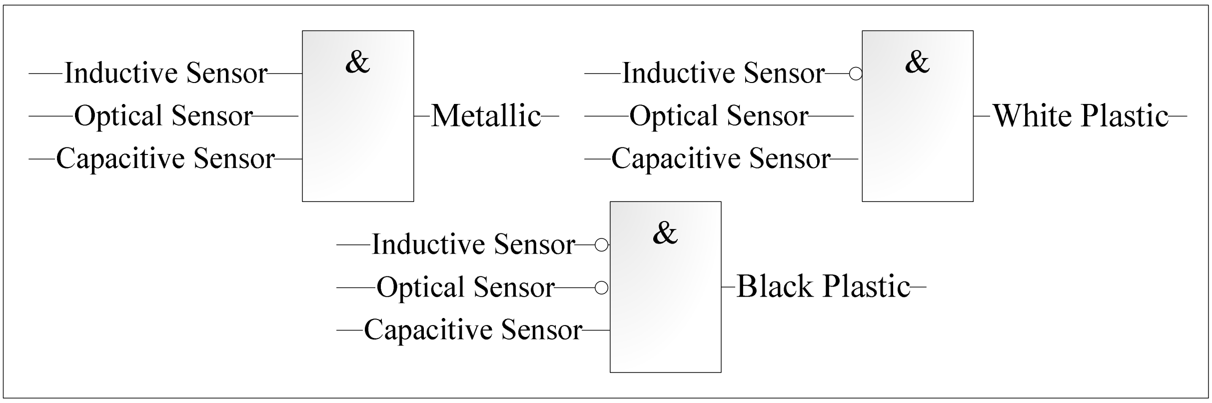


Figure 4.7 –Implementation of "1.1 Work_Piece_Identification" using Function Block Diagram

The next part of the exercise requires the students to implement the "storage module". Using the activity diagram developed in the previous exercise, the students implemented the "storage module" using sequential function chart (Figure 4.8). Different functions for the "sorting module" are also discussed, for example starting the conveyer belt when the capacitive sensor sense a work piece is on the conveyer belt, or starting the conveyer belt with a start button. A sorting application in TrySim is demonstrated to the students. Similarities between the two systems are drawn, and blocks that can be implemented using TrySim are also discussed. By demonstrating the examples in TrySim, the students are able to visualise the answer, this will appeal to students who are spatially (visually) intelligent as mentioned in "Gardner's theory of multiple intelligences" (section 2.5.2). Apart from that, students are also able to test their programs by downloading TrySim and implementing the other modules. This may work as an incentive for the students to try out the other modules practically.

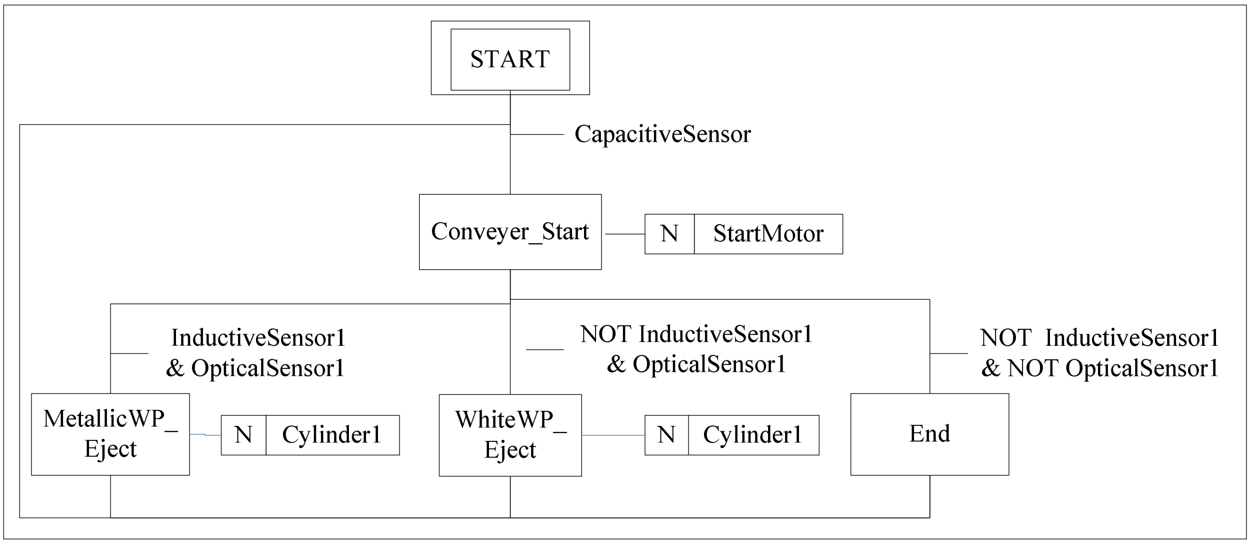


Figure 4.8 – Implementing “sorting module” based on activity diagram, using sequential function chart

Bench Scale Equipment in System Testing:

In ES2, 2 types of testing are taught, black box testing and white box testing. Two areas covered in the exercise for white box testing are Halstead complexity measure and McCabe complexity measure [MW96]. These two testing methods are related to testing of application programs for embedded system. The areas covered in black box testing are parameter test and requirements testing.

White box testing:

Halstead complexity metrics can be directly applied on the Structured Text. Structure text supports iteration loops, and conditional executions that can be mapped to the function point cyclomatic complexity. A different program is used for this exercise, as the structured text developed in the previous exercise is very simple. McCabe complexity measure depends on the control flow path. The modelling diagrams taught in previous chapters, namely the process and control flow diagram in SA/RT; and the activity diagram and state machine diagram in SysML; are based on flow of events and data. Therefore, McCabe complexity of measure, which is normally applied in the conventional software program, can be applied here. The exercise for McCabe complexity measure is based on the activity diagram developed in the previous exercise, and then later on the sequential function block diagram for the control software. The calculation is valid when the development of the application program is based on the designed models. This again highlights the importance of modelling diagrams that are informative, useful and can be applied for not only the design process, but the development and testing process as well. Interested students can try to implement the McCabe complexity measure on function block diagram. Here, the third level of AK’s taxonomy is achieved. The students applied testing method for conventional software program in the context of embedded system testing using not only the application programs but the modelling diagrams as well. As the previous lifecycle, the students can also apply it in modules that are not discussed in the class.

Black box testing:

Black box testing focuses on the input and the output of the control software. The first exercise requests the students to give the different combinations of values from sensors and the expected output. In the group work for ES2 SS2009, the students are requested to develop different test cases based on the user requirements.

Didactically, the chapter “Testing” brings a close to the development of an application program for embedded system. The students are able to follow through the whole lifecycle development of an application program for embedded system. Using the same bench scale equipment, the students are able to relate the importance of the information acquired and provided in the different phases.

4.2 Active Learning Methods Implemented – In the Class

As mentioned in sub-chapter 2.4, active learning methods require the involvement of students in the class. There are different methods grouped under the term “active learning” (Table 2.4). In this sub-chapter “in class implementation” active learning methods that are implemented in ES1 and ES2 will be presented. Here the steps for different active learning methods will be introduced. Two active learning methods namely pop quizzes and muddy card will be discussed separately, as these two methods will be evaluated separately in chapter 6.

4.2.1 Reviews on Available Implementations

There are many researches that look into implementation of active learning methods. [Mi99] mentioned that the students welcomed the used of visual aid the most. The second highest feedback is for traditional lecture, this is especially useful for students who learn by listening.

Pop quiz is another method implemented in this research. [Sn06] mentioned that pop quiz can be considered as practise test. The schedule for the tests is made known to the students. It can be at the beginning of every class, at the end of every chapter, or at a specific date [CR00], [Ha03], [Mi99], [St74]. Pop quizzes as practise test may improve the students’ grade for the exam. [Ha03] mentioned that pop quizzes improved the students result even though not significant, [Sn06] see improved results after practise test. Apart from using pop quizzes as practice test, ES1 and ES2 tested if pop quizzes would encourage students’ attendance. [DF96] mentioned that “to discourage absenteeism, instructors may want to consider giving short quizzes at the beginning of every class period”. Therefore, the implementations of the pop quizzes in this research are at random and not known by the students. The idea is since the students do not know when pop quizzes will be conducted, they will be coming to class frequently, hoping “to catch” the pop quizzes.

The third area that will be discussed is the implementation of muddy card. Muddy card is implemented to get students' feedback on the topic the least understand. [HW+02] implemented the "muddiest-point in lecture card". It is conducted after every lecture, and students are given two minutes to write their remark. Some of the instructors do ask students to identify the most important point of the lecture too. In the implementation by [KN02] also request students to write down possible criticism and positive remarks on the A7 card. This implementation is done before one third of the course is conducted. Spontaneous feedbacks on the comments are given by the "course leader" after the lecture. Response for comments that are not answered in the class will be uploaded on the course web page. This also serves as "informal" course evaluation within the semester.

4.2.2 Methods Implemented in the Lecture and Exercise

The students are provided with lecture slides a week before the lecture. The students can download the slides, print it and make necessary notes during the class. As the class setting is a large class, intensive discussion between the teaching instructor and the students cannot take place during the class. Questions that are interesting to one might not interest other students. The level of understanding between the students differs. There are questions posed that are so simple that it bores the other students upon second iteration. On the other hand, there are also difficult questions that go beyond the understanding of most of the students.

Due to the reasons above, discussion in a class is very limited. The teaching instructor will need to discuss the "too simple" questions or the "too difficult" questions privately with the students. These also influence other students to shy away from participating in the class. The passivity in class will cause students to lose interest and attention [SV10]. In order to effectively present the course content to the students, the teaching instructor needs to maintain the students' attention. Among the methods implement in the class are:

- in class demonstration,
- asking questions to the students without embarrassing them, and
- presenting the idea visually on the board.

These methods are selected based on the available resources for the course. Other methods for "in class implementation" in Table 2.4, for example PBL, team teaching, flash card, concept test are not implemented. PBL requires a facilitator for each teams, the students need to have some knowledge to begin with the problem. ES1 and ES2 are foundation courses for embedded systems, therefore it is possible that these students do not have the basic knowledge required to tackle an embedded system problem. Team teaching is not implemented due to the limited human resource as only 1 department is responsible for this course. Flash card and concept test are not implemented due to the limitation of physical resource, as there is no lecture hall equipped with computerised personal response system and distributing coloured cards for each lecture is tedious. Apart from that, muddy card is implemented to get the in between feedback from the students.

The active learning methods implemented in ES1 and ES2, taps into different types of intelligence as proposed by Gardner. The different intelligences involved will be elaborated in the sub-sections explaining the methods.

In Class Demonstration:

Students are asked to participate in a demonstration when the teaching instructor explains a particular concept. Examples of demonstration are comparison of technical processes with daily events, and the demonstration of a system with help from students.

Comparison of technical processes with daily events: the concepts of priorities and deadline were presented using a dentist – patient system [LG99]. Patients who register with the doctor and wait in turns to see the doctor are similar to the processes of the same priority waiting in queue. This reflects the "first-in-first-out" scheduling policy. Then, suddenly there is an emergency case patient who needs to leave with an international train. The time available for this patient to receive treatment is shorter and the problem is shorter. This presents the "earliest-deadline-first" problem. This particular patient then cuts the queue and is given treatment without waiting in the queue. Other scheduling processes like least laxity deadline, non-preemptive with fixed priority etc can also be explained using this "dentist-patient system" example. Other known examples that also demonstrate the problem of resource sharing, concurrency and multi-tasking is the "dining philosopher" problem.

Demonstration of a system with help from students: Students are requested to role play in the class. They can be part of a concept or part of a computer system. For example the concept of place, token and the transition was explained with the help of four students. One student holds the pen cover, and the other the pen. These students are the places with the pen and the cover as the tokens. Next, the pen and the cover will be passed to a student who plays the "transition" role. The student will put the pen and the cover together, and then passed the covered pen as a single token to the next student. Different role plays are conducted to explain the different process that can take place in a Petrinet. By involving students to demonstrate the different concepts, the bodily-kinaesthetic intelligence is being tapped into. Students with this intelligence will be able to digest the concepts better when they "act out" the concepts.

Asking Questions:

Students are asked question during the lecture to keep the students attentive. During the exercise session the students are also asked to participate in answering the questions presented in the question sheets. Students can answer question from their seat or they can work out the solution on the board. There are different question asking techniques that were implemented in ES1 and ES2

Cold calling: There are two types of cold calling implemented. The first is the round robin questioning. Similar to the round robin scheduling process, the students are asked one after the other. If the student is not able to answer the question, then the person next to him/her will be the next in row to answer this question. The method can randomly start at any row in the class. The back row, or students who were talking between themselves are good places to start. The second method is cold calling using specific description, for example, “the young man with a red T-Shirt in the last row”, or “the lady with the laptop” etc. The waiting period for a student to come out with the answer for both methods should not be too long to avoid embarrassment. The average waiting time spent in ES1 and ES2 are within 5 to 10 seconds.

Calling by name: In order to create personal touch in the class, the teaching assistant tried to remember as many names as possible. The students are called by name to answer the questions posed during the exercise. This creates positive rapport between the teaching assistant and the students. The students are willing to work out the problem on the board or to answer a particular question posed. However, the number of names remembered is very limited. At most only 15 names are remembered in the semester. It is also not exactly fair to call upon the same students all the time. Therefore, the second method is implemented. The second method uses the students’ name list. Names are called from the list of participants. There is a mix response to this method. Some of the students do participate but some of the students pretend that they were absent and did not proceed to work out the problem even though they are in the class! It is possible for the students to pretend to be absent as there are 90 names on the list but only 50 students are sitting in the class, the absent rate is almost 50% and the teaching assistant does not know all the students by name.

Random throwing: In summer semester 2009, the teaching assistant tried a new approach in the exercise. The tool is a packet of facial tissue paper. The teaching assistant posed a question and threw the packet to the first student who is supposed to answer the question. The student then in turn can throw the packet of tissue to another student for the next question. The waiting period here is similar to that when “cold calling” is implemented, which is within 5 to 10 seconds. Using this method, the students get to participate in involving other students. As compared to the calling by name method, this method is more welcomed. This also brought some fun factors into the class.

Informal Group Discussion: Apart from asking students as individuals, another alternative is informal group discussion. During the exercise session, the students are required to work out the questions that were uploaded the week before the exercise session. However, the response has not always been positive. Sometimes the students will come into the class totally unprepared. The students will be then be requested to turn to their neighbour to discuss and solve the problems. Depending on the difficulty of the question, the students are given 2 to 5 minutes to discuss the problem. The group representative will then work out the problem on the board. These few minutes of discussion with the neighbour, helps the student to answer the question asked. Discussing the ideas with peers help students with interpersonal intelligence. These students are able to remember and digest the concept better when ideas are being exchanged when discussion takes place.

Presenting the Idea Visually on Board:

Even though the information is available in the handouts for the students, it helps to draw the students' attention when this information is once more presented on the board. Instead of referring to the slides directly, it is good to especially illustrate processes on the board. For example, explaining the concepts of semaphore. The concept of semaphore is explained using a railway track example. Only one train can use the track at a time. When the train leaves the track (releases the track), then the other train may use the track. Another example is drawing a microprocessor part by part, and explains the functions of each component as they are drawn on the board. The control unit, communication unit and the arithmetic logical unit are first drawn. Then the components in these units are added to the diagram. This helps the student to group and organise the components. Another example is the presentation of preemptive scheduling to the students using the overhead projector. The students are asked to work out the problems using different colours of pen for different levels of priority. Working through the scheduling process step by step, using different colours, help students to understand the concept better. Visual representation motivates students' with spatial intelligence.

However, it is necessary to highlight here that the main medium in ES1 and ES2 are still slides and not board like "Mathematics" and "Control Theory" classes. The usage of board for "Mathematics" and "Control Theory" may be higher as the board is used not only to present new ideas, but also to solve and discuss the problems. The purposes to present the idea visually on board are:

- a change to the normal medium,
- students get to follow the ideas step by step, thereby guiding them through their thinking process, and
- providing an opportunity for the student to participate in the discussion.

Remarks to Motivating Lecture for ES2 SS2009:

During ES2 SS2009 a new teaching instructor took over the class. However, the teaching assistant remains the same. The new teaching instructor is very detailed and capable in explaining the slides, but there is almost no communication with the students. Unlike the previous semesters where motivating lecture is implemented, the involvement from the students is low.

From the 3 categories mentioned above, namely "in class demonstration", "asking questions" and "presenting the idea visually on the board", only little is done. No "in class demonstration" with students are conducted. The new teaching instructor do bring learning objects to class. One of them is a UML/SysML mug during the SysML class. The mug is being passed around during the lecture session. The other is reference books that the students can use. Demonstration objects are interesting but the impact is not as good as when the students are part of the demonstration. As for "asking questions", the new teaching instructor does ask if the students have any question. However, most of the time there is only silence and the new teaching instructor will continue on with the next slide. Cold calling is not implemented here. No students are specifically questioned, or asked to solve a specific problem on the board. The same situation does happen with the previous lecturer. However, by asking specific students, the tendency for the students to respond to the question is higher. The third point "presenting the idea visually on the board". There are a few times when the black board is being used to visually present an idea, but the drawings on the board are often very light and difficult to read. There is little or almost none motivating lecture for ES2 SS2009. Therefore, ES2 SS2009 will not be included in the semester that implemented motivating lecture for the discussion on impact of implemented methods in chapter 6.

The methods implemented during exercise session still involve the participation of students as the previous semesters.

4.2.3 Pop Quizzes

Pop quizzes are impromptu quiz conducted at the beginning of the class. The purpose of pop quizzes in ES1 and ES2 is to encourage students' attendance and to have an in between overview on how much did the students understood in the previous lectures.

As the pop quiz schedule is not known, students have to attend the class consistently to participate in all the pop quizzes. Two or three pop quizzes will be conducted through out the semesters. Pop quizzes are conducted in the beginning of the class. Students who come in late to the class can also participate in the quiz but they are required to hand up the quizzes the same time as other students. The lecture or exercise will resume as normal after the pop quizzes. It is also possible that the students only participate in one from the two or three pop quizzes offered. Students who participated in the pop quiz have a chance to get additional (bonus) points for the final exam. Extra 15% score for the final exam can be accumulated through pop quizzes. The calculation of the extra score is as follows:

$$extra\_score = \frac{acquired\_score\_for : PopQuiz1 + PopQuiz2 + PopQuiz3}{full\_score\_for : PopQuiz1 + PopQuiz2 + PopQuiz3} * 0.15$$

The extra score should not be that high that the student can do "almost nothing" in the final exam and yet still pass. The passing grade is normally 45%. Meaning if the students acquired full score in the pop quiz, they still need to have at least score 30% in the exam to pass. From this perspective the extra score can be between 10% and 15%. However, 10% extra score has little impact on the grade and might not be "attractive enough" for the students. Therefore, it is decided to award 15% to the extra score. Another alternative is to award the score only if the student passed the exam. This then again might put off student who are not sure if they can pass the final exam.

Pop quizzes will usually cover the topic from the last pop quiz to the latest exercise. This means that the subjects will only be tested once during the pop quizzes and once again during the final exam. The students need to understand the course content in order to do well for the pop quiz. This will serve as practise test for the students. The difficulty of each pop quiz increases in the course of the semester. During the first pop quiz, the questions are mainly knowledge and comprehension level questions. Questions based on the third level of AK's taxonomy, namely application level, are usually in the second or third pop quizzes. This also shows the increase of difficulty for the type of question. Therefore, even though the same 15 to 20 minutes are allocated, students have more questions to complete and the questions are also more difficult. This prepares the students to estimate the time available for them to answer the exam question.

4.2.4 Muddy Card - 5 Important and Muddiest Points in the Lecture (5IMPLe)

The muddy card (section 2.4.1) implemented in ES1 and ES2 does not focus only on the "muddiest part of the lecture". The other focus here is the "most important part of the lecture". The students are requested to write down 3 to 5 feedbacks concerning these two areas. Therefore the name "5 Important and Muddiest Points in the Lecture" (5IMPLe) is given. The differences between muddy cards from previous research to 5IMPLe are:

- the muddy card is usually conducted at a consistent basis, for example after each lecture [Mo89],
- it is implemented in a small card to limit what the students can write [KN02], and
- the feedbacks are either presented in the class or uploaded onto the internet [HW+02].

5IMPLe is implemented for the first time in ES1 WS0809. During the last lecture, the students are requested to write down what they deem as important and the subjects that they have difficulty understanding through out the semester. This is to prepare for the revision session before the final exam. The students spontaneously tear a piece of paper from their notepad to write down their opinion and hand it up to the teaching assistant before leaving the class. The responds are filtered and arranged to categories. Subjects where the students have difficulties understanding are tackled in the coming exercise.

The following semesters ES2 SS2009, 5IMPLe is implemented twice. The students in ES2 SS2009 are the same group of students who participated in ES1 WS0809. This means that they are familiar with 5IMPLe. The 5IMPLe implementations for ES2 SS2009 are more organised. Instead of asking students to just tear a piece of paper from their writing block, printed forms are distributed. In this form there are field for name, matric number, and the subjects for evaluation. The first 5IMPLe covers four topics – "Bus System", "Requirements Engineering", "System Modelling – SA/RT", and "System Modelling – SysML". The second 5IMPLe covers three topics – "Automation Architecture", "Programming in Embedded System", and "Verification, Validation and Test". The students are given the forms after the lecture and they need to hand it up before leaving the class. There are two purposes in preparing these forms. Firstly, the topics printed on the form serve as a reminder on the subjects covered. Secondly, the matric number is for data collection purposes. As the subjects covered are accumulated across few classes, the students are given 5 to 10 minutes as compared to the normal 2 [HW+02] to 5 [KN02] minutes for a class.

4.3 Active Learning Methods Implemented – Partially Outside the Class

In this sub-chapter, an active learning method that needs extra effort from the students and teaching instructor or teaching assistant to meet apart from the lecture or exercise hours will be presented. This method is known as group work. Two different types of implementation from group work will be explained in this sub-chapter. The first type is group work that covers only part of the lifecycle development. This will be known as "group work – A" (GWA) as this is the first version. The second type is group work that covers the whole lifecycle development. This will be known as "group work – B" (GWB) as this is the second version. As an incentive for students' participation, students who participate in GWA or GWB have the opportunity to accumulate additional 15% for the final exam. Students who did not participate in GWA or GWB still have the opportunity to score 100% based on the final exam.

4.3.1 Reviews on Available Implementations

Group work is one of the most common methods for active learning [SV09]. Different terms that can be associated to group work are problem based learning, group projects, industrial projects, and cooperative learning. Most of the group works require extensive supervision. Problem based learning and final year projects are conducted for students at higher semesters, with 3 to 4 person in a group [Bi99], [KC+05]. The participants for ES2 are normally in their fourth semester, and not in the sixth or seventh semesters. Unlike the group size in other researches, the group size here is bigger. The group works described in Table 2.3 have either more departments than disciplines, or the same number of contributing departments to the disciplines to oversee the work. This means that more resources are available for them. For group work in higher semester, the group work is supervised by 1 or more teaching assistants [JK+08], [HL+09]. The usual group size in ES2 is between 6 to 10 members. This group size may not be optimal for group work in lower semesters, as the students not only have to deal with course content but they might have to deal with different problems that comes along with group work, for example group coordination and job distribution. However, this decision is made based on the available resources to oversee the group. There is only one teaching assistant responsible for all the groups. Apart from this, projects in other research are a subject itself where students earn not only grades but credits. The participation in ES2 group work is voluntary and students are awarded with extra points. Therefore, the motivation to participate in ES2 group work is also lower.

4.3.2 Group Work with Partial Lifecycle Development (GWA)

As the content of ES2 focuses on the development of an Embedded System, the idea is to allow the students to go through the different lifecycle phases using group work. In ES2 SS2007, the students are divided into 9 groups. Each group has about 10 members and is responsible for a chapter covered in the course. In order to encourage communications from the different disciplines, it is a pre-requisite that the group should be 70% Computer science students and 30% Mechatronics students. The ratio between Computer Science and Mechatronics students is based on the ratio derived from the "informal registration list". Figure 4.9 describes the different tasks undertaken by each group.

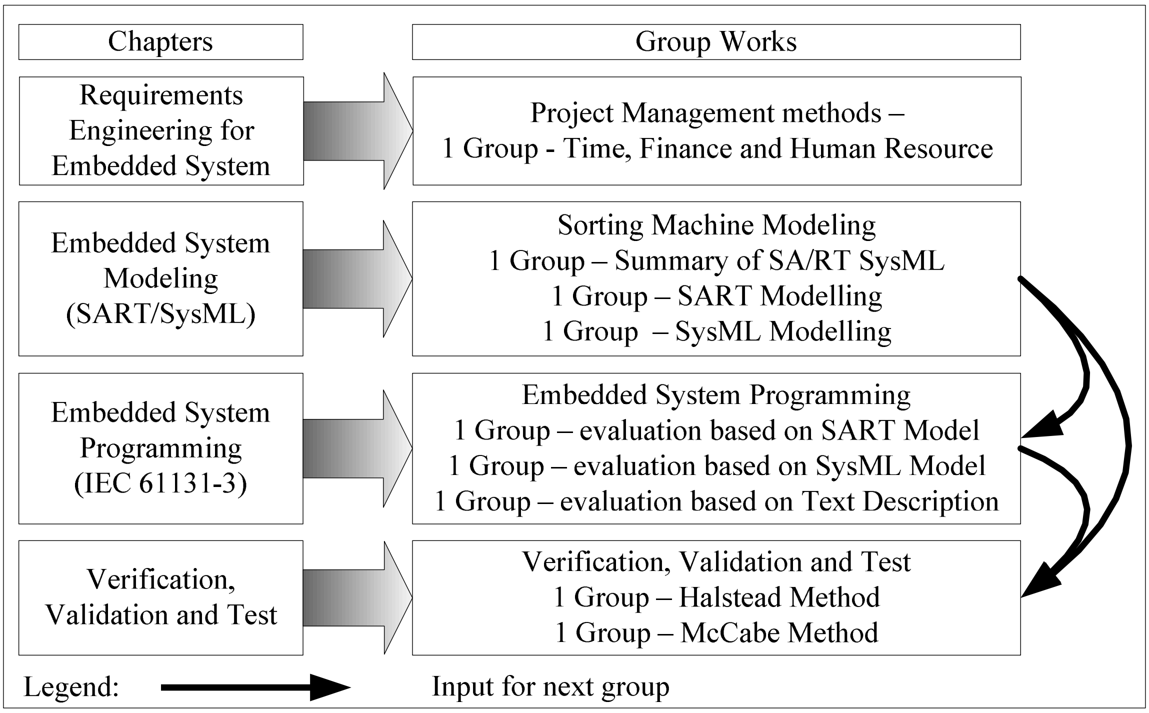


Figure 4.9 – Description for GWA

The first group presented a selected range of tools for project management. The focus is on managing the project resources and handling deadlines. The students are given a few tips on what to look for, but they need to work independently to come out with a list of available tools and select one for presentation. This trains the students to think and work independently.

The second group presented the summary of SysML and SA/RT. The basic modelling notations for both these method are presented in the class during the exercise session. Two other groups who modelled the bench scale equipment, each using SA/RT and SysML, did not present their models in the class. Instead they print out their models and are the facilitators for the programming exercise. During the exercise for “Embedded System Programming”, the class is divided into nine groups, three groups each for programming based on SA/RT model, SysML model and text description. The group members who prepared the SA/RT models and SysML models facilitate the students who developed the programming based on their models. This is because the students who are supposed to develop the programs may not understand the models. This also helps the facilitator to identify the weaknesses in the model prepared by his/her group.

The results from this programming exercise are then passed to another three groups who are responsible for the evaluation of these programs. A group evaluates the programming based on SA/RT model, another the programs based on SysML model, and the third group evaluates the program based on text description. Each group evaluates the number of functions that are fulfilled, the functions that are in the program code but not in the model or text description, and functions that are not in the program code. The group also observed if there is any reusability in the programming code.

The results are then presented during the next exercise session. The last two groups evaluate the structured text code using Halstead method and McCabe method. The complexities of each program functions are presented.

No specific tool is implemented in GWA. This is to avoid the extra burden that might come along in learning and applying a new tool. The main objective here is to observe what the students can understand from the modelling diagram. The students can select tools that they know or ordinary drawing tools to model the system in SA/RT or SysML. The programming is conducted in the exercise session by writing the program either in structured text or in function block.

Each group has a fix appointment with the teaching assistant two weeks before the presentation session. This ensures that the students are on the right track and well-prepared for the presentation. They can make extra appointments when necessary. The presentation session from each group takes about 10 to 15 minutes. Therefore, almost 30 minutes from each exercise session are allocated for the presentations from the different groups. Even though it is not a pre-requisite that every member should participate in the presentation, almost every group members presented a small section of the group work. The students are also reminded that the purpose of presenting in front of the class is to train their confidence and it is an opportunity for them to learn public speaking. In order to keep within the time limit, the number of questions addressed to the presenting group is limited.

During the implementation of GWA, cooperation within the group can be observed during the group presentation. Almost all members participated in the discussion. This shows that interpersonal intelligence is being tapped into here. Apart from this, GWA also reaches the fourth level in AK’s taxonomy, the analysis level. In order to prepare for the presentation, the students need to be able to organise the contents in a way that they can understand before presenting it to the class.

4.3.3 Group Work with Whole Lifecycle Development (GWB)

In summer semester 2009, a different concept of group work is implemented. In this group work (GWB), each group needs to follow through the whole lifecycle development of an embedded system. In GWA students who worked in a particular phase knows more about that particular phase. Therefore, it is perceived that GWB should help the students understand every phase in the lifecycle development of an embedded system better. There are two main differences between this group work (GWB) with the group work implemented in ES2 SS2007 (GWA). They are as follows:

- in GWA a group is responsible for one phase of the lifecycle, whereas in GWB a group needs to cover the whole lifecycle phases, and
- in GWA the students present during the exercise session, whereas in GWB extra appointments after the semesters are made for presentation.

Due to the same resource constraint in GWA, the students are divided into groups of tens. This time instead of making it a pre-requisite, the students are encouraged to have a group with members from both disciplines. This is to test if the students will grouped into group of Computer Science and Mechatronics students. They are told that different disciplines tend to solve problem differently and have different focus. They can make use of this opportunity to integrate with students from other disciplines. However, apart from the two groups that are organized by the teaching assistant, Computer Science students grouped automatically with the Computer Science and the Mechatronics with the Mechatronics. This shows that students are still more comfortable to work with the people that they are familiar with. This is supported by the students themselves. When asked why they did not make an effort to form a group with students from other disciplines. The answer is it is easier to work with the people you already know and whom you can trust. This shows that in order to encourage interdisciplinary communication and group work, the group distribution with students from different disciplines needs to be intentional.

As the bench scale equipment is known to the students, it saves time when the same actuators and sensors are also applied in GWB. The students are required to develop a new system based on the Mechatronics parts with different sensors and actuators found in the bench scale equipment. The mechatronics parts from the bench scale equipment serve as a guideline for the students. The students are free to implement any other mechatronics parts deemed necessary for their new system. Table 4.2 presents the different assignments in GWB. This achieves the third level of AK’s taxonomy as the students are able to apply the same mechatronics parts in a new system.

Table 4.2 - Different lifecycle phases' assignments for GWB

Project Assignment	Description
Part A: Requirements Engineering 6 points (or 2.3%)	1) Decide on the job description of each member 2) Requirement document for the system
Part B: System Modelling (SA/RT) 8 points (or 3.0%)	1) System modelling document
Part C: System Modelling (SysML) 5 points (or 1.9%)	1) Report on modules that can be reuse
Part D: System Development (Own Group) 6 points (or 2.3%)	1) System simulation 2) Report on document evaluation from part A to C
Part E: System Development (Another Group) 6 points (or 2.3%)	1) System simulation 2) Report on document evaluation from part A to C
Part F: Test and Validation 4 points (or 1.5%)	1) Test cases and test result for own system
Part G: Group Work Evaluation 3 points (or 1.1%)	1) Evaluation report
Individual points: 2 points (or 0.8%)	Awarded by teaching assistant for individual work quality

In order to make sure that there are progresses along the semesters and the work are not completed last minute, different milestones are being set up. The different part of the project assignment in Table 4.2 serves as milestones. The different project assignment will be referred to as "Part A, B, C, D etc" as described in Table 4.2. Firstly in "Part A", the group members decide on the functions of the new system. They are to describe the system using two dimension diagrams and text description. Next in "Part B", the functionalities in this new system are modelled using SA/RT. Using the process flow diagram and control flow diagrams, the students specified the flow of information from/to the sensors and actuators. Following this in "Part C", the students take notes on the modules that can be reused. For "Part A" to "Part C", an appointment was made with the teaching assistant after each mile stone. During the appointment, the teaching assistant will review the documentation, and discuss with the students if any improvements are required to make the information more useful and understandable. The students are free to use any drawing tools that they are familiar. However, the students are encouraged to draw the SA/RT diagrams using Innovator software [MID], whereas the SysML diagram are drawn using MagicDraw UML [Magic Draw]. Both these software are also covered in another coupled course Software Tools that will be discussed in sub-chapter 4.4.

It is important to guide the students during from "Part A" to "Part C", as the documentations here will be used in the development and the validation phase ("Part D" to "Part F"). During GWA, the team who developed the modelling diagrams assisted as facilitator during the exercise session for programming. In GWB a higher challenge is given, namely the documentation should be complete and understandable so that the intended system can be developed. In order to test if the documentations are understandable and suitable to develop a system, the documentations are given to another group for evaluation and development. This represent distributed development in the industry, where the design team and the development team may be in different location and it is not that easy to just “walk over” and iron out the ambiguous points. The group representatives draw blind lots to decide on the group that is to be evaluated. Only the teaching assistant knows which group is evaluating which group. As the group do no know who created the documents, they will not have a chance to question or discuss the documentations with the author. They can only develop the system based on the documentations provided. The group will also evaluate the documentations from another group. This is a form of peer assessment. Research on peer assessment shows that when using a blind process, the students are able to access their peers impartially [Po09]. Using peer assessment evaluation, the peer from another group views the documentation from a neutral point of view.

During the second half of the group work, from “Part D” to “Part F”, the students are expected to develop the system and the test cases for the system. Both “Part D” and “Part E” develops the system in TrySim and evaluates the documentation from “Part A” to “Part C”. Part D requires the students to develop their own system and to evaluate their own documentations. The students working on part E will develop the system and evaluate the documentation from the anonymous group. This means each system and documentations will be developed and evaluated by two groups. This serves as comparison. A system that is successfully developed by the original group but not the anonymous group may have poor documentations. If the system developed by both group are identical, this means that the documentations are complete and understandable. The system development is implemented using the software TrySim [TrySim]. This software is introduced in the exercise and taught in the coupling course Software Tools. The students are provided with an evaluation form to evaluate the documentations. There are four sections in this evaluation form, the first part evaluates the sketches and description of the new system; the second part evaluates the SA/RT documents – if the diagrams are consistent, if the processes extends the text description of the system; the third part evaluates the SysML documents; and the fourth part requests the students to gives an overview on which document helped them the most.

The students working on part F will develop the test cases for their own system. Here the third level of AK’s taxonomy is also achieved as the application of knowledge acquired for testing is applied in a new setting. As “Part D” and “Part E” evaluate and develop the system based on the documentations, no milestones appointment is conducted. This is to make sure that the students are not influenced by the teaching assistant. The students need to make decision on which documentation is most useful for the development. This achieved the fifth level of AK’s taxonomy, where the students need to evaluate the different methods. With the activities in GWB, the students have the opportunity to learn which documentation is suitable for what purpose. Finally, the group needs to make a final appointment with the teaching assistant to present the developed system. During this final appointment, the group representative will describe the simulated system and also the test cases. As the final appointment is during the semester break, it is not compulsory for all the group members to attend. However, at least one representative who is able to present the system should be around.

The last part of the group work requires the students to hand in the group work evaluation form. There are four sections to this evaluation form – the evaluation of team members, the contribution of lecture and exercises to the group work, own contribution, and specific questions concerning part of the project that they were working on. This is to evaluate the implementation of GWB and to see if GWB benefited the self development of the students. The results and findings will be further discussed in chapter 6. The students are not graded base on the feedback. Every student who completes the evaluation form gets 3 points for the individual score. The form can be handed in per email or in hard copy.

4.4 Active Learning Methods Implemented – Outside the Class (Coupling with another Course)

The course ES2 provides the foundation for the development of an embedded system. However, due to time and teaching resource constraint no practical classes can be offered to the students. According to the “cone of learning” theory described in section 2.5.3, practical learning is important to help students remember what they learnt. Therefore, if practical classes can be offered to the students, they will be able to apply, analyse, and evaluate the lessons learnt in class. Through the practical classes, misunderstanding due to different cognitive mind set may also be ironed out when applying the theory/concept on the software tools.

There is a course in the department that focuses on application of software tools that can assist the engineering process. This course is known as Software Tools and has the course code FB16-6959 (SWT). It is introduced in SS2007. This is a two semester hour’s week course, meaning the students need to attend a 90 minutes class/lab session each week. This course is compulsory for the Mechatronics students but optional to the Computer Science students. As these different software tools can be mapped into the different development lifecycle phases of an embedded system, SWT and ES2 are coupled together.

The coupling between SWT and ES2 benefits both courses. Firstly, the theory needed in SWT can be introduced in ES2. Secondly, the practical session that will help students remember the course content better can be conducted in SWT. Thirdly, course coupling did not add much new work load for the teaching assistants. Therefore, course coupling not only able to overcome the different cognitive mind set problem, it also solve the limited resources problem.

4.4.1 Reviews on Available Implementations

From the conducted research, there is no course coupling method to found. Examples of coupling work available are team-teaching, where two professionals are coupled to teach a class [Da95]; project coupling, where bachelor students are coupled with master students to work in a project [WW+08]; and course coupling, where the course content shares the basic information [JK+08]. The closest implementation to ES2 and SWT course coupling is the course coupling by [JK+08]. However, the course coupling by [JK+08] only shares the basic course content. The progress in one course does not effect the implementation of another course and the students in both courses are different. The course coupling between ES2 and SWT requires co-ordination as the course content is inter-related.

4.4.2 Loosely coupled Software Tools

During the first semester where SWT is introduced, the teaching assistants for SWT and ES2 discussed the content and schedule for both course. The schedule is coordinated in a way that the subject will be first taught in ES2, and then followed by the exercise, and finally the practical work in SWT. Table 4.3 shows the coupled subjects between ES2 and SWT.

Table 4.3 - Subjects covered by ES2 and SWT during summer semester 2007

Subjects in ES2 SS2007	Tools taught in SWT SS2007
Requirements Engineering for Embedded System	Requirements Engineering - IBM Rational® RequisitePro®
Embedded System in Automation Control	--
V-Model and Cost Analysis	*in SWT introduction lecture **
System Modelling using SA/RT	*in SWT introduction lecture **
System Modelling using UML/SysML	UML - Enterprise Architect for UML
Programming using IEC 61131-3	IEC 61131-3 - TrySim
Verification, Validation and Test	*in SWT introduction lecture **
	Petrinet simulation - Visual Object Net++

*Legends: * but no tools are implemented for the practical session in SWT*

As SWT is a new course, only a few selected subjects are implemented for this course coupling. In ES2 the students are taught the different methods to gather requirements for a system, the different stake holders, the resources involves and how to manage the deadline. Using IBM Rational® RequisitePro® [ReqPro], the students learnt to gather and organise data for the project, manage the resources, and extracting the necessary data from the tool. These few functions are only a few from what RequisitePro® can do. Due to the tight schedule only these few aspects of RequisitePro® are introduced to the students.

The next tool introduced is Enterprise Architect for UML [EntArch]. The same bench scale equipment but with different sorting functions is used. Instead of sorting according to one work piece type to a station, the sorting requires three different work pieces to be sorted to each station (Figure 4.10). The students defined the different objects, and the relationships between these objects. With these in mind, they drew class diagrams and composition diagrams to represent the objects. In order to model the behaviour of this bench scale equipment, the students implemented sequence diagram and state diagram.

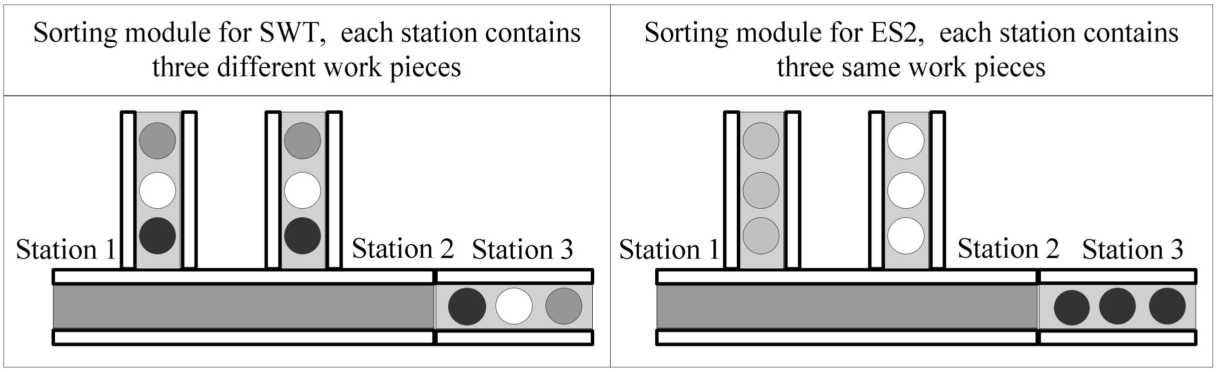


Figure 4.10 – Sorting module for SWT vs. ES2

A subject that was not covered in ES2 SS2007 but was taught in SWT SS2007 is Petrinet. The tool used to implement Petrinet is Visual Object Net++ [VON++] for Petrinet. The students are requested to model the movement of a few work pieces on the conveyer belt. The last tool introduced in SWT SS2007 is TrySim [TrySim]. TrySim is used to implement IEC 61131-3 programming. TrySim is a simulation tool. The teaching assistant provided a draft with all the necessary objects to simulate the bench scale model. The students work in pairs to program the functions required. There is no written test for SWT SS2007, the students grade are assigned based on the practical assignments in this course.

SWT SS2007 is considered as loosely coupled to ES2 SS2007 because:

- no other discussion to align the course content or course schedule is done after the semester began. Each course took its own way during the semester, and
- Petrinet was covered in SWT SS2007 but it was not taught in ES SS2007.

A few lessons to improve the course coupling were learnt through the course feedback in ES2 SS2007. The comments taken into consideration are the course scheduling, the confusion between SysML and UML, and the software functions for bench scale equipment.

In loosely coupled SWT SS2007, the schedule for the subjects in SWT SS2007 and ES2 are not discussed during the semester. This caused some problems, as at times due to unforeseen circumstances, the lectures for ES2 take longer to finish, or the schedule between the lecture and exercise in ES2 are rescheduled. This caused the practical in SWT to be conducted before the lecture or exercise in ES2. The highest feedback from the students is the confusion caused by this interrupted schedule, as they did not learn the theory before the implementation of the practical work

Both SysML and UML modelling languages share a number of similar points though they are different. Introducing two similar modelling languages at the same time causes the students to mix up the notations and the names of the different diagrams.

The students are confused with the software functions in ES2 and SWT because in their mind, the same hardware should be doing the same function. They are irritated when the sorting module function for ES2 and SWT differs.

4.4.3 Closely coupled Software Tools

Based on the feedback from SWT SS2007, adjustments are made to SWT SS2008. For the course coupling in SS2008, SWT SS2008 started two weeks later than ES2. The first two weeks, are used for organizational work. The students arranged themselves in groups. A brief introduction on the different lifecycle phases and the different tools is conducted during the second week. The practical session for SWT starts with the ILC150 Starter Kit from Phoenix Contact [ILC150a].

The second highest comment is the students are confused between SysML and UML. In response to this comment, only SysML is introduced in ES2 SS2008 and SWT SS2008. SysML is selected over UML because SysML is able to model the hardware as well as the software.

The third feedback from the students is using the same bench scale equipment for both courses confuse the students. During summer semester 2008, the difference is emphasised in the class. The teaching assistants for both courses mentioned right from the beginning that the hardware is the same but the software functions can differ depending on the way we program the system (Figure 4.10). There are two students, who “reminded” the ES2 teaching assistant that the sorting functions in SWT are different. However, this confusion is quickly put to rest. Apart from the both teaching assistants also update each other on the progress and feedback of the students after each ES2 exercise and SWT practical session.

As compared to SWT SS2007, which is considered as loosely coupled to ES2, SWT SS2008 and SWT 2009 are considered as closely coupled to ES2 because:

- there are continuous feedback between the teaching assistants for both courses concerning the schedule and progress of the students, and
- only the courses that are covered in ES2 will be implemented in SWT.

The subjects covered by ES2 and SWT during summer semester 2008 are as in Table 4.4. The tool IBM Rational® RequisitePro® is excluded from SWT SS2008 because the students require at least three practical sessions to familiarised with the tool. It is not possible to allocate more time to learn and use this tool. As compared to ES2 SS2007, Petrinet is introduced as one of the modelling methodology in ES2 SS2008. Using Petrinet the students can model a system process. This is implemented using Visual Object Net++. During SWT SS2007, UML was implemented using Enterprise Architecture. The focus for ES2 SS2008 and SWT SS2008 is SysML. SWT SS2008 introduced Magic Draw [Magic Draw] to the class. Magic Draw has a SysML plug-in to support the SysML diagram. The tool used to implement programming using IEC 61131-3 remains the same, namely TrySim.

Table 4.4 - Subjects covered by ES2 and SWT during summer semester 2008

Subjects in ES2 SS2008	Tools taught in SWT SS2008
Bus System (brought forward from ES1)	Bus System - Starterkit ILC150
Requirements Engineering and V-Model	*in SWT introduction lecture **
Embedded System in Automation Control	
System Modelling using Petrinet	Petrinet simulation - Visual Object Net++
System Modelling using SA/RT	SA/RT - Innovator 2007
System Modelling using SysML	SysML - Magic Draw
Programming using IEC 61131-3	IEC 61131-3 - TrySim
Verification, Validation and Test	*in SWT introduction lecture **

*Legends: * but no tools are implemented for the practical session in SWT*

One new tool introduced during SWT SS2008 is the Starterkit ILC 150 ETH from Phoenix Contact (Starterkit) [ILC150a]. The first reason to include this in SWT SS2008 is ES1 WS0708 did not manage to cover the subject bus system and it has to be brought forward to ES2 SS2008. The second reason is during that semester, the department received 6 sets of Starterkit. The students are able to setup the connectivity between the computer and the controller. This gives the students an opportunity to experience the different elements, for example the controller, the computer, the cable, and the software, involved in setting up a bus system. During this practical session, the students learnt how to assign an IP address to the controller and set up ftp connection between PC Worx and the controller. As the Starterkit comes with a ready program, the students also have the opportunity to make/build the program, load the program to the controller, view the variables in debug mode etc.

During SWT SS2009, a few minor adjustments are done. Petrinet is not part of SWT SS2009 because it is also not included in ES2 SS2009. Table 4.5 presents the course in SWT 2008 and SWT 2009.

There are two lecture sessions during SWT SS2009, one to introduce the different tools selection and another to introduce SysML. During SWT SS2009, the teaching instructor could only make it for one session of lecture. Therefore, SysML is not part of the lecture subject for SWT SS2009. This means that there are 3 available practical sessions for SWT SS2009 as compared to SWT SS2008, one from the lecture on SysML and two from the practical session for Petrinet. These three practical sessions are then distributed among bus system, SA/RT, and IEC 61131-3. From the arrangement of schedules in Table 4.5, it is observed that the arrangement in SS2009 is better. This is due to the following 2 reasons:

- The students have more time to work on the specific tools. For example, only 2 practical sessions are allocated for SysML and IEC 61131-3 in SS2008 as compared to 3 practical sessions in SS2009.

- The SysML practical for SS2008 is interrupted by public holiday in week 8. The practical sessions for SS2009 are not interrupted by public holidays.

Table 4.5 - Course schedule for SWT SS2008 and SWT SS2009

<table>
<tr><th>Week</th><th>SWT SS2008</th><th>SWT SS2009</th></tr>
<tr><td>1</td><td>Arrangement of Appointments</td><td>L: Introduction and Tools Selection</td></tr>
<tr><td>2</td><td>L: Introduction and Tools Selection</td><td>Arrangement of Appointments</td></tr>
<tr><td>3</td><td>Lecture: SysML</td><td>Public Holiday Break</td></tr>
<tr><td>4</td><td>P: Bus System - Starterkits (ILC 150)</td><td rowspan="2">P: Bus System - Starterkits (ILC 150)</td></tr>
<tr><td>5</td><td>Public Holiday Break</td></tr>
<tr><td>6</td><td>P: SA/RT (Innovator)</td><td>Public Holiday Break</td></tr>
<tr><td>7</td><td>P: SysML – BDD (MagicDraw)</td><td rowspan="2">P: SA/RT (Innovator)</td></tr>
<tr><td>8</td><td>Public Holiday Break</td></tr>
<tr><td>9</td><td rowspan="2">P: SysML - BDD (MagicDraw)</td><td rowspan="3">P: SysML - Block Definition Diagram (MagicDraw)</td></tr>
<tr><td>10</td></tr>
<tr><td>11</td><td rowspan="2">P: Petri net (Visual Object Net ++)</td></tr>
<tr><td>12</td><td rowspan="3">P: IEC 61131-3 (TrySim)</td></tr>
<tr><td>13</td><td rowspan="2">P: IEC 61131-3 (TrySim)</td></tr>
<tr><td>14</td></tr>
</table>

Legends: L - Lecture; P – Practical; BDD – Block Definition Diagram

For the last assignment during SWT SS2009, the students have the options to choose either to complete the assignment or to participate in an experiment to evaluate a modelling and programming tool. 1 or 2 students mentioned that they faced difficulty in developing the program for GWB as they participated in the experiment and did not take part in practical session for TrySim. This disadvantage should be highlighted earlier to the students. However, on the other hand this feedback also shows that the participation in SWT helped the students in GWB.

4.5 Comparison of Implemented Methods to Learning Pedagogies

The various methods discussed in chapter 4 have different impacts on the students. Throughout the presentation of implemented methods in chapter 4, some discussion on the impact of the methods as compared to AK's taxonomy and Gardner's theory of multiple intelligences are presented. This sub-chapter will systematically organise the impacts together.

Table 4.6 shows an overview of the methods implemented across the different semesters. 5IMPLe will not be compared to the learning pedagogies as it is a method to get feedback from the students, and not a method to pass knowledge to the students. Lab bench equipment will also not be accessed against AK's taxonomy as it is a tool used in the exercise session. The impact of using lab bench equipment depends on the context it is implemented. Actuators and sensors from bench scale equipment are also implemented in GWB, but the impact of GWB implementation differs to that of exercise. However, this will be evaluated against Gardner's theory of multiple intelligences, as a physical system with simulation can be applicable to the different intelligences.

As the changes for loosely coupled SWT and closely coupled SWT only involve the course content and the co-ordination between the teaching assistant, both these methods will be grouped under "course coupling". The students in both loosely coupled SWT and tightly coupled SWT both use specific engineering tools to solve the exercises presented in SWT.

Table 4.6 – Implemented methods across the semesters

Methods Implemented	WS0607	SS2007	WS0708	SS2008	WS0809	SS2009	Objectives
Bench scale equipment	--	X	X	X	X	X	provide consistent example to assist students' understanding
Motivating Lecture	X	X	X	X	X	--*	involve students in the class to understand concepts better
Exercise	X	X	X	X	X	X	deepen students understanding through participation
Pop Quizzes	--	--	X	X	X	--	practise test for the students, encouragement for attendance
5IMPLe	--	--	--	--	X	X	collect students' feedback on subjects presented
Group Work (GWA)	--	X	--	--	--	--	focus on a single phase of lifecycle for embedded system
Group Work (GWB)	--	--	--	--	--	X	provide opportunity to develop an embedded system
Loosely Coupled SWT	--	X	--	--	--	--	provide opportunity to implement software tools
Closely Coupled SWT	--	--	--	X	--	X	provide opportunity to implement software tools

Legends: X applicable to the semester, -- not applicable to the semester

** motivating lecture not implemented as change of teaching instructor and thus the teaching style also changes*

4.5.1 Comparison of Methods to Anderson and Krathwohl's taxonomy (AK's taxonomy)

The implemented methods are compared to AK's taxonomy to see the level of impact the method has on the students. AK's taxonomy has six levels. They are remember, understand, apply, analyse, evaluation and create. It is observed that methods that require more participation from the students achieve higher level of the taxonomy. Methods that require the students to work independently and come up with their own solution achieve higher level in the taxonomy. Table 4.7 shows the level achieved by the different methods.

Table 4.7 – Comparison of methods implemented to Anderson and Krathwohl's taxonomy

	Motivating Lecture	Exercise	Pop Quizzes	Group Work A	Group Work B	Course Coupling
Create	--	--	--	--	--	--
Evaluation	--	--	--	X	X	--
Analyse	--	--	--	X	X	--
Apply	--	X	--	X	X	X
Understand	X	X	X	X	X	X
Remember	X	X	X	X	X	X

Legends: X applicable to the semester, -- not applicable to the semester

Motivating lecture in large class environment can best only satisfy the first two levels in the hierarchy. The students are able to formulate their thoughts and understanding to answer questions thrown during the cold-calling sessions. They are also able to pose questions for the points that they do not understand. The application of the theory or knowledge learnt seldom happens in the lecture, as there is not enough time in the lecture and secondly, the main of the lecture is to lay the foundation needed for the exercise session later.

Exercise enable students to recall the information they learnt in the class to answer the question. If the questions for exercise require the students to explain (a term, trend, process etc), then the students have the opportunity to formulate their understanding in their own words. Giving questions where the students need to apply their knowledge in a new area achieves the third level – application level. The examples in the lecture should be different as the one in the exercise. This will require the students to apply their knowledge in a new context.

The levels achieved in pop quizzes depend on the questions. The questions are influenced by the course content and the time allocated. As pop quizzes normally take 15 to 20 minutes, the questions designed are also within this time frame. During the first pop quiz more knowledge based questions are being set. During the second or third pop quizzes, there might be one question that requests the students to explain a concept using their own words. This achieve the understand level. As the time is limited, the questions in pop quizzes are mostly based on the lecture contents.

GWA and GWB both allow student to recall the course content and to apply it in the specific assignment that they are supposed to complete. In order for discussion to take place, the students need to be able to put the ideas in their own words. Depending on the assignment given different levels are achieved.

In GWA, the group who introduced different tools for project management will need to evaluate and compared the different tools available before deciding on which to introduce to the class. The level analysis is being achieved as the students need to organise the information and present it to the class.

In GWB, the students used the modelling notations to model their own their system. The analysis level is achieved by organising the being able to organise it to understand the different modelling notation better. The students can organise the different modelling notations or the different descriptions method for PSPEC and CSPEC according to its functionality. For example both activity diagram and process and control flow diagram are able to represent the process in a system; or the PSPEC or CSPEC can be written in text form, table form, graph from etc. In the evaluation level, the students should be able to evaluate the strength and the weaknesses of the different modelling notations.

Course coupling focuses on the implementation of theory learnt in ES2 using the software tools. The first two levels remember and understand are covered through discussion and comparison on what is taught in the class and how the specific software fulfils these requirements. Even though the same bench scale equipment is being used, different software functions are implemented. Here the students have a chance to apply the knowledge in a new setting. The analyse level is not included for course coupling, as the modelling notation that the students should implement are predefined. The students need not to compare the different modelling notations or description methods to determine the best option that should be implemented.

4.5.2 Comparison of Methods to Gardner’s Theory of Multiple Intelligences

Gardner’s theory of multiple intelligences in 1993 has 7 elements. They are musical intelligence, intrapersonal intelligence, interpersonal intelligence, bodily-kinaesthetically intelligence, spatial intelligence, logical-mathematical intelligence, and linguistic intelligence. The different methods implemented have different impacts for the different intelligences. One person may have multiple intelligences at varying degrees. Table 4.8 shows the comparison of methods implemented to the intelligences as described by Gardner.

Table 4.8 – Comparison of methods implemented to Gardner's theory of multiple intelligences

	Bench Scale Equipment	Motivating Lecture	Exercise	Pop Quiz	Group Work (GWA, GWB)	Course Coupling
Musical Intelligence	--	--	--	--	--	--
Intrapersonal Intelligence	--	X	X	X	X	--
Interpersonal Intelligence	--	X	X	--	X	X
Bodily-kinaesthetically Intelligence	--	X	X	--	--	--
Spatial Intelligence	X	X	X	--	X	X
Logical-Mathematical Intelligence	X	X	X	X	X	X
Linguistic Intelligence	X	X	X	X	X	X

The physical bench scale equipment is not presented in the class. However, the students can see the bench scale equipment in the lab. A three dimension simulation for this bench scale equipment is available and is used in the class. With this the spatial intelligence is covered. Students with spatial intelligences are more sensitive to visual factors, like depth, space, colour and movement. The description for the system is in text and the process requires logical steps. These cover the linguistic intelligence and logical-mathematical intelligence.

Much can be done in a motivating lecture. By using slides, with flows and logics and diagrams, to explain a certain concepts, the spatial intelligence, logical-mathematical intelligence, and linguistic intelligence are being covered. Apart from using slides, in class demonstration students' participation allow bodily-kinaesthetically intelligence students to remember the concept presented using body movements. The volunteers who demonstrated a certain concept might not be the students with bodily-kinaesthetically intelligence. However, this intelligence is included in motivating lecture, as this can potentially happen. During motivating lecture with active learning, the students are given the chance to pause and think about the idea or to discuss it within informal group learning. This covers the students with intrapersonal intelligences or interpersonal intelligences.

During the exercise session, the students also get a chance to either work out the problems individually or with a partner. Therefore, intrapersonal intelligence and interpersonal intelligence are covered here. They can do this during their own time as the exercise questions are uploaded a week before the exercise sessions. Apart from that, the students are also given a few minutes to prepare for the questions before the discussion on the questions starts. The students are also at times requested to participate in demonstrating a certain concept. For example during the exercise for "scheduling", the "sleeping barber" is demonstrated with participation from students. Therefore, bodily-kinaesthetically intelligence is covered here. Diagrams to explain different concepts are also implemented during the exercise session. This provides the spatial intelligence students the opportunity to digest the information better. The need for logical-mathematical intelligence and linguistic intelligence students are catered by having logical-mathematical questions and questions posed in textual form.

The pop quizzes should be completed individually. Here taps into the intrapersonal intelligence as it provides the room for one to solve and digest the problem on its own. As no discussion is allowed, the interpersonal intelligence is not covered here. The pop quizzes implemented did not require students to neither draw nor model. Therefore, spatial intelligence skill is not considered as being catered here. Logical-mathematical intelligence and linguistic intelligence are covered as the students are required to understand the questions and solve the basic problems.

Group work and course coupling are very similar in terms on how the methods influence the students with different intelligences. The only difference between group work and course coupling is course coupling does not “officially” include intrapersonal intelligence. Students in a group work can choose to complete the assignment with other members or to solve the problem on their own. However, the students in course coupling have to work in team; the room to work alone is not officially provided. Therefore, the intrapersonal intelligence is not included in course coupling. The spatial intelligence student is able to visualise the system that they are going to develop, how the system functions, and the simulation in TrySim. The logical-mathematical intelligence student is able to write down the functions step by step, model it in SA/RT consistently, and describe the details using tables and other types of specifications. The linguistic student has the chance to describe the modules involved in text description.

Musical intelligence is not catered in any of the methods implemented. Example on how students with musical intelligence can be catered is remembering the information through music. The second least catered intelligence is bodily-kinaesthetically intelligence. Only motivating lecture and exercise tap into this intelligence by having concept demonstration with participations from students. Methods that cover larger range of intelligences may appeal to bigger group of students. However, as this is an engineering course, most students who are here may have logical-mathematical intelligence. This is intelligence is covered by all the methods implemented. The linguistic intelligence is also fulfilled by all the methods, as communication through spoken words and written text is most common in discussing and passing on knowledge.

5 The Design of Techniques to Evaluate the Implemented Methods

In order to assess the effectiveness of the different methods implemented in ES1 and ES2, different techniques to ensure correct and fair analysis need to be identified or designed. One of the approaches to measure the effectiveness is feedbacks from involving parties. This involves the students, teaching instructors and teaching assistants. In order to collect feedback from the students departments' course evaluation and group work course evaluation is being designed. The feedbacks from teaching instructors and teaching assistants are conducted through discussion and informal reviews of the methods. However, as this is not officially recorded, they would only be used for heuristics discussion.

As it is expected that the implemented methods will help students to understand the course content better; the next option is to use the students' final exam grade for evaluation. As different methods are implemented in the different semesters, it is believed that the impact of each method will be reflected in the final exam grade. However, comparing the grade from the different semesters might pose a problem. It is also possible that one group of students are "naturally" better than the other. Therefore, the improvement might not due to the methods implemented but rather the "natural" talent that they have. In order to have a fair and justifiable comparison, the final exam grades for other courses in the semester are used as control data. This will eliminate the factor that a particular batch of student has good final exam result for ES1 and ES2 because they are especially good and not because the implemented methods are effective.

As the implementation of 5IMPLe itself is an evaluation, it is not noted in the departments' course evaluation. This method also does not directly impact the students' final exam. Therefore, the technique to assess this method is based on the response for 5IMPLe itself.

As described in section 2.3.3, one of the problems faced by institutes of higher education is limited resources. Therefore, it is also important to know the effort required for each method. In order to achieve this, the teaching assistant who is responsible for the implementations recorded the hours spent for each method using a time sheet.

Each technique is only suitable to evaluate certain methods Table 5.1 presents the list of techniques that are implemented to evaluate the different methods across the different semesters. The following sub-chapters will elaborate each technique one by one. First the evaluation, then followed by the final exams for each semester, the feedback from 5IMPLe and finally the time sheet for efforts required The analysis from these different techniques will be presented in chapter 6.

Table 5.1 - Different techniques to evaluate the methods according to the semesters

Techniques for (Methods Implemented)	WS0607	SS2007	WS0708	SS2008	WS0809	SS2009
Department's Course Evaluation (Lecture and Exercise)	--	X	--	X	X	X
Group Work Evaluation (Pop Quizzes, GWB, Bench Scale Equipment, Lecture and Exercise)	--	--	--	--	--	X
Control Data Exam Result (GWA, GWB, Pop Quizzes, Course Coupling)	X	X	X	X	X	X
Course Exam Result (GWA, GWB, Pop Quizzes, Course Coupling)	X	X	X	X	X	X
5IMPLe Feedback (5IMPLe)	--	--	--	--	X	X
Time Sheet for Effort (All Methods Implemented)	--	X	X	X	X	X

Legends: X implemented in the semester, -- not implemented in the semester

5.1 Using Department’s Course Evaluation as Evaluation Technique

The university conducts a course evaluation during every summer semester. This evaluation is prepared by the “Study and Learning Department” University of Kassel. However, this evaluation is not suitable to evaluate the methods implemented. Therefore, the department course evaluation is being used to evaluate the implemented methods. There are 2 reasons why the department’s evaluation is being used instead of the university’s evaluation. Firstly, the university evaluation is conducted in the middle of the semester and the questions asked are limited. The department’s evaluation is conducted at the end of the semester and so the students might have a better overview about the course. Secondly, the University decided to use electronic evaluation for summer semester 2007. During this evaluation, each student is given a unique login. Only 15 students responded to ES2 evaluation and the feedbacks are poor. The department’s evaluation on the other hand has 44 respondents and the feed is between moderate to good.

The sections below will first discuss on the evaluation categories, namely the areas of evaluation. During winter semester 2008/2009, the department’s evaluation form is being modified to acquire information needed for this research. This will be presented in section 5.1.2. Finally, the process to conduct this course evaluation will be presented.

5.1.1 Evaluation Categories

The evaluation form has 8 categories. The first concerns personal information, for example age, sex, discipline, and semester. The second category is the motivation, this evaluates if the students are interested in the course. The third category evaluates the difficulty of the course. The fourth category evaluates the lecture session. This includes the presentation method, clarity and understandability of the lecture, the coordination between lecture, exercise and practical etc. The fifth category evaluates the exercise session, whereas the sixth category evaluates the practical session. This category caused some confusion with the students, as there are no practical session for ES2. The “practical” session for ES2 is the course coupling with SWT. There are students who participated in SWT who answered this section according to their learning experience in SWT. There are also students in summer semester 2008 who mentioned “there was no practical session”. Students have less confusion after they are told that only the students who participated in SWT need to give feedback to this question. However, this also highlights that the result on practical session in summer semester 2007 cannot be used for the evaluation of SWT as the students might have misconception. The seventh category concerns the physical setting of the lecture hall, for example tables, chair, light, temperature etc. ES1 takes place in the same lecture hall every winter semester. The same applies to ES2 in the summer semester but in another lecture hall. The feedbacks from the students on these external elements will be used as the bench mark for assessment when necessary. As the tables, chairs and the lighting system in the lecture hall did not change through out the years, the evaluation for these external settings should be similar for all the evaluation. The ninth category is evaluation on personal commitment. This category evaluates how many hours the students spent revising for their studies, how often they visited the classes, and how often the students visit the lecture during the semester.

Likert scale [Li32] is used in the evaluation. Depending on the question the scale for answer can be as in Table 5.2.

Table 5.2 – Likert scale for the evaluation

always	1	2	3	4	5	never
sufficient	1	2	3	4	5	insufficient
very useful	1	2	3	4	5	very unuseful
almost never	1	2	3	4	5	almost too often
lively	1	2	3	4	5	boring

too slow	-2	-1	0	1	2	too fast
too little	-2	-1	0	1	2	too much
too seldom	-2	-1	0	1	2	too often

The first type of Likert scale has the value between 1 and 5. 1 represents the most positive answer while 5 is the most negative answer. The questions are positively formatted. Depending on the question different options of answer will be available as selection. For example to the question “does the course content correlate to your interest?” The second type of scale in the evaluation form is between -2 and 2. -2 and 2 represents extremism. The most positive selection here is 0, followed by -1 and 1.

5.1.2 Amendments on the Evaluation Form

A few changes were done on the general department's course evaluation form to gather more information. During the evaluation for ES1 WS0809, a few extra questions were added.

- Students are asked if they participated in the other recommended courses for the semester. This is to evaluate if it is fair to use the overall result from the "recommended courses" as the control data. 57% of the Computer Science students and 97% of the Mechatronics students follow the semester plan.
- Students are asked if they have any working experience. This is to evaluate if the study behaviour is anyhow influenced by working experience. Section 2.6.3 mentioned that higher percentage of Computer Science students have technical school qualification, namely 48% as compared to 33% of Mechatronics students. However, when comparing the working experience, the department evaluation shows that Mechatronics students have more working experience. From the 21 Mechatronics students who replied 14% worked as student assistant (Studentische Hilfskraft), 52% had apprenticeship (Ausbildung), 38% had hands-on training (Praktikum), and 43% had other jobs before. The percentages for the 32 Computer Science students are at 9%, 28%, 25%, and 28% respectively.
- The age is also asked. This is to investigate if both groups of students are of the same age. The average age for Computer Science students is 22.5 and 23.6 for the Mechatronics students. The age is not significantly different with 20% actual confidence level.
- The questions are asked more precisely. For example the students should only evaluate the consultation session with teaching instructor or teaching assistant if they have been to one of the consultation hours. Another example is instead of asking "How many hours in a week do you spend revising for your study?" it is changed to "How many hours in a week do you spend revising for this course?" The answers provided to the former question ranges between 2 hours and 40 hours a week. It is possible to say that the students who answered 40 hours, meant the total hours spent for all the courses and the students who answered 2 hours meant the hours spent for this course. However, the in between answers like 5 to 10 hours are difficult to judge if it is the total hours for all the course or if it is the hours spent for this course.
- The options provided are more precise. Instead of giving the scale "regular-seldom" to the question "How often did you attend this course in this semester?", another set of answer with the scale "1-3", "4-6", "7-9", "10-12", and "13-15" was provided.

With these different amendments, a more accurate conclusion can be drawn for the evaluation.

The evaluation form for ES1 WS0809 is printed in 2 pages per page and is 3 pages long. The students sigh and moan when filling up this lengthy evaluation form. The evaluation form is reduced to 2 pages using 2 pages per page printing for ES2 SS2009. The demographic questions are reduced, and the questions concerning the recommended subject are also left out.

In order to get possible connections between the evaluation responds and the exam result, each evaluation form has a form number. The students are requested to mail (or write the grade on the department's board) their final ES exam's grade with the unique form number. Part of the evaluation form with the form number can be torn away for the students' keeping. This can remind the students of its form's number. This evaluation will not influence their exam grade.

5.1.3 Conducting the Evaluation

The department's course evaluations are normally conducted in the summer semester, namely ES2 SS2007, ES2 SS2008, ES1 WS0809 (the only winter semester), and ES2SS2009. Except for ES2 SS2009, the respondents for the evaluation are more than 50% of the course participants. The evaluations are normally being conducted between the 11th and the 13th week of the semester. The students will not be notified before hand on when the evaluation will take place. Students who participated in the evaluation are the students who attend the class for "normal learning" purposes, and not especially to evaluate the class. Evaluation forms are distributed at the beginning or at the end of the class. The students are given 10 to 15 minutes to complete the evaluation form. The evaluation form will be collected directly when the time is up. The department evaluation will be analysed internally. Result from the evaluation will be presented to the students during the last exercise session and whenever possible, the feedbacks will be integrated into the coming semester.

5.2 Using Group Work Evaluation as Evaluation Technique

Through out ES2 course, two group works are conducted. The first group work is conducted in ES2 SS2007 (GWA), whereas the second is conducted during ES2 SS2009 (GWB). The feedbacks on GWA are through the comments from the department's course evaluation form. The GWB implemented in ES2 SS2009 is different from GWA. In order to cover more grounds to evaluate if GWA or GWB is more suitable for the students, a group work evaluation form is designed. Aspects of group work that are too tedious to be covered in the department's course evaluation are for example team work, and lessons learnt from group work. The group work evaluation form will be discussed in the following sections. Firstly, the evaluation categories will be discussed. Next, the method on how this evaluation is conducted will be presented.

5.2.1 Evaluation Categories

In order to evaluate the different aspects of the group work, four categories are designed for the group work evaluation. The first category evaluates the co-operation between the team members. The student first evaluates the group work on the whole and followed by the specific members in the group. Following this, the students describe what they learnt from other team members and what other team members learnt from them. This provides the room for the students to reflect on contributions of each team member.

The second category evaluates the relevance of the course content for the group work. The students evaluate if the lecture, exercise or the bench lab equipment helped them to progress in the group work. The students also evaluate if the appointments with the teaching assistant was effective for them to conduct the group work. Lastly, the students need to decide if the current lecture to exercise ratio is sufficient to help them understand the course content.

In order to evaluate the impact of group work toward students' understanding on course content and the commitment they have. The third category requests the students to comment on both this aspects. The number of hours spent revising ES1 from last semester, when pop quiz was conducted, as compared to the number of hours spent revising ES2, when group work was conducted, are also asked. The students are also free to give comments on what they like about the group work, and what can be improved.

The fourth category is subjective questions that are divided according to the group work's assignment. The students only need to answer the questions for the assignment part that they are responsible for. Depending on the assignment part, the students might be asked what do they find most useful, most difficult, and recommendations to overcome the challenges they faced when doing this assignment.

The evaluation scales are mostly based on Likert scale with the scale "totally agree – totally disagree". The statements are positively formed. Totally agree is the best rating, whereas totally disagree is the poorest rating. Apart from this another two scales were used. One is to evaluate the ratio between lecture and exercise, and the other is the scale for hours spent working on ES2 subjects. The scales implemented are as in Figure 5.1.

	totally agree				totally disagree
The lecture content is sufficient for me to start off with the group work.	☐	☐	☐	☐	☐
The number of hours (Std.) spent for ES2	<1Std. ☐	1≤Std.<2 ☐	2≤Std.<3 ☐	3≤Std.<4 ☐	> 4 Std. ☐
The ration between lecture (V) with exercise (Ü)	4V:2Ü ☐	3V:2Ü ☐	2V:2Ü ☐	2V:3Ü ☐	2V:4Ü ☐

Figure 5.1 – Scales used for group work evaluation

5.2.2 Conducting the Evaluation

As described in section 4.3.3, there are seven parts of assignment for GWB. The group work evaluation is the last assignment. The group work evaluation is not anonymous. Three points will be awarded to the students who hands in the evaluation report. This is individual points. Each group member receives the evaluation form through Email; the students can fill up the form and send it back to the teaching assistant by Email or hand in the form in hard copy.

5.3 Using Exam Results from Other Courses as Evaluation Technique

Control data is needed to have a fair comparison across the semesters. It is possible that a certain batch of students are smarter than the other, and therefore the improvements of exam results are not due to the implementation of a particular method, but simply because the students are smarter. The average for control data will be known as “average control grade”. The exam results from other courses are used as control data here. The following section will describe the criteria used in selecting the relevant courses for comparison.

5.3.1 Determining the Criteria to Select Courses for Control Data

There are many courses offered in a semester. However, not all the courses are relevant to the students participating in ES1 and ES2. In order to have more accurate comparison only courses that are relevant should be used. Two criteria are used. Firstly, both Computer Science and Mechatronics students have a course plan “recommending” the suitable subjects that they should take each semester. The courses selected here are the “recommended” compulsory courses for the Computer Science and Mechatronics students for the particular semester. For example the Computer Sciences students in ES1 are those in their third semester, and therefore only the “recommended” Computer Science courses in the third semester will be considered. The same applies to the Mechatronics students and the ES2 course in the fourth semester. Even though it is not compulsory for the students to follow the semester plan, 57% of the Computer Science students and 97% of the Mechatronics students do follow the semesters plan. This information is collected through one of the course evaluation. Therefore, it is reasonable to use the final exam grade from these courses for the particular semester as control data.

Secondly, the suitable class size needs to be considered. Even though the courses selected are all “recommended” courses, some classes might have lesser students attending than the other. Section 2.3.2 mentioned that exam results correlate positively with the class size. This is also observed from the final exam result for the courses mentioned above. For example the class “Theoretical Computer Science – Formal Language” (Theoretische Informatik – Formale Sprache” has average grade 2.25, and the class “System Programming” (System Programmierung” with 42 students has average grade 2.3. Bigger classes like "Programming Methodology" (Programmiermethodik) with 71 students has average grade 3.2, and "Database I" (Datenbanken I) with 89 students has 3.3 as average grade. Therefore, only courses with more than 50 students are taken into consideration for control data.

Students participating in the "compulsory courses" mentioned that extra points for participating in classes, handing in exercises, or handing in assignments are also awarded. However, as it is not possible to determine the exam grade without these extra points, the final exam grade from the "Student's Centre" of the faculty will be used as it is. The average grade for control data will be known as "average control grade".

5.4 Using Final Exam Result for ES1 and ES2 as Evaluation Technique

In order to evaluate the effectiveness of methods that are directly involved with the students' grade, the final exam result from ES1 and ES2 is being used. Firstly, a short introduction on the course exam result will be presented. Next, techniques to meaningfully analyse these results in chapter 6 will be presented.

5.4.1 Background for ES1 and ES2 Exam Result

The course exam results are the results from the final test conducted for ES1 and ES2. Each written test takes 60 minutes, and the students can score maximum 60 points. The points are then changed to percentage and will be graded. The grades that can be assigned to students are as in Table 2.6.

Students who participated in pop quizzes or group work have the opportunity to accumulate 15% extra points for the final exam (see section 4.2.3 for pop quizzes, section 4.3.2 for GWA and section 4.3.3 for GWB). The final grades for the students are determined using the following steps:

- Final exam points based on the exam results are calculated for the students.
- The exam grade's curve is determined based on the final exam grade.
- The extra 15% is added to the final exam points. Students who participated in pop quizzes or group work will get the "upgraded" exam grade based on this new final exam points.

5.4.2 Using ES1 and ES2 Exam Result Meaningfully

In order to assess the implemented methods correctly, it is necessary to compare the exam result meaningfully. Based on the description above, there are two types of points here, namely "No Extra Point" (NEP) and "With Extra Points" (WEP). NEP is the score without the extra points and WEP is the score with the extra points. The average for the course exam result will be known as NEP average grade and WEP average grade.

The following techniques are used when comparing the exam result to evaluate the implemented methods:

- No direct inter semester comparison using NEP or WEP average grade should be conducted. This is because the students might be of different "quality".
- Assessment on effectiveness of a method should be based on the difference and the significance of difference with average control grade.
- When comparing the grade with control data, WEP average grade will be used. This is because the WEP average grade is the "official grade" and some "compulsory courses" in control data also have "extra points". Comparison with control data will gauge the performance of ES1 and ES2 to other relevant courses.
- When comparing the grade between students who participated in a particular method with those did not, the NEP score will be used. This comparison will enable the assessment of the direct impact of the methods implemented on the students who participated in them.
- The average grade between students in the semester should use NEP average grade if "soft skills" obtained in group work are not taken into consideration.

5.5 Using 5IMPLe Feedbacks to Evaluate 5IMPLe

As described in section 4.2.4, 5IMPLe is implemented once in winter semester 2008/2009, and twice in summer semester 2009. As the department's course evaluation only caters for lecture, exercise and practical work, it is not possible to evaluate 5IMPLe using this source. On top of that 5IMPLe is an evaluation itself. From the course of this research, it has yet to determine the relation between students' evaluation on the courses with their exam results; therefore, it is deemed as inappropriate to evaluate 5IMPLe using the final exam's grade. Another factor that influences this decision is the number of time 5IMPLe is being implemented. If 5IMPLe had been implemented more frequently, for example for the assessment of each chapter, then it might be more possible to look for a relationship between both aspects. However, this is not the case. Therefore, the evaluation on 5IMPLe will base on the feedbacks received through this implementation.

5.6 Time Sheet for Implemented Methods

In order to estimate the required effort for each method, time sheet recorder by the teaching assistant is used. The time sheet is half hourly basis daily record on the work being completed for each working day. Any activities that require more than 15 minutes to accomplish will be recorded as one half hour slot, and any activities that require less than 15 minutes will not be recorded. This simplifies the recording process. However, it is necessary to take note that this time sheet does not include the time needed to conceive the ideas, for example the idea on GWA and GWB. These ideas are sketched and planned irregardless if its office hour or non office hour. Only the “official” discussion hours with colleagues, when the idea is “about mature” are recorded.

Using this time sheet the number of hours needed can be calculated. The time sheet is recorded through out the 6 semesters, where the different methods are implemented. Therefore, it is possible to compare what are the efforts needed when the method is first implemented and what are the effort required when it is being implemented for the second time etc. Even though, the hours recorded is not the “exact hours” spent, but this is the closest approximation available.

6 Evaluations on Implemented Methods

Chapter 5 presented the techniques to evaluate the implemented methods. Using the data collected from these different techniques, the methods as summarised in Table 4.6 will be analysed.

In order to provide a clearer view on how the statistical analysis is conducted, the first sub-chapter will introduce the foundation of statistical analysis. This is then followed by the analysis on the effectiveness of each implemented method. The analysis begins with the implementation of bench scale equipment, followed by “in class implementation” for active learning, then active learning methods implemented “partially outside the class” and lastly active learning by “coupling with another course”. This out line is similar to that in chapter 4. Each sub-chapter will present the statistical findings for the particular method, followed by discussion of the results. The last sub-chapter will calculate the number of hours spent to prepare and implement the different methods.

6.1 Foundation of Statistical Analysis

The data had been analysed using SPSS [SPSS] and is found normal. Therefore, it is valid to use 2- tail t-test, analysis of variant (ANOVA), and correlation analysis to affirm or disprove the hypothesis. The probability of error stands at $\alpha = 0.05$ unless other wise mentioned.

The tables in the following sub-chapters/sections have the following legends:

- CS or Computer Science – Computer Science students
- M or Mechatronics – Mechatronics Students
- N – number of input (students, exam results, evaluations’ feedbacks etc),
- Avg. or Average – average value for data set,
- Std. Dev. – standard deviation for data set,
- Sig. – significance (2-tailed) between data in the same row,
- Sig. (Hor.) – significance (2-tailed) between one horizontal (same row) data in the table (similar to Sig.),
- Sig. (Ver.) – significance (2-tailed) between two vertical data (same column) in a table, and
- Corr. – correlations between data sets.

Most of the fields in the evaluation use Likert scale ranging from 1 to 5, with 1 as strongly agree and 5 as strongly disagree. The questions are posed in positive form, meaning the option strongly agrees shows a positive response from the students. The exam results are also graded in similar way, with 1 as the best grade and 5 as failing the subject (Table 2.6).

As the data collected here are evaluated according to its semesters and there are only 6 semesters available. A simple method to identify if there is any dependence is implemented. The simplest relationship between two elements is linear relationship. This can be evaluated using Pearson correlation. Techniques of comparison for the following analysis in order to accept or disprove a hypothesis are as described in chapter 5.

6.2 Evaluation on Implementation of Bench Scale Equipment

6.2.1 Findings on Bench Scale Equipment

As mentioned in sub-chapter 4.1, one of the reasons in implementing bench scale equipment is to provide a consistent example for the students. The objective of using a consistent example through out ES2 and also group work is to help students understand the course content better. The implementation of this bench scale equipment is refined across the semesters. In ES2 SS2007, the bench scale equipment is implemented in the group work and SysML exercise. In the following ES2 SS2008, it is implemented in the whole modelling chapter using SA/RT and SysML, and the programming exercise. In ES2 SS2009, apart from using the bench scale equipment in the exercise session, the students also use the parts mentioned in the bench scale equipment to develop their own systems. A study that agrees with this idea is [CB89]. According to [CB89] students learn better when given an example, instead of only learning the theory and solving the problem. This leads to Hypothesis 1.

Hypothesis 1 – Implementation of consistent bench scale equipment will help student to understand the course content

Table 6.1 shows the student's feedback on using the same bench scale equipment for ES2 SS2009's exercises and in the group work.

Table 6.1 - Students' feedback on usability of the bench scale equipment and the pairing with the grade from the respective students (Data from ES2 SS2009)

Exam Grade	Usability of the same Bench Scale Equipment									
	1.0 (very useful)		2.0 (useful)		3.0 (neutral)		4.0 (not useful)		5.0 (not useful at all)	
	CS	M	CS	M	CS	M	CS	M	CS	M
5.0	2	1	2	0	0	0	0	0	0	0
4.0	1	0	1	0	1	0	0	0	0	0
3.0 - 3.7	2	1	1	0	2	0	0	0	1	0
2.0 - 2.7	0	1	4	1	1	0	0	0	1	0
1.0 - 1.7	2	1	1	1	1	4	1	1	0	0
No Exam	2	0	0	0	0	0	1	0	0	0
Total	***9***	***4***	***9***	***2***	***5***	***4***	***2***	***1***	***2***	***0***
GrandTotal	13		11		9		3		2	

40 students participated in the group work evaluation for ES2 SS2009. 2 Computer Sciences students did not answer to the question “the bench lab equipment helped me to understand the course content better”, and another 3 who answered the question did not participate in the exam. The ratings from the Computer Science and Mechatronics students tend to favour the implementation of bench scale equipment. The highest ratings for Computer Science students are “very useful” and “useful”, whereas the Mechatronics students rated “very useful” and “neutral” highest. 63.16% of the students agree that the bench scale equipment does help them to understand the course content better by voting for “very useful” and “useful”.

Table 6.2 presents the average grade for the students who rated the usability of the bench scale equipment. 2 students who rated “very useful” and 1 who rate “not useful” did not participate in the final exam.

Table 6.2 – Average grade for each category using WEP

Usability of the same Bench Scale Equipment	N	WEP					
		Average	Std.Dev.	Sig. To "1.0"	Sig. To "2.0"	Sig. To "3.0"	Sig. To "4.0 & 5.0"
1.0 (very useful)	11	3.22	1.47	1.000	0.491	0.058	0.198
2.0 (useful)	11	2.79	1.39	0.491	1.000	0.197	0.446
3.0 (neutral)	9	2.03	1.14	0.058	0.197	1.000	0.749
4.0 (not useful) & 5.0 (not useful at all)	4	2.25	1.05	0.198	0.446	0.749	1.000

*Legends: *“not useful” and “not useful at all” are combined as N is less than 3*

When comparing the significant differences of average grades between the different ratings, the categories “very useful” and “useful” have almost 50% probability to be the same. The students who favoured the implementation of bench scale equipment than students who viewed it neutrally or students who rated it “not useful” and “not useful at all”. The average grade difference for category “useful” is not significant as compared to other categories except for category “neutral”. Students who evaluated the bench scale equipment as “neutral” have the best average grade. This shows that even though the students rated that the bench scale equipment is useful to understand the course content better; this might not be reflected in the final exam grade.

Hypothesis 1 is acceptable because 24 out of 38 students rated it as “very useful” or “useful”. However, the understanding is not reflected in the final exam result.

Other Findings on Bench Scale Equipment:

Apart from overcoming the different cognitive mindset of the students, the other purpose of implementing the same bench scale equipment for the exercises in ES2 and also the coupled course SWT is to overcome the limited resources available for the course. The usage of bench scale equipment saved preparation time. As the same bench scale equipment is applied in both SWT and ES2, the slides prepared can be used for both courses. The slides need to be only prepared once and be used in two courses. Apart from that, no extra time is required to explain a new system before the students can proceed to solve the exercises. The initial introduction of bench scale equipment in ES2 and SWT is about 30 minutes. By using the same bench scale equipment, more time can be spent at solving more challenging questions.

6.2.2 Discussion on Bench Scale Equipment

The students who rated the bench scale equipment as "very useful" and "useful" have poorer average grade as compared to the others. Instead, students who graded the bench scale equipment as "neutral" have the best average grade. A probability might be bench scale equipment has more impact on students who "naturally" have difficulty understanding the course content. Students who are able to understand the course content on its own do not need the bench scale equipment to assist them in understanding the course content. However, this is only a proposed explanation as there are no control groups that can be compared to in this research. Even though the bench scale equipment did not influence the students' final exam grade, implementation of bench scale equipment as a consistent example through out the whole course is still being proposed. This is because majority of the students agreed that it is useful or at least did not go against this implementation. Secondly, the implementation of bench scale equipment saved preparation time and time spent on explaining the system to students.

6.3 Evaluation on Implementation of "In the Class" Methods

6.3.1 Findings on Methods Implemented in the Lecture and Exercise

As mentioned in section 2.3.2, one of the challenges faced by large class size is the attendance problem. According to [CR00], the attendance in large classes usually dwindles to 40% or 30% at the end of the semester. [DF96] mentioned that in order to combat absenteeism a learning environment that "promotes opportunities for student interaction and critical thinking" should be created. Three main methods are implemented in the lecture and exercise, namely "in class demonstration", "asking questions" and "presenting the idea visually on board". These different methods are described in section 4.2.2.

The next hypothesis will investigate if the methods implemented in the lectures and exercises are able to encourage students' attendance. The evaluation cannot be conducted for the exercise sessions, as the methods are implemented for all the six semesters. However, this is possible for the lecture session as these methods are barely implemented in ES2 SS2009 (section 4.2.2).

Hypothesis 2 – Motivating lectures encourage students' attendance

The different active learning methods implemented in the lecture, involve students' participation. The effectiveness of these implementations will be assessed using the department's course evaluation. In the department's course evaluation, there is a field requesting students to evaluate the lecture presentation. The scale given is 1 for motivating, to 5 for not motivating. This category is selected as lectures that involve students' participation (interactive lecture) should be more motivating as compared to lectures where no interaction takes place. Table 6.3 shows the average rating for lecture's presentation across 4 semesters. As the department's course evaluation is not conducted during WS0708, there is no available data for this semester.

Table 6.3 – The average rating from department's course evaluation across 4 semesters for lecture presentation

ES2	Semester	N	Average	Std. Dev.	Sig. to SS2009
Lecture Presentation Motivating – Not Motivating	SS2007	42	3.429	1.0393	0.564
	SS2008	34	2.735	0.9312	0.004
	WS0809	51	2.745	0.7961	0.006
	SS2009	20	3.600	1.1877	--

The presentation of lecture in SS2007 is not graded as well as SS2008 and WS0809. This is because SS2007 is the first semester where new course contents are being introduced (section 2.6.1). The rating for lecture presentation in the following 2 semesters improved as However the rating for SS2009 dropped to 3.600 and it is also the poorest among all the 4 semesters. The ratings between SS2009, and SS2008 and WS0809, are significantly different at 0.004 for SS2008 and 0.006 for WS0809. This shows that lecture that does not implement active learning methods are rated as less motivating.

As there is no official attendance list, the attendance in this table is compiled through the participation in pop quizzes and department course evaluation. The number of students participating in pop quizzes or evaluation may not tally with the number of students participating in the final exam. This is because students who attend class can skip the final exam if they are not ready and students who do not attend the class may take the final exam if they feel they are ready. However, the normal case is students who sit for the final exam also attend the class. Neither pop quizzes nor department's course evaluation is conducted in WS0607. Therefore, it is not possible to evaluate the average attendance percentage for WS0607. Table 6.4 shows the average students attendance across the different semesters.

Table 6.4 - Students' attendance across different semesters

Semester (Participations in Exam)	Data Source	CS	M	Average	Semester Average
SS2007 (CS=50, M=24)	Department_Course Evaluation	29	15	59%	59%
WS0708 (CS=52, M=34)	Pop quiz 1	29	21	59%	45%
	Pop quiz 2	22	16	45%	
	Pop quiz 3	11	15	31%	
SS2008 (CS=46, M=27)	Pop quiz 1	23	18	56%	58%
	Pop quiz 2	20	12	44%	
	Department_Course Evaluation	24	29	73%	
WS0809 (CS=47, M=29)	Pop quiz 1	25	15	53%	55%
	Pop quiz 2	20	12	42%	
	Department_Course Evaluation	32	21	70%	
SS2009 (CS=56, M=24)	Department_Course Evaluation	11	9	25%	25%

As shown in Table 6.4, the attendance for ES1 is averagely 49%. The average attendance for SS2007 and SS2008 is about 58%. The normal attendance for the students looks like a valley. The trend observed for the past 5 semesters shows that more students will attend the lecture in the beginning and at the end of the semester. Therefore, it can be observed that the highest percentage of attendance happens when department course evaluation is conducted. The evaluation is conducted normally during the fifth to the third last class. However, this is not reflected in the department course evaluation for SS2009.

As observed in Table 6.4, the attendance across the semesters is rather consistent except for SS2009. The class settings, course contents, and slides for ES2 SS2009 are based on ES2 SS2008. There are two differences for ES2 SS2009 as compared to ES2 SS2008. Firstly, the method for extra point, pop quizzes is implemented in ES2 SS2008 and GWB in ES2 SS2009. However, the attendance for ES2 SS2007 where no pop quizzes are implemented and GWA is implemented is not as low as 25%. Perhaps it is these situations that cause one student to comment that "the new teaching instructor does not seem to be familiar with the course content". The students start to lose interest in the class. This is being reflected in the class attendance. The attendance for SS2009 dropped drastically when compared to other semesters. There are time when only 20 out of 80 students attended lecture. This shows that the interest to attend the lecture dwindles for ES2 SS2009. From observation, the attendance for ES2 SS2009 has been around 25% to 30% during the second half of the semester.

From Table 6.3, it is observed that ES2 SS2009 has the poorest rating for motivating presentation. The attendance for ES2 SS2009 is also the poorest. Other semesters with better attendance have better motivating presentation rating. When comparing the attendance during department's course evaluation, the highest attendance is SS2008, followed by WS0809, SS2007 and finally SS2009. The rating for motivating presentation also follows the same order. Therefore, it is acceptable that motivating presentation does influence the students' attendance.

Hypothesis 2 is acceptable. Motivating lectures (by implementing in the class active learning methods) do encourage students' attendance.

According to the "cone of learning" theory, a student remembers more when they are actively involved in the learning process (section 2.5.3). Gardner's theory of multiple intelligences suggests that students with different intelligences respond to different methods (section 2.5.2). As interactive lecture includes in class demonstration with students' participations and presenting an idea on the board, this research is interested to find out if interactive lecture also influence the students' understanding on the lecture and also the demonstration presented. Hypothesis 3 will evaluate the impact of interactive lecture on the comprehensibility of lecture, whereas Hypothesis 4 will investigate the impact on perceived usefulness of demonstration presented.

Hypothesis 3 – Motivating lectures will influence students' comprehensibility of lecture

Hypothesis 4 – Motivating lectures will influence the usefulness of demonstration

In the department's evaluation form, there are two categories that can be used to analyse Hypothesis 3 and Hypothesis 4. They are "comprehensibility of explanation" and "usefulness of demonstration" in the lecture. Table 6.5 presents the average rating for these two categories. The number of students for various categories in a semester may differ as not all the students answer all the questions.

Table 6.5 - The average rating from department's course evaluation across 4 semesters for "comprehensibility of lecture" and "usefulness of demonstration"

ES2	Semester	N	Average	Std. Dev.	Sig. to SS2009
Hypothesis 3 Comprehensibility of Lecture (sufficient - insufficient)	SS2007	44	2.568	1.0432	0.086
	SS2008	35	2.457	0.8168	0.021
	WS0809	51	2.471	0.9456	0.025
	SS2009	21	3.048	1.0235	--
Hypothesis 4 Usefulness of Demonstration (effective -ineffective)	SS2007	34	2.618	1.1014	0.265
	SS2008	26	2.846	1.0077	0.652
	WS0809	40	2.500	0.9871	0.109
	SS2009	16	3.000	1.1547	--

From the course work evaluation, it can be observed in this table that the comprehensibility of lecture and the usefulness of demonstration are also at its lowest during SS2009. There are also significant differences for the ratings for comprehensibility of lecture. The difference to SS2007 is significant at p = 0.1, whereas the significant difference to SS2008 is at 0.021, and WS0809 is at 0.025. There is no significant difference for usefulness of demonstration.

Table 6.6 - The correlation between motivating presentation with understandability of the lecture and usefulness of demonstration (using Pearson's correlation)

Motivating Presentation	Hypothesis 3 Comprehensibility of Lecture (sufficient - insufficient)		Hypothesis 4 Usefulness of Demonstration (effective -ineffective)	
CS and M	Correlations	Sig.	Correlations	Sig.
SS2007	0.459	0.002	0.337	0.055
SS2008	0.275	0.115	0.286	0.157
WS0809	0.561	0.000	0.462	0.003
SS2009	0.523	0.018	0.542	0.030

Based on the Pearson correlation for the three semesters in Table 6.6, there is a positive correlation between motivating presentation and comprehensibility of the lecture. From the four semesters, only the correlations for SS2008 are not significant. Therefore, it is acceptable that motivating presentation do influence comprehensibility of lecture and the perceived usefulness of demonstration.

Table 6.7 presents the correlation of these three factors according to the disciplines. This will provide a more detailed view if the opinions between the two disciplines are different.

Table 6.7 - The correlation between motivating presentation with understandability of the lecture and usefulness of demonstration according to the disciplines (using Pearson's correlation)

Motivating Presentation	Hypothesis 3 Comprehensibility of Lecture				Hypothesis 4 Usefulness of Demonstration			
	Computer Science		Mechatronics		Computer Science		Mechatronics	
	Corr.	Sig.	Corr.	Sig.	Corr.	Sig.	Corr.	Sig.
SS2007	0.491	0.009	0.352	0.199	0.405	0.069	0.330	0.295
SS2008	-0.262	0.346	0.593	0.007	0.200	0.579	0.414	0.111
WS0809	0.631	0.000	0.491	0.028	0.272	0.178	0.788	0.001
SS2009	0.452	0.189	0.401	0.284	0.866	0.012	0.513	0.193

Except for SS2008, the Computer Science students have positive correlations between motivating presentation and comprehensibility of lecture. The correlations for SS2007 and WS0809 are significant. The Mechatronics students have positive correlations between motivating presentation and comprehensibility of lecture for all the four semesters. The significance for the correlations in SS2008 and WS0809 are strong at 0.007 for SS2008, and 0.028 for WS0809.

For the correlations between motivating presentation and the perceived usefulness of demonstration, Computer Science students have positive correlations for all the four semesters. The correlation during SS2009 at 0.012 is significant. The correlations for Mechatronics students are also positive for all the semesters. The correlation during WS0809 is most significant at 0.001; the correlation is also strong at 0.788.

Other than positive correlations for almost all the correlations, no specific trends can be observed for Computer Science and Mechatronics students. However, when the data are observed without distinguishing the disciplines as in Table 6.5, significant correlations between motivating presentation and comprehensibility of lecture, as well as motivating presentation and usefulness of demonstration can be observed.

Hypothesis 3 is acceptable. 3 out of 4 semesters have significant correlation between motivating lecture and comprehensibility of lecture

Hypothesis 4 is acceptable. 3 out of 4 semesters have significant correlation between motivating lecture and usefulness of demonstration

Heuristics Findings on Visual Aid Used in the Class:

Another factor worth mentioning here is the slides used in the class. The students who participated in ES1 and ES2 commented that there are “too many slides”. 6 from 15 students who gave feedback on ES2 SS2007 mentioned that there are “too many slides” and “the scope is too big”. As comparison, other courses in the faculty provide lecture notes and as a result have simpler lecture slides. As there are no prepared lecture notes for ES1 and ES2, the slides have the function to provide the additional related information that the students need. Efforts to reorganize the slides took place in ES2 SS2008. Duplicated slides for references are removed from the uploaded copy for the students and sub-chapters are added to group the slides. However, the students still find the slides and the course content overwhelming. In the evaluation for ES2 SS2008, the percentage of students who commented and complained about the slide was reduced to 26.3% as compared to 37.5% in ES2 SS2007, a reduction of 11.2%. During ES2 SS2009, the same slide from ES2 SS2008 slides but without the in-between chapter slides are used. 4 students out of 20 responds, which is 20%, commented that there are too many slides. More time consuming suggestions are preparing lecture notes for the students. Even if lecture notes are not being prepared, the slides for the class should be kept simple, less information and more explanations [Ca98]. Slides that serve as references should be uploaded separately from the presentation slides use for the lecture.

6.3.2 Discussion on Methods Implemented in the Lecture and Exercise

ES1 and ES2 are being conveyed through lectures and exercises. The students do not know the schedule for the lecture and exercise. However, the lecture normally alternates with the exercise. The slides for both the exercise and lecture mainly remain the same through out the six semesters. However, the methods implemented through out the different semesters are different.

In order to overcome the problems that comes with large class size. Different active learning methods are implemented. Among the problems with large class size is dwindling attendance from students towards the end of the end of the semester. This may due to the lack of interaction in the class (section 2.3.2). Motivating lecture is implemented to overcome large class size problem. Using the different active learning methods as described in sub-chapter 4.2, one can increase the students' engagement in the class. The time required to involve students' participations may depend on the creativity of the teaching instructor. However, there are also methods like cold calling that do not need any preparation at all. This is a good method that should be implemented by the teaching instructors for large class environment. By implementing active learning methods in the lecture, the presentation is more motivating.

Hypothesis 2 affirmed that motivating lecture will encourage students' attendance. One may argue that the course content is unfamiliar to the new teaching instructor who taught ES2 SS2009. Therefore, the students are not interested to attend the class. This can be easily dismissed as the course content is also new during ES2 SS2007. However, during this semester, the students' attendance is still at the average 49%. The uncertainties with course content may cause the motivating presentation to be rated poorer as in ES2 SS2007 and ES2 SS2009. This may due to unfamiliar slides, or the yet to be corrected mistakes on the slides. However, when motivating lecture is not implemented, the students will lose interest in attending the class.

By implementing active learning in class, students may understand the course content better. This is demonstrated by the significant positive correlation between motivating lecture and comprehensibility of the lecture. The demonstrations in the class are also perceived as more useful when motivating lecture is implemented.

6.3.3 Findings on Implementations of Pop Quizzes

One of the problems faced by large class size is the attendance problem. As mentioned in section 2.3.2, the class size dwindles at the end of the semester. [DF96] mentioned that incentives to attend class can be provided by having short quizzes at the beginning of every class, As pop quizzes enable the students to collect extra 15% points for the final exam, it is expected that this motivation will encourage students' attendance. Pop quizzes in ES1 and ES2 is held randomly (section 4.2.3). By not making known when pop quizzes will be held, students need to attend the class regularly to take part in pop quizzes. This leads to Hypothesis 5, mentioning that pop quizzes will motivate students' attendance.

Hypothesis 5 – Pop quizzes will motivate students' attendance as compared to semesters without pop quizzes

As shown in Table 6.4, the average percentage for attendance when pop quizzes are implemented is 45% during WS0708, 58% during ES2 SS2008, and 53% during ES1 WS0809. The average attendance confirms the findings of other research. [CR00] mentioned that in large classes the attendance usually drops to 30% to 40% at the end of the semester, there are also classes with 50% attendance at the end of the semester. The attendance when pop quizzes are conducted lies at the upper percentage of attendance, which is between 45% and 58%. However, during ES2 SS2007, where no pop quizzes are implemented, the attendance at the end of the course is 59%. The lowest attendance percentage when pop quizzes are implemented is 31% during the third pop quiz in ES1 WS0708. The implementation of pop quizzes did not encourage a high increase, for example 80%, of attendance from the students. Therefore, it is not evident that pop quizzes motivate the student's attendance.

Hypothesis 5 is not acceptable. Pop quizzes do not motivate students' attendance as compared to semesters without pop quizzes.

According to various research, the exam grade deteriorates when the class size is bigger [Ku07], [AW04], [GL+96]. [DF96] mentioned that one possible way to overcome this problem is by providing practise test to the students.

As mentioned in section 4.2.3, pop quizzes are surprise test conducted through out the semesters. Pop quizzes serves as practise test for the students, and also help students to focus on the core of the subjects instead of getting lost in the numerous details. It is expected that pop quizzes will benefit the students and this will be reflected in the final exam. Hypothesis 6 will investigate if pop quizzes do prepare the students to answer the questions in the final exam.

Hypothesis 6 – Pop quizzes will prepare the students for the final exam

Comparison between Pop Quizzes' Participants and Control Data:

Table 6.8 shows the average grade comparison for pop quizzes participants as compared to average control grade from the control students. As the courses from control data also includes extra points from different class activities, average grade WEP will be used here.

Table 6.8 - Average grade comparison for pop quizzes participants with control students

Pop Quizzes			Participants (WEP)				Control Data				Sig. (Hor.)
			N	Average	Std. Dev.	Sig. (Ver.)	N	Average	Std. Dev.	Sig. (Ver.)	
CS	ES1	WS0708	27	3.052	1.0184	0.004	258	3.322	1.2917	0.804	0.294
	ES2	SS2008	26	2.162	1.1486		147	3.354	1.1637		0.000
	ES1	WS0809	29	2.638	1.4115	--	108	3.392	1.3670	--	0.010
M	ES1	WS0708	20	2.260	1.0748	0.360	140	2.800	1.1000	0.239	0.041
	ES2	SS2008	20	1.915	1.2717		149	2.951	1.0771		0.000
	ES1	WS0809	15	1.973	1.3019	--	173	3.293	1.3332	--	0.000

Pop quizzes are first implemented in ES1 WS0708. During this semester, the average grade for the Computer Science students and the Mechatronics students are better than the average control grade, but the difference is not significant. However, for the following semester ES2 SS2008, the average grade for both Computer Science and Mechatronics students are significantly better than the average control grade. In fact, the average control grades are poorer during ES2 SS2008 as compared to the semester before. However, the opposite is true for average grade. The improvement is significant for the Computer Science students. During the implementation in ES1 WS0809, the Computer Science performed better than the average control grade, whereas the Mechatronics students' average grade is significantly better than the average control grade.

The impact of pop quizzes during the implementation in ES1 does not show significant improvement as compared to average control grade for the Computer Science students. The Mechatronics students did significantly better during the implementation in ES1 WS0809, but only better during the implementation in ES1 WS0708. However, the implementations in ES2 SS2008 have significantly better average grade as compared to average control grade for both Computer Science and Mechatronics students. When comparing the improvement within a batch for WS0708 and SS2008, the improvement for Computer Science participants is the most significant at 0.004. This is followed by the Mechatronics participants at 0.175. The improvements for participants in pop quizzes from ES1 WS0708 to ES2 SS2008 are better than the control data. This shows that the impact of pop quizzes when only implemented in ES1 is not conclusive. However, when it is being implemented in two consecutives semesters, then the result is better.

Comparison between Pop Quizzes' Participants and Non-Participants:

In order to analyse the more "detailed" impact of pop quizzes, comparison between students who took part with pop quizzes and those did not will be done. For a fair comparison, NEP average grade will be used here. Therefore, the number of participants will be the same as in Table 6.8, but the average grade will be different, as Table 6.8 uses WEP grade. Table 6.9 compares the participants and the non participants in pop quizzes.

Table 6.9 - Average grade comparison for participants and non participants in pop quizzes

Pop Quizzes			Participants (NEP)				Non-Participants (NEP)				Sig. (Hor.)
			N	Avg.	Std. Dev.	Sig. (Ver.)	N	Avg.	Std. Dev.	Sig. (Ver.)	
CS	ES1	WS0708	27	3.248	1.0024	0.096	25	3.784	0.8601	0.290	0.045
	ES2	SS2008	26	2.742	1.1673		20	4.070	0.9251		0.000
	ES1	WS0809	29	3.069	1.3696	--	18	3.506	1.2027	--	0.272
M	ES1	WS0708	20	2.535	0.9366	0.704	14	4.193	0.7205	0.360	0.000
	ES2	SS2008	20	2.400	1.2703		7	3.857	0.8772		0.010
	ES1	WS0809	15	2.320	1.4047	--	14	4.114	0.9281	--	0.000

The average grades for pop quizzes participants are better than the non-participants for all the semesters. The average grade improvement for Computer Science students between ES1 WS0708 and ES2 SS2008 is 0.506, which according to Table 2.6 is almost equivalent to 2 levels of grading, and the difference is significant at $p = 0.1$. The non-participants' average grade on the other hand declined. The average grade improvement for Mechatronics students between ES1 WS0708 and ES2 SS2008 is 0.135, and is not significant (0.704). Even though the improvement for non-participants batch is also not significant (0.360), but it is better than improvement of the participants. This suggests that the impact of consecutive implementation of pop quizzes is bigger for Computer Science students as compared to the Mechatronics students.

The implementation in ES1 WS0809 has positive impact for the Computer Science students, but the improvement is insignificant (0.272) for the Computer Science students. The difference for the Mechatronics students is significant (0.000). Pop quizzes participants have average grade 2.320 and non participants 4.114.

The results above shows that the participation of pop quizzes positively influence the exam results. The theory here is by doing pop quizzes, they have more "practise test" and therefore, will be able to do better in the "real exam" or the final exam. If this is the case, then the number of participations should also influence their grade, and according to the "practise test" concept, it should be the more the better. Table 6.10 presents the correlation of NEP and WEP average grade with the number of participation in pop quizzes.

Table 6.10 – Correlations between number of time in pop quizzes participation and average grade (using Pearson's correlation)

		NEP	WEP
Number of Times Participating Pop Quizzes	Correlation	-0.305	-0.351
	Sig. (2-tailed)	0.000	0.000
	N	142	142

The number of times a student participate in pop quizzes correlates with the NEP and WEP average grades. The participations increase in nominal number (1, 2, or 3 times); whereas the smaller the grade (for example 1.0) is considered a better grade than a bigger number (for example 4.0). There is a negative correlation, meaning the more often a student participates in pop quizzes, the better the grade will be. Therefore, it can be concluded that the participation in pop quizzes will positively influence the students' final exam result.

Hypothesis 6 is acceptable. Students who participated in pop quizzes do better in the final exam.

6.3.4 Discussion Implementation of Pop Quizzes

The implementation of pop quizzes is to overcome the attendance problem in large class size and to provide "practise test" for the students. From the analysis above, it is not evident that the implementation of pop quizzes encouraged students' attendance. Even though, the percentage of attendance for the three semesters where pop quizzes are implemented is about 50%, the attendance when pop quiz is not implemented varies too much between 59% for SS2007 and 25% for SS2009. More specific responds from the students are needed to draw a strong conclusion if pop quizzes do affect the class attendance.

The implementation of pop quizzes influenced the students' grade positively. Students who participated in pop quizzes have better average grade when compared to the average control grade from control data, and average grade from non-participants. With pop quizzes, the students prepare themselves for the coming written examination. The students are able to estimate and be familiarised with the pattern of questions. As mentioned in 4.2.3, the degree of difficult for pop quiz will increase within the semester. As the time allocated for pop quizzes are at most 20 minutes each time, most of the questions designed are based on knowledge, comprehension and application. These are the first three levels in AK's taxonomy. The questions in the final exam include these three levels and the fourth level that is analysis. One of the reasons why students who participated in pop quizzes do better than those did not in the final exam might be that they have the opportunity to practise at least three different levels of questions in the pop quizzes. This also helps the teaching instructor to convey what are the important aspects in the course and the students can revise the course work accordingly. Apart from this, receiving the result for each pop quizzes serves as a reminder that the students should be on their toes in revising the course content. This serves as a further motivation for them to do better in the next pop quiz or exam.

There are other suggestions why students who participated in pop quizzes will do better than those did not. One of the suggestions is students who attend pop quizzes are normally students who attend the classes. Students who did not participate in the pop quizzes have higher absenteeism. However, as the evaluation conducted in this research is anonymous, this point cannot be deduced within this research.

6.3.5 Findings on Implementation of 5IMPLe

In order to overcome different cognitive mind set and large class sizes, 5IMPLe is implemented in ES1 WS0809 and ES2 SS2009. 5IMPLe as described in section 4.2.4 is feedback card/paper to know what the students understand, and the subjects that need more revision. It is a variant of the muddy card concept. Using this feedback, the exercises are tailored. It is hope that this will encourage the students to attend the class as it meets their personal needs. Table 6.11 shows the responses of 5IMPLe through out the two implementations. As mentioned in section 4.2.4, 5IMPLe is only conducted once in winter semester 2008/2009 and twice in summer semester 2009.

Hypothesis 7 – Course feedback using 5IMPLe will motivate the students to provide feedback concerning the course content

Table 6.11 - 5IMPLe response as compared to the course attendance

5IMPLe Implementation (Number of Students in Final Exam)	N of response	N of meaningful response	% of meaningful response	% of meaningful response to class size
WS0809 (CS=47, M=29)	26	26	100%	34.21%
SS2009 (CS=56, M=24)	17	15	88%	18.75%
	8	5	63%	6.25%

Table 6.11 shows that the response for the first implementation is good. 15 out of the 17 feedback forms are meaningful. The 2 not meaningful feedbacks are “why would I know?”, and “everything”. For the second 5IMPLe implementation, only 5 from the 8 feedbacks are meaningful. The not meaningful feedbacks are “a question mark”, “everything” and the remark “exam relevant information”.

Even though 5IMPLe is implemented in both semesters, it is observed that the response for SS2009 is poorer, especially during the second implementation. The percentage of meaningful response decrease over the implementations, the first attempt in WS0809 is the best, and the second attempt in SS2009 is the poorest. One possible reason is, as proposed by Hypothesis 2, the students are losing interest in the class. The implementation of 5IMPLe alone is not able to motivate students to provide feedback concerning the course content.

Hypothesis 7 is not acceptable. The percentage of meaningful response for 5IMPLe decreases over implementations.

6.3.6 Discussions on Implementation of 5IMPLe

The effectiveness of muddy card is not prominent in this research. Higher frequency of implementation is needed to observe the actual impact. Muddy card received good feedback from [KN02]. As compared to the implementation of 5IMPLe, the muddy card implementations by [KN02] and [HW+02] are conducted at higher frequency. They are conducted almost at the end of every class. Students only have 1 to 2 minutes to write their opinion on an index card. The size of the index card also limits the students to write only a few sentences or key words. As 5IMPLe is not conducted that frequently, the scope for each 5IMPLe is also larger. Students might not necessarily still remember what was being taught in the previous chapter. For example, during the implementation of 5IMPLe in SS2009, only 9 from the 17 students mentioned on what is important during the first round, and 1 from the 8 students during the second round. [HW+02] who implanted a few rounds or "muddy cards" with the students mentioned that "*comments that reflect students' perceived value of muddy cards declined in the subsequent year … we believe that as other active learning techniques were implemented more effectively … the muddy cards were less essential to students as a form of feedback*." This is almost similar with the situation observed in ES1 and ES2. The first implementation has higher response as compared to the following. However, as there are only 3 implementations for both ES1 and ES2, no strong conclusion on 5IMPLe can be drawn.

6.4 Evaluation on Implementation of "Partially Outside the Class" Methods

6.4.1 Findings on Implementation of Group Work

One of the problems faced by ES1 and ES2 as an interdisciplinary course is the presentation of interdisciplinary subject to a group of interdisciplinary students. Section 2.3.1 mentioned that interdisciplinary classes have students with different cognitive mindset. The two groups of students here are Computer Science students and Mechatronics students, according to [WT+01] the courses for Mechatronics students should be *"problem- based, product-design oriented, and project-team organised"*. Apart from this, according to the "cone of learning" theory presented in section 2.5.3, students will remember 90% of what they learnt if they practically do what they learnt. Therefore, it is expected that the implementation of group work will help the students to understand the course content better. This will then be reflected in the final exam grade for the students. These lead to the following 2 hypotheses. As there are two types of group work in ES2 (sub-chapter 4.3), Hypothesis 8 will discuss the implementation of GWA and Hypothesis 9 on the implementation of GWB.

Hypothesis 8 – GWA participants will have better average grade than non-participants

Hypothesis 9 – GWB participants will have better average grade than non-participants

Comparison between Group Work Participants and Control Data:

Table 6.12 compares the average grade for group work participants to average control grade from the control data.

Table 6.12 - Average grade comparison for group work participants with control students

Group Work (Semester Implemented)		Participants (WEP)			Control Data			Sig.
		N	Average	Std. Dev.	N	Average	Std. Dev.	
GWA ES2 SS2007	CS	41	2.593	0.8748	139	2.900	1.0476	0.089
	M	19	3.158	0.8821	91	2.924	1.1261	0.397
GWB ES2 SS2009	CS	34	2.824	1.1855	129	3.468	1.3187	0.011
	M	17	1.724	1.0533	123	2.493	1.4392	0.013

Group work was conducted for the first time in a large class environment during ES2 SS2007. This is known as GWA. The Computer Science students have better average grade than the average control grade. The difference is 0.307 and it is significant at $p = 0.1$. The Mechatronics students on the other hand, have poorer average grade than the average control grade, but the difference is not significant. The analysis suggests that GWA seems to benefit Computer Science students and no significant impact on the Mechatronics students.

The second implementation of group work, which is GWB, has more significant impact than GWA. Both Computer Science and Mechatronics students who participated in GWB have better average grade as compared to the average control grade from control data and the differences are significant. The comparison with control data shows that implementation of GWB helps both Computer Science and Mechatronics participants to rise above the performance from other courses in the semester.

Comparison between Group Work Participants and Non-Participants:

In order to have a more detailed view on the impact of group work within the same class of students, comparison between participants and non-participants are conducted here. NEP average grade, which is the average grade without considering the extra 15% possible marks from group work, is being used here. Table 6.13 compares the average grade for group work participants with the average grade from non-participants.

Table 6.13 - Average grade comparison for participants and non participants in group work

Group Work (Semester Implemented)		Participants (NEP)			Non-Participants (NEP)			Sig.
		N	Average	Std. Dev.	N	Average	Std. Dev.	
GWA ES2 SS2007	CS	41	3.029	0.8878	9	4.144	0.6966	0.001
	M	19	3.653	0.9845	5	2.660	1.1082	0.063
GWB ES2 SS2009	CS	34	3.659	1.0252	22	2.891	1.3140	0.018
	M	17	2.394	1.1448	7	3.571	1.5119	0.049

The Computer Science students who participated in GWA have significantly better average grade than non-participants, the difference is 1.115. However, the average grade from the Mechatronics who participated is poorer than the non-participants, the difference is at 0.993. Even though, the difference is not significant, but this is the opposite observation as compared to the Computer Science students. It is possible that GWA is more suitable for the Computer Science students as compared to the Mechatronics students.

The opposite trend from GWA happens in GWB. The Computer Science students who participated in GWB have significantly poorer average grade than non-participants. However, the average grade for Mechatronics students who participated in GWB is significantly better than non-participants. It is possible that GWB is more suitable for the Mechatronics students as compared to the Computer Science students.

From the comparison with control data, it is observed that GWA benefited the Computer Science students but the impact on Mechatronics students is not evident. However, through the comparison with non-participants, the non-participants have significantly better average grade the participants. Therefore, Hypothesis 8 is not totally acceptable.

Hypothesis 8 is not totally acceptable. The implementation of GWA benefited the Computer Science students significantly but not the Mechatronics students.

When comparing to the control data, GWB participants have better (WEP) average grade. When using NEP average grade for comparison, the Mechatronics participants have better average grade than non-participants, but this is not true for Computer Science student. The difference is caused by the extra points for the Computer Science students. As the participants and non-participants both participated in ES2, the comparison is closer to the context of implementation. Therefore, Hypothesis 9 is not totally acceptable.

Hypothesis 9 is not totally acceptable. The implementation of GWB benefited the Mechatronics students significantly but no significant positive impact can be concluded for the Computer Science students.

The purpose of group work implementation is to provide practical work opportunity as proposed by “cone of learning” theory in section 2.5.3. In GWB, the students have the opportunity to work in more than 1 phase of the lifecycle. Through participation in more than 1 phase of the lifecycle, the students might have more “practical work”. Hypothesis 10 will investigate if students who participated in more than 1 phase of the lifecycle have better grade.

Hypothesis 10 – Students who participated in more than 1 lifecycle phase will have better grade than those who only participated in 1 lifecycle phase.

The extra points for group work are awarded based on the whole group work. This includes contribution from different members. In order to evaluate the impact on individual members, NEP average grade that does not consider the contribution from other members will used for this analysis. Table 6.14 shows the comparison of average grade between participants who participated in more than 1 phase of lifecycle to those who only worked in 1 phase.

Table 6.14 – Comparison between GWB participants who worked only in a phase and more than a phase

GWB Participants	Single Phase (NEP)			More than 1 Phase (NEP)			Significance
	N	Average	Std. Dev.	N	Average	Std. Dev.	
CS	24	3.708	1.0434	10	3.54	1.0244	0.669
M	12	2.583	1.2074	5	1.94	0.9317	0.306

Out of the 65 students who participated in GWB, only 51 students took part in the final exam. 36 students participated in only one phase and 15 students participated in more than one phase of the lifecycle. The students who participated in more than one phase have better grade than those in single phase. The difference for Mechatronics students is bigger than Computer Science students. However, the difference for both Computer Science and Mechatronics students are not significant. Therefore, Hypothesis 10 is not conclusively true.

Hypothesis 10 is not totally acceptable. There is a tendency for participants in more than 1 lifecycle phase to have better average grade. However, the difference is not significant.

Heuristics Findings on Peer-Influence:

In class presentation conducted in GWA is similar to some of the techniques applied in peer-education [GG76]. Among the benefits of peer-education is the "student-tutor" would learn more in the process of preparation, and the presentation is "closer" to the students. This might be a motivating factor for students to present before their peers. GWA requires the students to present the group finding during exercise session. Unlike GWA, GWB does not require the students to present in front of their peers. From observation, students in GWA are more serious than GWB. The students were more prepared to present during GWA, whereas in GWB, 3 from the 9 groups did not prepare all the requested presentations. The very fact that students can make their own appointments, and yet are not prepared for the presentation shows the lack of commitment

6.4.2 Discussions on Implementation of Group Work

The implementation of group work is to overcome the different cognitive mindset between the students. Through group work students also able to have practical work that is "supposed to" help them remember and understand the course content better. GWA and

GWB have different impacts on the different disciplines. GWA have significant positive impact on the Computer Science students and GWB on the Mechatronics. These may due to the effort required by GWA and GWB.

From the informal interview with Computer Science students, one reason for not participating in GWB is the high effort required. The Computer Science students prefer to use the time to study and revise on their own, instead of working in the group work and has the possibility to add another 15% extra point to the final exam. [GH05] mentioned that Mechatronics courses have emphasis in practical work. This is also reflected in the participation of Mechatronics students in GWA and GWB. The Computer Science students, participation in GWB dropped to 61% as compared to 82% in GWA. Unlike the Computer Science, the participations from Mechatronics students did not drop so much with the increase of effort required. The participation of Mechatronics students in GWA is 79% and in GWB is 70%. The drop is only 9% as compared to 21% by Computer Science students. Many Mechatronics courses include practical works (Table 2.3); therefore, even though GWB requires more effort, they are ready for it. According to the department’s course evaluation, the number of hours spent by Mechatronics students on GWB in ES2 SS2009 almost doubles the hours spent for pop quizzes in ES1 WS0809. However, the numbers of hours spent by Computer Science students are almost the same for both semesters. The Computer Science students “seemed” not ready to put in the extra effort required. The “effort for practical work” seems to be the key to the difference between Computer Science and Mechatronics students. With more “practical work” elements the Mechatronics students tend to fare better, but the opposite is true for Computer Science students.

According to “cone of learning” theory, practical work will help students to remember better. The purpose of GWB is to allow the students to follow through each phase of the life cycle and thus getting more “practical work”. However, this did not work as expected. GWB allows the students to determine the job distribution. The intended impact for GWB, where the students can follow through the whole lifecycle development of an embedded system was not totally achieved. Only 29.4% of participants participated in more than 1 lifecycle phase. The students who participated in more than one phase of lifecycle have better average grade than those who participate in only a single phase. 3 students commented in the group work evaluation that they do not know what is happening in other phase as they only worked in a particular phase. Their proposal is to have smaller groups so that one student can cover more phases in the lifecycle. The impact of GWB might have been more positive if more students participate in more than phase of lifecycle development.

Non Hypothesis Related Discussions for Group Work:

The problem with group size is also being highlighted in the group work evaluation for GWB. Table 6.15 shows the number of respondent to the strength and opportunities of GWB.

Table 6.15 – Students feedback on group work in ES2 SS2009

What can be improved for the group work		What do you like best about the group work	
Respondent= 31, Non Respondent=9	N	Respondent= 28, Non Respondent=12	N
Smaller group size	13	Team work	20
Co-ordination within the group work	8	Understand the course content better	7
All is good	5	Practical Work	3
Smaller scope for group work	4	Extra point for the exam	2
When possible, participate in all phases	3	Nothing	2
Prefer pop quizzes	3		
More weight in exam	2		

The number of response may be more than the number of respondent as the subjective response can be categorise into multiple categories. The biggest problem faced by the students is the group size. Students in GWB prefer to have smaller groups and to be able to work through all the phases in the lifecycle. 13 students prefer to have smaller group size, 8 students mentioned that the group work's co-ordination needs to be improved. The desired group size is between 4 and 5 person a group. However, this would mean more appointments and more consultation hours. The proposed group size is between 5 and 6; whereas group size of 2 or 3 is too small for positive discussion [Sm96]. The students in GWA did not specify any preference for a smaller group size. This might due to lower co-ordination effort required in GWA, as GWA only needs to work on a single phase and not a real system development.

The impact of GWA and GWB will not necessarily be reflected in the exam. The side-benefits of group work, for example team work, are not evaluated in the final exam. From the 40 students who participated in the group work review, 20 students learnt about the importance of team work, followed by 7 saying that the group work helps them to understand the course content better. The exam questions are a combination of theoretical and application question. Even though the students did apply the theory in the group work, they might not score well as they fail to interpret the exam questions correctly. Other positive effects of the group work are not measured in the final exam.

The impact for group work might be more significant when the weight given for group work in the final exam is higher. Efforts for group work can be managed by focusing on the journey instead of the result. Soft-skills development through group work should also be taken into consideration. Most of the students realised that co-ordination work is important and apart from technical skills, communication skill is also prized.

6.5 Evaluation on Implementation of Methods "Outside the Class"

6.5.1 Findings on Course Coupling between ES2 and SWT

As proposed by "cone of learning" theory in section 2.5.3, practical works will assist the learning process of students. Sub-chapter 4.4 described that by coupling ES2 with SWT, practical works opportunity can be offered to the students The first attempt was loosely coupled SWT. The implementations in loosely coupled SWT are modified for the following semesters (section 4.4.3). The implementations after these changes are known as closely coupled SWT. Therefore, Hypothesis 11 proposes that closely coupled SWT will have more positive impact as compared to loosely coupled SWT.

Hypothesis 11 – Closely coupled software tools course (CC SWT) has more positive impact as compared to loosely coupled software tools (LC SWT)

Comparison between Software Tools Participants and Control Data:

Table 6.16 compares the average grade for participants in SWT as compared to the average control grade from the control data. WEP average grade will be used here. No

Table 6.16 - Average grade comparison for LC SWT and CC SWT with control students

Software Tools		Participants (WEP)			Control Data			Sig.
		N	Avg.	Std. Dev.	N	Avg.	Std. Dev.	
LC SWT SS2007	CS	10	2.700	0.7318	139	2.900	1.0476	0.554
	M	22	2.968	0.9245	91	2.924	1.1261	0.865
CC SWT SS2008	CS	2	3.000	2.8284	147	3.354	1.1637	--
	M	24	2.221	1.4213	149	2.951	1.0771	0.004
CC SWT SS2009	CS	7	3.186	1.4265	129	3.468	1.3187	0.583
	M	22	2.014	1.2353	123	2.493	1.4392	0.144

The Computer Science students did better than the control data for LC SWT, whereas the Mechatronics did poorer, but the difference is not significant for both. For the implementation of CC SWT SS2008, comparison will be conducted for Computer Science students in SS2008 as there are only 2 participants. The Mechatronics students on the other hand did significantly better than the control data. The Computer Science and Mechatronics students did better than the average grade in LC SWT SS2009. However, the differences are not significant for both. 2 out of the 7 Computer Science students who participated in CC SWT SS2009 failed ES2. When the grades from these two students are excluded, the average grade from the other five students totals to 2.460.

The implementation of LC SWT seems to benefit the Computer Science students but not the Mechatronics students. The implementation of CC SWT on the other hand seems to benefit both disciplines. From the data, it can be concluded that CC SWT is more effective than LC SWT for the Mechatronics students.

Comparison between Software Tools Participants and Non-Participants:

In order to observe the direct impact of SWT more clearly, ES2 average grade between participants and non-participants will be compared. Table 6.17 presents the average grade comparison for participants and non-participants in SWT.

Table 6.17 - Average grade comparison for participants and non participants in LC SWT and CC SWT

Software Tools		Participants (NEP)			Non-Participants (NEP)			Sig.
		N	Avg.	Std. Dev.	N	Avg.	Std. Dev.	
LC SWT SS2007	CS	10	3.200	0.7409	40	3.238	1.0081	0.913
	M	22	3.350	1.0541	2	4.500	0.7071	--
CC SWT SS2008	CS	2	3.150	2.6163	44	3.327	1.2138	--
	M	24	2.583	1.2761	3	4.333	0.5774	0.029
CC SWT SS2009	CS	7	3.857	1.0830	49	3.286	1.2057	0.241
	M	22	2.532	1.2029	2	5.000	0.0000	--

The comparison between participants and non-participants is not entirely possible for all the semesters. Due to poor participation from the Computer Science students and that almost all the Mechatronics students took part in SWT valid comparison for all the categories is not possible.

The participation of Computer Science students in both LC SWT and CC SWT are low. Only 20% (10 out of 50 students) participated in SS2007, 4.3% (2 out of 46 students) in SS2008, and 12.5% (7 out of 56 students) in SS2009. The average grade of participants in LC SWT SS2007 but the opposite is true for CC SWT SS2009. However, the differences are not significant. The participation from the Mechatronics students is higher. 91.7% participated in SS2007, 88.9% in SS2008 and 91.7% in SS2009. The average grades for participants are better than non-participants. The difference during CC SWT SS2008 is significant.

As the comparison between participants and non-participants is not able to provide much information, another possible comparison is the grade improvement/decline within the batch. For this evaluation the average grade in ES1 of an individual student will be compared to his/her grade in ES2. The difference between the average grade in ES1 and ES2 might be able to provide some information to see a clearer picture. Table 6.18 shows the one to one comparison for participants in software tool as to the average control grade from the control data. WEP grade will be used here.

For the batch WS0607-SS2007, where LC SWT is implemented during SS2007, the improvement for Computer Sciences participants is not significant, whereas the control data has significant average control grade improvement. The Mechatronics participants improvement is significant at $p = 0.1$, whereas the improvement for the control data is not significant. This leads to the conclusion that LC SWT impact on Computer Science participants is not evident. However, LC SWT has positive impact on Mechatronics participants.

Table 6.18 - One to one comparison for software tool's impact as compared to the grade in the previous semester for ES1

		Participants (WEP)				Control Data			
		N	Avg.	Std. Dev.	Sig. (Ver.)	N	Avg.	Std. Dev.	Sig. (Ver.)
CS	ES1 WS0607	8	2.525	1.3285	0.963	298	3.436	1.2232	0.000
	LC SWT SS2007	8	2.500	0.6698		139	2.900	1.0476	
	ES1 WS0708	1	3.300	0.0000	--	258	3.322	1.2917	0.804
	CC SWT SS2008	1	1.000	0.0000		147	3.354	1.1637	
	ES1 WS0809	5	3.000	1.5811	0.825	108	3.392	1.3670	0.662
	CC SWT SS2009	5	3.200	1.1576		129	3.468	1.3187	
M	ES1 WS0607	16	3.662	1.3446	0.077	78	3.082	1.1320	0.366
	LC SWT SS2007	16	2.894	1.0070		91	2.924	1.1261	
	ES1 WS0708	21	2.405	1.1698	0.284	140	2.800	1.1000	0.239
	CC SWT SS2008	21	1.986	1.3275		149	2.951	1.0771	
	ES1 WS0809	20	2.345	1.3189	0.169	173	3.293	1.3332	0.000
	CC SWT SS2009	20	1.815	1.0604		123	2.493	1.4392	

CC SWT is implemented twice, in SS2008 and SS2009. Firstly, the participations from Computer Science students will be analysed. For the batch WS0708-SS2008, there are not enough participants to analyse the impact on Computer Science participants. For the batch WS0809-SS2009, both Computer Science participants and the control data have poorer average grade during SS2009 as compared to WS0809. However, the decline for participants is less significant (0.825) than the control data (0.662). CC SWT did benefit the Computer Science participants.

Secondly, the impact of CC SWT on Mechatronics students will be analysed. The average grade for Mechatronics participants improved for both batches WS0708-SS2008 and WS0809-SS2009. The opposite is true for the average control grade from the control data for batch WS0708-SS2008. The average control grade declined from 2.800 in WS0708 to 2.951 in SS2008. This shows that CC SWT has significant positive impact on the Mechatronics students.

Using the comparison with control data, LC SWT SS2007 has the tendency to benefit the Computer Science students. However, from the one to one comparison the results of participants improved but it is not significant. The impact of LCSWT SS2007 on Mechatronics students is positive but not significant.

The implementation of CC SWT is positive for both Computer Science and Mechatronics students when compared to the control data. The one to one comparison shows bigger impact. In the semesters where there is a decline within the batch; participants in SWT have improvement of grade. Therefore, CC SWT is better than LC SWT

Hypothesis 11 is not totally acceptable. The impact of CC SWT as compared to LC SWT is better but not significant for Computer Science students and significantly better for Mechatronics students.

6.5.2 Discussion on Course Coupling between ES2 and SWT

The purpose of coupling ES2 with SWT is to provide practical opportunities to the students. The improvements implemented in CC SWT brought positive changes. From the data analysis, it is observed that CC SWT is better than LC SWT. These improvements are also supported by the comments in the department's evaluation. Even though more co-ordination between took place in CC SWT SS2008 as compared to LC SWT SS2009, there are still comments concerning the co-ordination. Nine comments concerning SWT SS2008 mentioned that the co-ordination between SWT and ES2 class needs to be improved and the work given was too complex to be solved within few classes. The co-ordination problem may due to the "public holidays" that interrupt the implementation of software tools in CC SWT SS2008 (Table 4.5). In order to reduce the complexity, more time is provided for the each subject. Coincidently Petrinet is not covered in ES2 SS2009; therefore, Petrinet is removed from SWT SS2009. This gives more time slots for other subjects. No negative comment is submitted during the evaluation for SWT SS2009. Instead there are two positive comments, one from a Computer Science student the other from a Mechatronics student. The Computer Science student mentioned that "software tool class is good" and the Mechatronics student mentioned that "software tool class enables the students to look deeper into the course content". Another possible reason for the positive feedback is the implementation of group work. All students who participated in SWT SS2009 also participated in GWB. With the implementation of GWB, the students need to spend more time on their own to get to know the topic before implementing it in the SWT class. Therefore, they do not need to battle so much with the theoretical concept, instead they can concentrate in implementing the tool.

The positive impact on Mechatronics students are more evident as compared the Computer Science students. This may due to the low participation by Computer Science students. The participation by Mechatronics students is higher as this is a compulsory course. SWT on the other hand, is only a selective course for Computer Science students [FB16+06]. The highest participant from Computer Science students was 20% during SS2007. There are at least 8 application courses available for the students to select. The participation for CC SWT improved from 4.3% in CC SWT SS2008 to 12.5% in CC SWT SS2009. The increase of participants for CC SWT SS2008 might due to the implementation for GWB. The tools proposed in GWB are also the tools that are taught in CC SWT SS2009. Therefore, it will be an advantage for the Computer Science students to participate in SWT as compared to other application courses.

6.6 Evaluation on Learning Preferences

6.6.1 Findings on System Development Preferences

Section 3.2.2 and 3.2.3 described the coming trends of using system modelling languages a communication between the different disciplines. The system modelling languages covered in ES2 are SA/RT and SysML. These methods are presented during the lecture, practised in the exercise and implemented using software in the SWT course. An opportunity to implement the modelling languages “independently” is provided in the group work. During GWB, the students are guided to describe the implement their system using SA/RT. The process and control diagrams describe the processes for the system. The diagrams are described to PSPEC and CPSEC levels. The students can transform the processes into a function, and PSPEC and CSPEC can describe the actions in this function. As it is possible to one to one map the modelling diagrams to the application program, it is expected that the students will use the SA/RT diagram or get some tips from the SysML diagrams when developing the system. Hypothesis 12 proposed that the modelling diagrams developed will be used more than the textual description.

Hypothesis 12 – As compared to textual description, students will find the modelling diagrams in SA/RT or SysML useful for system development

In GWB, the students are asked to rate which part of the information is most essential for them in developing the system. The students prepared three type of documentation for the project. The first is a text description of the system. The second is a system modelling of the control software using flow diagram and the third is a block diagrams that emphasizes on the reusability of different modules. The expectation is since the flow diagram in SA/RT are so carefully modelled and it represents the system requirements, the students will use the flow diagrams, its refinements of control and program specification, and the data dictionary for the system development. However, 4 out of 6 groups who responded to this question answered that the text description is the most useful document to develop the system. The other two groups missed the question and answered “documents from software tool” and “Google”. There is no mentioning of SA/RT or SysML as being helpful to the system development in any other comments field in the evaluation form.

Hypothesis 12 is not acceptable. The students did not find the modelling diagrams in SA/RT or SysML useful for system development.

6.6.2 Discussion on System Development Preferences

The findings from [FV07] mentioned that suitable modelling languages will assist the development of a system. The students in the experiment require lesser time to complete the mentioned that modelling languages (UML and ICL) are used for communication purposes, but there is no significance in terms of reducing errors as compared to textual description. This is influenced by the familiarity of the particular student with the modelling language. The experiments in [Pa08] show that “a graphical notation can be as precise as a textual one – and harder to understand”. This is reflected by the “unused” modelling diagrams in GWB. The findings from both research is reflected in ES2. Even though the students are taught and applied SA/RT in detailed in their project, the students developed their system mainly based on the text description. There are a few reasons for this development. Firstly, the students are able to express better in text and are more familiar with text. Taking into consideration that a student start to understand and work with text since kindergarten, and the modelling languages only in university or in higher secondary school, the experience the students have in understanding textual description should be longer Therefore, even though they may use modelling languages to describe the system, but when an option is given textual description (when suffice) will be selected. Secondly, the system developed by the student is not too complex and is still manageable to be described using text. The text descriptions provided by the groups in GWB are at most 2 pages A4 long. Therefore, this is still manageable. The development is also being conducted by 1 or 2 members of the team, and so the co-ordination to development the system separately or the management of interfaces between modules are manageable. Thirdly, the modelling methods are still new to the students. The students need more time to be familiar with its syntax and semantics before they can appreciate its usefulness. It is possible if the students are more familiar with the modelling notations, or had been implementing them in more system developments project, then they would appreciate its usefulness. However, more research is needed to confirm this.

6.6.3 Findings on Revision Effort

As the weekly lecture and exercise session for ES1 and ES2 is 1.5 hours, the students are encouraged to at least have 1.5 to 3 hours of revision per week. However, no specific record or the impact of specific methods in revision time is available. The WS0809 – SS2009 batch is suitable to evaluate the revision time with regards to the implemented methods. This is because methods implemented in both semesters are different (Pop Quizzes in ES1 WS0809 and GWB in ES2 SS2009), and the same group of students can answer to this question.

The work load for GWB is higher than pop quizzes. Students participating in GWB have fixed appointment where they need to present their work. As for pop quizzes, the students might learn for the pop quizzes but other than doing poorly, there are no extra consequences for not preparing. Hypothesis 13 proposed that the students will need to spend more time working on the course content on their own for participating in GWB as compared to pop quizzes. Table 6.19 presents the revision time for ES1 WS0809 and ES2 SS2009.

Hypothesis 13 – Group work encourages students to spend more time working on the course content on their own as compared to pop quizzes.

Table 6.19 - Comparison of revision time for ES1 WS0809 and ES2 SS2009

	PQ (WS0809)			GWB (SS2009)			Sig.
	N	Average	Std. Dev.	N	Average	Std. Dev.	
Computer Science	19	2.110	1.370	27	2.440	1.086	0.354
Mechatronics	11	1.180	0.405	11	2.090	0.944	0.008

During the GWB evaluation, the students are asked for the number of hours they spent revising the course content outside the lecture of the exercise session. 8 Computer Science students did not answer the number of hours they spent revising for pop quizzes in the previous semester. It is expected that with the implementation of group work, the students will be spending more time working on the course content on their own. This is true for the Mechatronics but not for the Computer Science students. There is a significant different for the Mechatronics students at 99% confidence level. However, there is no significant difference between the revision time in ES1 WS0809 and ES2 SS2009 for the Computer Science students. In ES1 WS0809 (with pop quizzes), the Computer Science students spent averagely 2.440 hours revising the course content, and 2.110 hours in ES2 SS 2009 (with group work).

Hypothesis 13 is not totally acceptable. Both spent more time for self revision when GWB is implemented as compared to Pop Quizzes. However, the difference is significant for Mechatronics students but not for Computer Science students

6.6.4 Discussion Revision Effort

One possible reason for these phenomena could be the inconsistency in attendance for the Computer Science. Their attendance is generally poorer than Mechatronics students. It is possible that they have to put in more effort to understand the course content. Therefore, the Computer Science students make up for the missed classes by spending more time revising the course content on their own. The Mechatronics students on the other hand, had been consistent with the attendance. By attending the class, the revision process is easier and therefore, they need to spend less time to revise the content in ES1 WS0809.

The implementation of group work should increase the time for revision. This is because the work load for GWB is higher. The Mechatronics students as expected spent more time for revision during the implementation of GWB. The increase of revision time is also significant. However, interestingly the Computer Science students did not increase their revision time significantly and the attendance is also poorer as compared to the previous semester as shown in Table 6.4. The number of hours spent on self revision is higher than the previous semester but the increase is not significant. One possibility might be 2 hours of revision is what the students can afford or willing to commit for a 3 credits subject. This point needs to be further affirmed in future research.

6.6.5 Findings on Practical Learning

The different learning behaviour for Computer Science and Mechatronics students are described in section 2.6.3. It is also mentioned that practical works are helpful in assisting students to learn and remember. This sub chapter will compare the preference of practical work between Computer Science and Mechatronics students. As exercise session requires more interactive learning than lecture classes, the preferred ratio between exercise and lecture will be investigated. Hypothesis 14 suggests that Mechatronics students will appreciate to have more exercises as compared to lecture, when compared to Computer Science students. Table 6.20 shows the evaluation of lecture's (L) and exercise's (E) structure and comprehensibility.

Hypothesis 14 – As Mechatronics students are more practical based, they would appreciate to have more exercise to lecture ratio as compared to Computer Science students

Table 6.20 - Comparison of department's evaluation result

Evaluation Criteria		L-Structure		L-Compre-hensibility		E-Structure		E-Compre-hensibility	
Discipline		CS	M	CS	M	CS	M	CS	M
SS2007	N	29	15	29	15	26	15	26	15
	Avg.	2.517	2.533	2.621	2.467	3.346	3.267	3.423	3.267
	Std. Dev.	1.090	0.743	1.083	0.990	1.018	0.799	0.902	0.799
	Sig.	0.959		0.648		0.797		0.581	
SS2008	N	16	19	16	19	16	17	16	17
	Avg.	1.938	2.632	2.125	2.737	2.688	2.471	3.375	3.294
	Std. Dev.	0.680	1.065	0.500	0.934	1.015	0.624	1.500	0.920
	Sig.	0.026		0.020		0.462		0.854	
WS0809	N	28	20	31	20	31	19	31	19
	Avg.	2.893	2.800	2.355	2.650	2.323	2.526	2.613	2.474
	Std. Dev.	0.994	0.952	0.877	1.040	0.791	0.964	0.803	1.172
	Sig.	0.747		0.281		0.420		0.620	
SS2009	N	11	9	11	9	11	9	11	9
	Avg.	2.545	3.000	2.455	3.667	2.091	2.556	2.000	2.556
	Std. Dev.	0.820	0.707	0.688	1.000	0.701	0.882	0.633	0.882
	Sig.	0.207		0.005		0.205		0.118	

For 3 out of the 4 semesters, the Computer Science students rated the lecture's structure and comprehensibility better than the Mechatronics students. The exception for lecture's structure happened in WS0809 and in SS2007 for comprehensibility of lecture. The rating between Computer Science and Mechatronics students for the category lecture's structure only significantly differs in SS2008, whereas the category lecture's comprehensibility is significantly different during SS2008 and SS2009. This shows that there is a tendency for Computer Science students to prefer lecture.

The rating for exercise's structure is inconclusive, with Computer Science rating it 2 semesters than Mechatronics, and vice versa. For the category exercise's comprehensibility, the Mechatronics students rated it better than the Computer Science students for 3 out of 4 semesters. However, the differences are not significant. When considering exercise's structure and comprehensibility on the whole, there is a tendency for the Mechatronics students to prefer exercise.

No department course evaluation is conducted in ES1 WS0807, and the university evaluation shows that the comprehensibility for both lecture and exercise are rated at 2.7. However, as the university evaluation does not distinguish the Computer Science students from the Mechatronics students, no comparison between Computer Science and Mechatronics students is possible for this semester.

With the focus on comprehensibility of lecture and exercise, the course evaluation shows that the Computer Science students are able to understand the lecture better than the Mechatronics students and the opposite is true for understanding the exercise content. Out of the 4 semesters, Computer Science students have better rating for lecture comprehensibility as compared to Mechatronics students. The only exceptional case happens in SS2007, where Mechatronics students graded lecture comprehensibility (at average rating 2.467) better than Computer Science students (at average rating 2.621). However, the difference is not significant. The opposite is true for exercise comprehensibility. Out of the 4 semesters, Mechatronics students rated 3 semesters better than Computer Science students. The only exceptional case happens in SS2009, with Mechatronics rating it at 2.556 and Computer Science at 2.000. The difference is also not significant for this case. Therefore, in term of understanding the course content, the delivery in lecture impacted the Computer Science students better and the exercise impacted the Mechatronics students better.

Based on the analysis from Table 6.20, the Computer Science students tend to prefer lecture better than the Mechatronics students. The Mechatronics students on the other hand understand the exercises better than the Computer Science students, but the result on exercise's structure is indecisive.

In order to have a clearer answer on the ratio between lectures and exercise the result from group work evaluation will be reviewed. During the group work evaluation for ES2 SS2009, the students were asked:

"In order to understand the lecture's content better, the ration between lecture and exercise should be _______", and the options provided are:

“4L: 2E”, “3L: 2E”, “2L: 2E”, “2L: 3E”, and “2L: 4E”, where L stands for lecture and E stands for exercise.

Table 6.21 presents the preferred ratio between lecture and exercise.

Table 6.21 - Students’ preferred ratio between lecture and exercise

Computer Science		N	Valid Percent	Cumulative Percent
1	2L:4E	4	15.40	15.40
2	2L:3E	4	15.40	30.80
3	2L:2E	13	50.00	80.80
4	3L:2E	4	15.40	96.20
5	4L:2E	1	3.80	100.00
	Total	26	100	

Mechatronics		N.	Valid Percent	Cumulative Percent
1	2L:4E	0	0.00	0.00
2	2L:3E	4	33.30	33.30
3	2L:2E	5	50.00	83.30
4	3L:2E	2	16.70	100.00
5	4L:2E	0	0.00	100.00
	Total	12	100	

As observed in Table 6.21 the 1 to 1 ratio between lecture and exercise has the highest percentage. 50% of the Computer Science students and 50% Mechatronics students selected 1 to 1 ratio. The cumulative percent shows that the 83.30% Mechatronics students prefer to have either the 1 to 1 ratio or more exercise, whereas the Computer Science students’ percentage stands at 80.80%. As compared to the Computer Science students, there is a tendency for Mechatronics to have more exercise than lecture.

Table 6.22 compare the preference on lecture and exercise ratio for Computer Science and Mechatronics students statistically.

Table 6.22 – Comparison between the preference on lecture and exercise ratio

	Computer Science			Mechatronics			Sig.
	N	Average	Std. Dev.	N	Average	Std. Dev.	
Lecture Exercise Ratio	25	3.240	1.052	11	3.180	0.751	0.870

Each ratio is given a value. “4L:2E” is assigned with 1, “3L:2E” is assigned with 2 and so on. This means the higher the average is than the higher ratio for exercise is preferred, and the opposite is true for lecture with the highest value at 1. The median is at 3 where the ratio between lecture and exercise is 1 to 1. The average ratios for both are slightly more than 3, meaning both have the tendency to prefer exercise more than lecture. Contradicting to the Hypothesis 14, the average ratio for Computer Science students is higher than Mechatronics students. This means that as compared to the Mechatronics students, the Computer Science students tend to prefer exercise more. However, the difference is not significant.

Hypothesis 14 is not acceptable. The accumulative percentage for Computer Science to have more exercise is higher but the difference is not significant.

As Mechatronics students have better understanding during the exercise session, and the nature of their study is more practical; the next hypothesis tests if the Mechatronics students are able to solve application questions better than Computer Science students. During the final exam for ES1 WS0809, there are two questions that are almost related. One is a theoretical question and the other an application questions. Table 6.23 shows the score distribution for both questions.

Hypothesis 15 – Mechatronics are able to solve application questions better than Computer Science students

Table 6.23 - Distribution of exam score for ES1 WS0809

Questions for ES1 WS0809	Discipline	N	Average Score	Std. Dev.	Sig.
3 (Theoretical)	Computer Science	47	4.790	1.885	0.280
	Mechatronics	29	4.250	2.372	
4 (Practical)	Computer Science	47	1.480	1.877	0.186
	Mechatronics	29	2.070	1.865	

Mechatronics students fare better than Computer Science students in terms of application questions. Question 3 and 4 are both on Petrinet. Question 3 is more theoretical; the students are given a Petrinet and should draw the “starting set”, “index matrix”, and “liveliness graph”. Question 4 on the other hand is more on application; it requests the students to draw and describe two Petrinet graphs, each representing the “polling procedure” and the “interrupt procedure”. The students were taught Petrinet in “Chapter 3 – Typical Architecture”, and polling and interrupt in “Chapter 4 – Scheduling”. Question 4 requires the students to combine both these knowledge, to solve the question. Table 6.23 shows that the Computer Science students did better in question 3. As for question 4, only 6 out of 99 students scored 5 points or above; 3 are from Computer Science and 3 from Mechatronics. The difference of score is more evident for the practical question as compared to the theoretical question.

The next comparison involves the same batch of students. During the ES2 SS2009 final exam, Question 4,5, and 6 are based on an elevator system The students are given the system description, a list of sensors and actuators and they are suppose to “design” and “implement” the system. Question 4 requests the students to design the system in SA/RT. In Question 5, the students should program three modules for the elevator system in structured text, function block diagram and ladder diagram. Question 6 requests the students to draw the block definition diagram in SysML, and architecture diagram in SA/RT. Table 6.24 shows the score distribution for these three questions.

Table 6.24 - Final exam question result for ES2 SS2009

Questions for ES2 SS2009	Discipline	N	Average Score	Std. Dev.	Sig.
4	Computer Science	56	7.087	2.523	0.492
	Mechatronics	24	7.533	2.933	
5	Computer Science	56	8.319	4.077	0.116
	Mechatronics	24	9.967	4.635	
6	Computer Science	56	5.844	2.892	0.850
	Mechatronics	24	5.704	3.273	

All the 3 questions described above are considered as application question according to Anderson and Krathwohl's taxonomy. The students are given a new scenario and they are required to apply the theory they learnt in this new situation. The Mechatronics scored higher than the Computer Science students for question 4 and 5. The difference of average grade question 4 is 0.446, and for question 5 is 1.648. The Computer Science students did better in question 6 but the average score difference is small, namely 0.14.

Hypothesis 15 is not totally acceptable. Mechatronics students do better in application questions but the difference is not significant.

6.6.6 Discussion on Practical Learning

Different research proposed that Mechatronics students prefer practical work. The purpose of section 6.6.6, is to find the impact of practical learning in relation to Computer Science and Mechatronics students. The analysis will be divided to impact and preference of exercises, and answering practical questions.

From the analysis above it is clear that as compared to the Mechatronics students, the Computer Science students gave better rating for the lecture's structure and understanding. The Mechatronics students on the other hand can understand the exercise better than the Computer Science students, but the preference ratio for exercise is lower than the Computer Science. One possible explanation for this trend is as the Mechatronics students are able to understand the exercise content; therefore, they find that the current ratio suffices. The Computer Science students on the other hand, gave poorer rating for exercise's comprehensibility is lower, and therefore, they felt more exercise is needed. From these observations, it can be concluded that exercise is important for both Computer Science and Mechatronics students.

On the aspects if the Mechatronics students can answer practical questions better, the findings show that the Mechatronics fared better than the Computer Science. However, this aspect is not so "cleanly" evaluated as Mechatronics students also have better average grade as compared to the Computer Science students for both WS0809 and SS2009. Therefore, there might be a tendency for Mechatronics students to do better at practical questions.

Generally, the practical aspects in learning are important for both Computer Science and Mechatronics students. There is no significant difference between both disciplines for practical learning.

6.6.7 Findings on the Importance of Exam and Attendance

Different methods have been introduced to attract students' interest towards the course ES1 and ES2. However, it is believed that the main motivation for the students to work on the course content on their own (personal revision) is not mainly due to personal interest but to do well for the final exam. Table 6.25 presents the students' respond concerning the motivation to do revision. The data is from course evaluation in ES2 SS2009.

Hypothesis 16 – The main motivation for students to revise the course content is to do well for the exam

Table 6.25 - Motivation for students to do revision

Motivation	Exam	Exercise	Group Work	Interest	Study Group
CS (11)	8	5	5	3	2
M (9)	9	5	2	3	2
Number of Students	17	10	7	6	4

Table 6.25 shows that the students selected “Preparing for exam” as their main motivation to revise the course content. All together 20 students participated in this evaluation and 17 selected “preparing for exam” as their motivation. This is followed by preparing for the exercise, working on the group work, self – interest and study group. Only 2 out of 9 Mechatronics students selected group work as the motivation to revise course content. There are also positive comments concerning the opportunity to accumulate extra points for the final exam through pop quizzes and group work.

Students will make decisions based on what will help them to score better in the final exam. The students who did not participate in GWB also deemed that they can do better when using the time for group work to revise for the exam. 2 students, both Computer Science students, who participated in GWB, mentioned that they would prefer to have pop quizzes instead of group work. This is because more effort is needed for group work but the extra bonus point for the exam is still the same. During the beginning of ES1 WS0809, the students are requested to write down their expectations on the course. 4 out of 6 students who wrote about the expectations for ES1 WS0708 mentioned that they would like to learn what is relevant to the exam; and the lecture and exercise should prepare the students towards doing well in the exam.

From the statistical data and the comments from the students, it can be concluded that the focus for the students is to do well in the final exam. Learning for the sake of interest is low. Therefore, the main motivation on revising the course content is doing well in the exam.

Hypothesis 16 is acceptable. The main motivation for students to do revision is to do well in the final exam.

According to [Ro93] and [AD+08], students' attendance correlates with the exam's grades positively. The students, who have perfect attendance, scored averagely a full grade better than the students who have half the attendance [Ro93]. Hypothesis 17 will test if this is also true for the students in ES1 and ES2. Table 6.26 compares the attendance with the student's grade.

Hypothesis 17 – Students who attend the class more often will do better in the exam as compared to students who did not

Table 6.26 - Attendance vs. ES1 WS0809 grade

Final Exam Grades		Attendance (Full attendance 15 classes)				Total
		0 to 6	7 to 9	10 to 12	13 to 15	
Better Grade | ES1 Grade | Poorer Grade	1.0	0	0	4	4	8
	1.3	0	0	0	1	1
	2.3	0	0	1	0	1
	2.7	0	0	1	0	1
	3.3	0	0	1	1	2
	3.7	1	0	1	0	2
	4.0	0	1	0	1	2
Total		1	1	8	7	17

As the evaluation data is collected anonymously, no direct comparison can be made for the whole class or whole evaluation. However, in ES1 WS0809, a group of 17 students volunteered and relate their grade to the evaluation form number after the final exam was made known. As shown in Table 6.27 from the 7 students who have almost perfect attendance ("13-15" times in a semester), 5 scored very good grades (1.0 and 1.3), 1 scored satisfactory grade (3.3), and 1 scored sufficient grade (4.0). Two students who attended less than 9 times scored only sufficient grade (3.7 and 4.0). Another observation is out of the 8 students who scored 1.0, 4 of them have almost perfect attendance and the other 4 have the second highest attendance category.

Table 6.27 – Correlations between ES1 grade and students' attendance

	Attendance		
	N	Correlations	Sig.
ES1 Grade	17	-0.434	0.082

As shown in Table 6.27, using Pearson correlation, there is a negative correlation between the grades and the attendance. The higher the attendance, the lower the grades will be. This means the higher the attendance, the better the grade will be. The correlation is significant at $p = 0.1$.

Hypothesis 17 is acceptable. There is a significant correlation between the student's attendance and the grade for final exam.

6.6.8 Discussion on the Importance of Exam and Attendance

Even though, the intake requirements for both Computer Science students and Mechatronics students are the same in University Kassel [SK07], [Le04], the learning preferences from both groups of students are different. ES1 and ES2 are for students in their third and fourth semesters. The characteristics that determine the students to choose the particular discipline and/or the foundation courses in the first four semesters plays a role in forming the different cognitive mind set. The Computer Science students are more receptive towards class lectures as compared to the Mechatronics students. However, when it comes to practical learning, both disciplines seem to place almost the same importance.

As described in section 2.3.2, the attendance impacts the final exam grade positively. This is also reflected in this research. Therefore, it is important to motivate students to attend classes. One of the methods is through the implementation of active learning methods. The most effective method to have motivating class is to engage students' involvement during lecture. In class demonstration, round-robin questioning, and cold-calling are the few methods that can be implemented. The effectiveness of constant feedback using 5IMPLe dwindles after being implemented a few times.

6.7 Evaluations on the Efforts for Implemented Methods

6.7.1 Findings on Efforts for Implemented Methods

Having compared the effectiveness of the methods implemented to the different disciplines. This sub-chapter will discuss on the efforts needed for each method. The efforts recorded for each method are estimated hours spent by the researcher. First time effort indicates the effort required to prepare the material if no prior knowledge on the subject is available. First time efforts include getting to know the subject, sourcing for the information required, and preparing the materials for the class. One time effort is the effort required to prepare the material for the class when the subject is already known, materials are already available. Modifications may be necessary or new information can be added but the basic information is readily available. The hours of effort indicated in the table are the preparation time needed for 90 minutes of class. Repetitive effort is effort based on the class size. The value "n" indicates the group of 10 students. If there are 80 students, then "n" will be 8. Occurrences in a semester are the number of times it occurs in the semester.

Table 6.28 shows the efforts needed for the different methods.

Table 6.28 – Comparison of efforts required for the different methods

Method	First Time Effort	One Time Effort	Repetitive Effort	Occurrences in a Sem.
Group Work				
a) GWA - Partial life cycle	not recorded	2.0 hrs	n(2.0 hrs)	1
b) GWB - Whole life cycle	not recorded	2.0 hrs	n(6.0 hrs)	1
Motivating Lecture & Exercise	Low	Low	Low	Low
Exercise	9.9 hrs	6.0 hrs		(a,b)
Pop Quizzes	3.5 hrs	3.5 hrs	n(0.5 hrs)	(a,b)
5IMPLe			n(0.2 hrs)	2
Bench Scale Equipment	12.0 hrs	4.5 hrs		2
Course Coupling				
a) Loosely coupled software tools		2.0 hrs		
b) Closely coupled software tools		3.0 hrs	c(0.2 hrs)	(b,c)

Legends: hrs – hour, n – groups of students in 10, a – occurrences of first time effort, b – occurrences of one time effort, c – occurrences of repetitive effort

The effort needed for group work equals to the summation of first time effort or one time effort added with repetitive effort, the summation will be multiplied with the number of occurrences. First time effort includes time to ponder and come up with the idea of the group work. The exact time needed to come up with the ideas for both group works are not being recorded. The one time effort includes the time for the group work guidelines. There is only one occurrence of group work per semester.

effortGroupWork = a(firstTimeEffort)+b(oneTimeEffort+repetitiveEffort), where a,b >0

There are usually 14 lectures and exercise weeks in a semester. Normally, out of the 14 weeks, 7 weeks are for exercises. If the exercise is a new subject then the hours needed is the “first time effort” and if the exercise is already available and only modifications are needed then the “one time effort” will be used. No repetitive effort is required as the exercise questions are not marked. The first time effort and one time effort also includes the time needed to prepare the questions and answer sheets and upload them to the internet.

effortExercise = a(firstTimeEffort)+b(oneTimeEffort), where a,b >0

The first time effort and one time effort for pop quizzes are the same as the pop quizzes implemented so far are not reused. The first time effort and one time effort are the hours required to prepare a 15 to 20 minutes quiz. Printing and stapling of the pop quizzes are not included as this is done automatically by the photocopy machine. The repetitive effort is the hours needed to grade the pop quizzes. Pop quizzes are conducted between 2 to 3 times in a semester.

effortPopQuiz = a(firstTimeEffort+n(repetitiveEffort))+b(oneTimeEffort+n(repetitiveEffort)) , where a,b >0

The effort needed to implement 5IMPLe depends on the feedback received. If the problems voiced by the students are almost of the same category, then the time needed to prepare for the feedback might be shorter. The time required to prepare for the answers will vary depending on the way it is done, preparing slides or full answer sheets for upload might take longer than just taking note of the categories and spontaneously answering it in the class. The effort presented here are the hours needed to go through the feedback and to categorise them to its categories. The feedback for summer semester 2009 is integrated in the exercise; therefore, the hours to prepare for the feedback are already included in the exercise.

effort5IMPLe = b(n(repetitiveEffort)), where b >0

Bench scale equipment had been used in the exercise session. In the coupled course SWT, the same hardware description is used, whereas the software functions are different. The same hardware descriptions are also given to the students who participated in GWB. Here the first time effort to prepare the diagrams, hardware lists, text descriptions, slides etc requires 12 hours. Modifications on current information to add or change a new software function require 4.5 hours. The hours in effort for bench scale equipment are hours saved. For example, as bench scale equipment was implemented in 3 different contexts, the time saved is between 24 hours ((3-1)*(12) =24) and 9 hours ((3-1)*(4.5) =9).

effortSaved = (a-1)(firstTimeEffort)+(b-1)(oneTimeEffort), where a,b >0

The effort required for the initial discussion concerning the schedule to couple two courses takes about 2 hours. In closely coupled SWT, the additional hour is for the feedback between the teaching assistants for both the courses after each exercise session. Only 5 to 10 minutes of discussion is required. Mainly, the concepts that are difficult to the students will be discussed. This discussion is not dependent on the class size.

effortCCSWT = b(oneTimeEffort)+c(repetitiveEffort), where b, c >0

The efforts required for each semester differs. For example pop quizzes are at times implemented 2 or 3 times in a semester. When a new subject is included in the exercise, then “first time effort” will be used for calculation instead of “one time effort”. As there are usually 7 sessions of exercises in the semester, an average value will be taken. 3 exercises will be calculated as prepared from scratch. Therefore, “first time effort” will be used here. The other 4 exercises will be considered as modifications from existing materials; therefore, “one time effort” will be used here. The calculation of efforts required are shown in Table 6.29

Table 6.29 – Estimation of efforts required for implemented methods

Method	Efforts Required
Group Work	
a) GWA - Partial life cycle	*effortGroupWork = 0(firstTimeEffort)+1(2+7(2.0))=16*
b) GWB - Whole life cycle	*effortGroupWork = 0(firstTimeEffort)+1(2+7(6))=44*
Motivating Lecture & Exercise	*Low*
Exercise	*effortExercise = 3(9.9)+4(6)=53.7*
Pop Quizzes	*effortPopQuiz = 2(3.5+5(0.5))=12.0*
5IMPLe	*effort5IMPLe = 2(2(0.2))=0.8*
Bench Scale Equipment	*effortSaved = (2-1)(12)=12.0*
Course Coupling	
a) Loosely coupled software tools	*effortLCSWT = 2*
b) Closely coupled software tools	*effortCCSWT = 1(3)+12(0.2)=5.4*

6.7.2 Discussion on Efforts for Implemented Methods

From Table 6.28, it is observed that motivating lecture requires the least effort to implement. There might be some preparation time. However, this depends on the teaching instructor or teaching assistant. Some people are “gifted” in having interaction with students.

The next effort that requires least effort is the coupling with software tools. The only time required is for discussion on the course timeline before semester starts and short updates after each session of exercise.

The third placing is pop quizzes. However, this depends on the number of students in the class. From the context of ES1 and ES2 this is doable. However, classes with more than 150 students would not be practical especially when there is only one teaching instructor and teaching assistant handling the class. Another possibility is the installation or personal response system for pop quizzes.

The next in row is group work. The effort to implement GWA is lesser than GWB. From the analysis conducted for group work in sub-chapter 6.4, the implementation of GWB positively influence both Computer Science and Mechatronics students as compared to GWA. However, the effort required is also higher. The implementation of exercise requires the highest effort. From the feedback in section 6.6.5, this is part of the core in ES1 and ES2. Both Computer Science and Mechatronics students give the same importance to exercise as to lecture.

The implementation of 5IMPLe does not require much effort. However, it is only implemented twice in this research. As discussed in Hypothesis 7, the impact of 5IMPLe is not visible. If this is conducted after every chapter then about 5 to 10 hours are required to implement 5IMPLe at higher frequency.

7 Conclusion on Methods Implemented

Chapter 6 evaluated the different implementation methods according to bench scale equipment, lecture and exercise, 5IMPLe, pop quizzes, group work and course coupling. Different research found on the course of this research focuses only on a certain group of students. This research has the opportunity of observing two different groups of students, working with the same set of course content, materials and methods. This is a good opportunity to compare one group of students from different disciplines. [HW+02] suggest that the implementation of an active learning method, for example muddy card, may be influence by another method. Therefore, an aspect that will be investigated in this chapter is the dynamics between the implemented methods.

In order to have a clearer view between the dynamics of methods implemented, the first sub-chapter will discuss the analysis based on different combinations of methods. This will then be followed by a general conclusion on the hypothesis, impact of methods to the disciplines and the effort required to implement these methods.

7.1 Dynamics between Methods Implemented

From the analysis in chapter 6, dynamics of the methods and the impact on each discipline will be discussed. The effectiveness of different methods to both Computer Science and Mechatronics students will be generalised. Table 4.6 presented the implementation of methods across the semesters together with its objectives. For ease of references, the table is once again presented here, but without the objectives of each method.

Table 7.1 – Implementation of methods across the semesters

Methods	Bench Scale Equip-ment	Moti-vating Lecture	Exercise	Pop Quizzes	5IMPLe	Group Work		Course Coupling	
						GWA	GWB	LC SWT	CC SWT
WS0607	--	X	X	--	--	--	--	--	--
SS2007	X	X	X	--	--	X	--	X	--
WS0708	X	X	X	X	--	--	--	--	--
SS2008	X	X	X	X	--	--	--	--	X
WS0809	X	X	X	X	X	--	--	--	--
SS2009	X	--	X	--	X	--	X	--	X

The different implementations can be divided into 2 groups; one is “general participation” and the other with “optional participation”. The general participation group means any students who participate in ES1 or ES2 will be part of the implementation, whether they want it or not. Methods in this category are the implementation of bench scale equipment, motivating lecture, and exercise. The other group, namely optional participation means the students can choose to participate or not to participate in this method. Three methods belong to this category, namely pop quizzes, 5IMPLe, group work and course coupling with software tools course. As described in section sub-chapter 4.4, software tools course is an optional application course for the Computer Science students but it is compulsory for Mechatronics students. Even though software tools course is compulsory for the Mechatronics students, they can still choose to take this course in other semesters. It is not interdependent with ES2, therefore, software tools course is also considered as “optional participation” for Mechatronics students.

Generally, it is not possible to relate the ratings for “general participation” methods to the final exam result, as the evaluations are conducted anonymously. Secondly, all students who participate in ES1 and ES2 are part of this implementation, and so the option to compare, for example “students who take part in motivating lecture but not exercise”, does not exist. Therefore, other than the discussion presented in sub-chapter 6.2 and 6.3, no extra relationship or dynamics between these methods can be drawn.

The following sections will discuss on the dynamics between methods for “optional participation”. The dynamics between group work, pop quizzes and software tools will be based on the average grade, as comparison between participants and non-participants for each method.

The combination for students participations are grouped to 6 different categories for analysis (Table 7.2). The differences between 2 values that are significant at ($p = 0.05$ or $p = 0.1$) will be connected using a line and a dotted line respectively. Differences that are not significant, but has close tendency to significance will be connected with line-with-dot.

Table 7.2 – Categories of participations for different methods

Category of Participation	Extra Points Method	Course Coupling
	Group Work (GWA or GWB) or Pop Quizzes	SWT (loosely coupled or closely coupled)
1	0 - No Participation	0 - No Participation
2	1 - Got Participation	0 - No Participation
3	0 - No Participation	1 - Got Participation
4	1 - Got Participation	1 - Got Participation
5	X - Not Considered	1 - Got Participation
6	1 - Got Participation	X - Not Considered

7.1.1 Comparison for Summer Semester 2007

Computer Science Students:

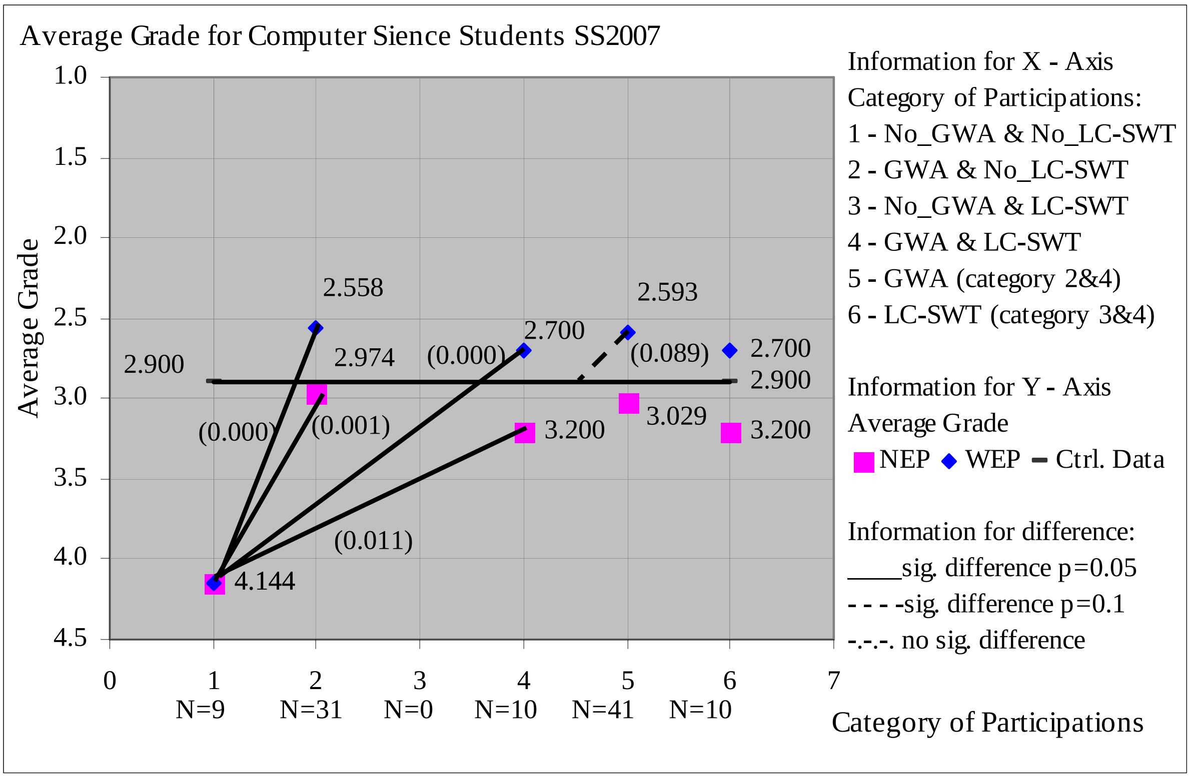


Figure 7.1 – Overview of Computer Science students result for summer semester 2007

During summer semester 2007, partial life cycle group work (GWA) is implemented. Students who participated in "neither GWA nor LC SWT" (category 1) have the poorest result. The difference is significant to those who took part in "GWA but not LC SWT" (category 2), as well as those who took part in "GWA and LC SWT" (category 4). Therefore the participation for students in GWA is beneficial for Computer Science students.

There are no students who "did not participate in GWA but in LC SWT" (category 3). Therefore, the sole impact of LC SWT on Computer Science students cannot be analysed. The closest NEP average grade to the control data is for students who participated in GWA. When not taking into consideration LC SWT, the difference between students who participated in GWA (category 6) is significantly better than control data at $p = 0.1$. This again affirms the discussion in section 6.4.1 that the impact of GWA is strongly positive for Computer Science students.

Mechatronics Students:

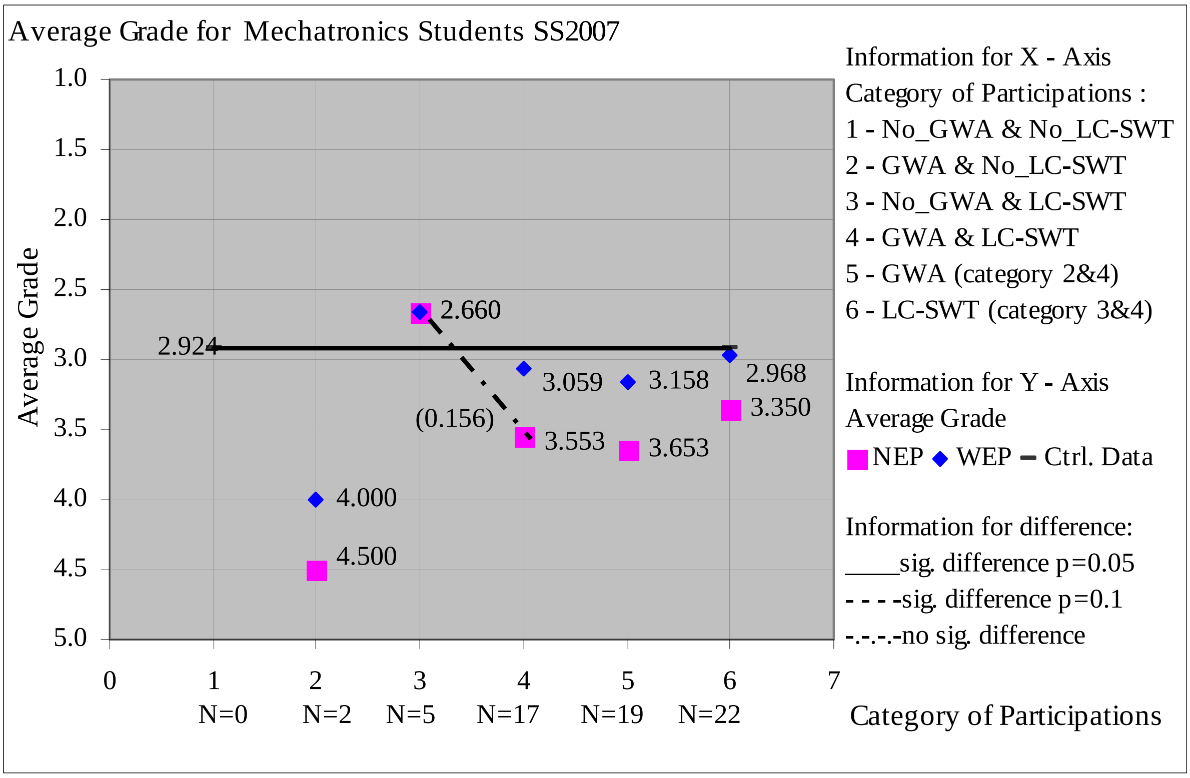


Figure 7.2 – Overview of Mechatronics students result for summer semester 2007

The implementation of GWA does not significantly impact the Mechatronics students. The group that did the best is the group who “did not participate GWA but in LC SWT” (category 3). The NEP average grade for this category is significantly different when compared to students who participated in both “GWA and LC SWT” (category 4). This shows that the impact of GWA to the students is not as positive as LC SWT.

The WEP average grade is generally lower than control data, but the difference is not significant. This shows that the Mechatronics students did poorer in ES2 as compared to other subject they took in summer semester 2007.

Similarly to the Computer Science students, the NEP average grade for Mechatronics students is similar are poorer than control data. The difference with control data for Mechatronics students is however bigger as compared to that of Computer Science. This confirms the discussion in section 6.4.1 that GWA has no significant impact on Mechatronics students. LC SWT seems to have more positive impact than GWA for the Mechatronics students.

7.1.2 Comparison for Summer Semester 2008

Computer Science Students:

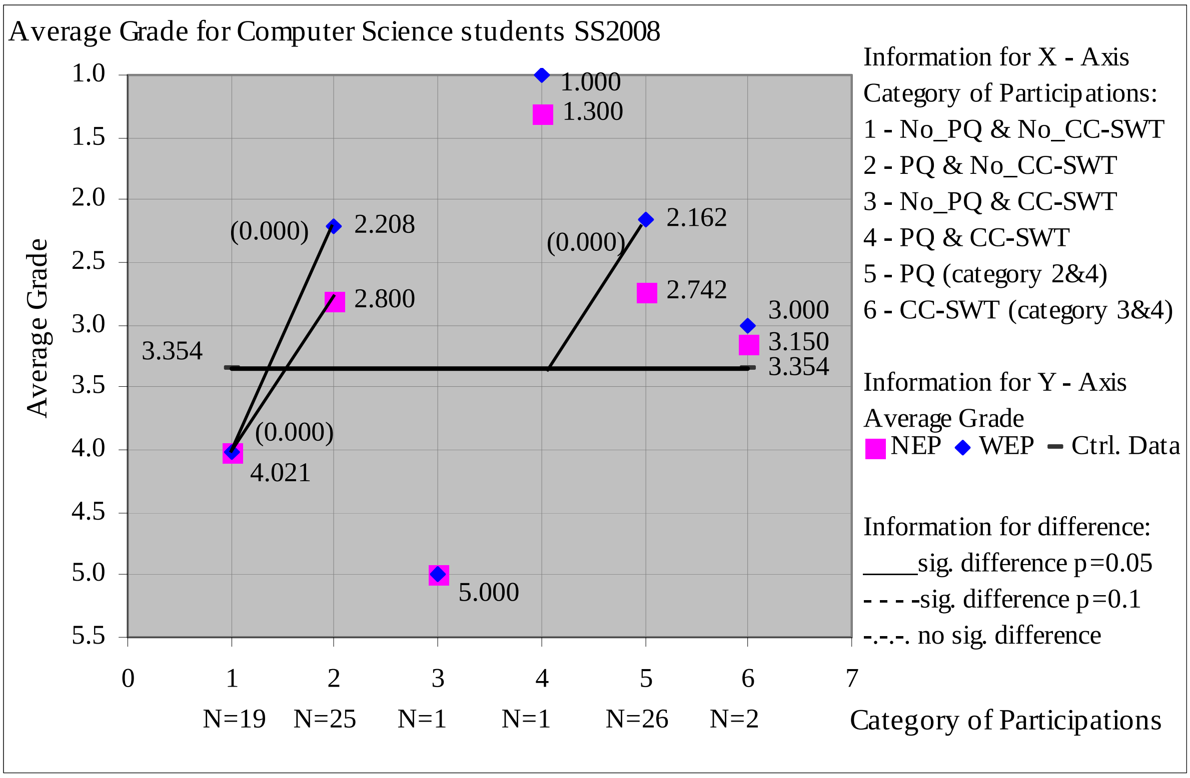


Figure 7.3 - Overview of Computer Science students result for summer semester 2008

During summer semester 2008, pop quizzes are implemented. Similar to the Computer Science students in summer semester 2007, students who participated in „neither pop quizzes nor CC SWT SS2008" (category 1) scored the poorest average grade. Students who participated in "pop quizzes but not in CC SWT SS2008" (category 2) have significantly better NEP and WEP average grade as compared to the students who participated in "neither pop quizzes nor CC SWT SS2008" (category 1). The only 2 students who participated in CC SWT SS2008 have two extreme results, one very good and the other very poor. Therefore, no definite conclusion on the impact of CC SWT SS2008 to Computer Science students can be concluded. The WEP average grade of pop quiz participants (category 5) is significantly better than control data.

In summer semester 2007, the NEP average grade for category 2, 4, 5, and 6 are poorer than control data. The NEP average grades for these categories in summer semester 2008 are the opposite. They are all better as compared to control data. This shows that the impact of pop quizzes is better than GWA for Computer Science students.

Mechatronics Students:

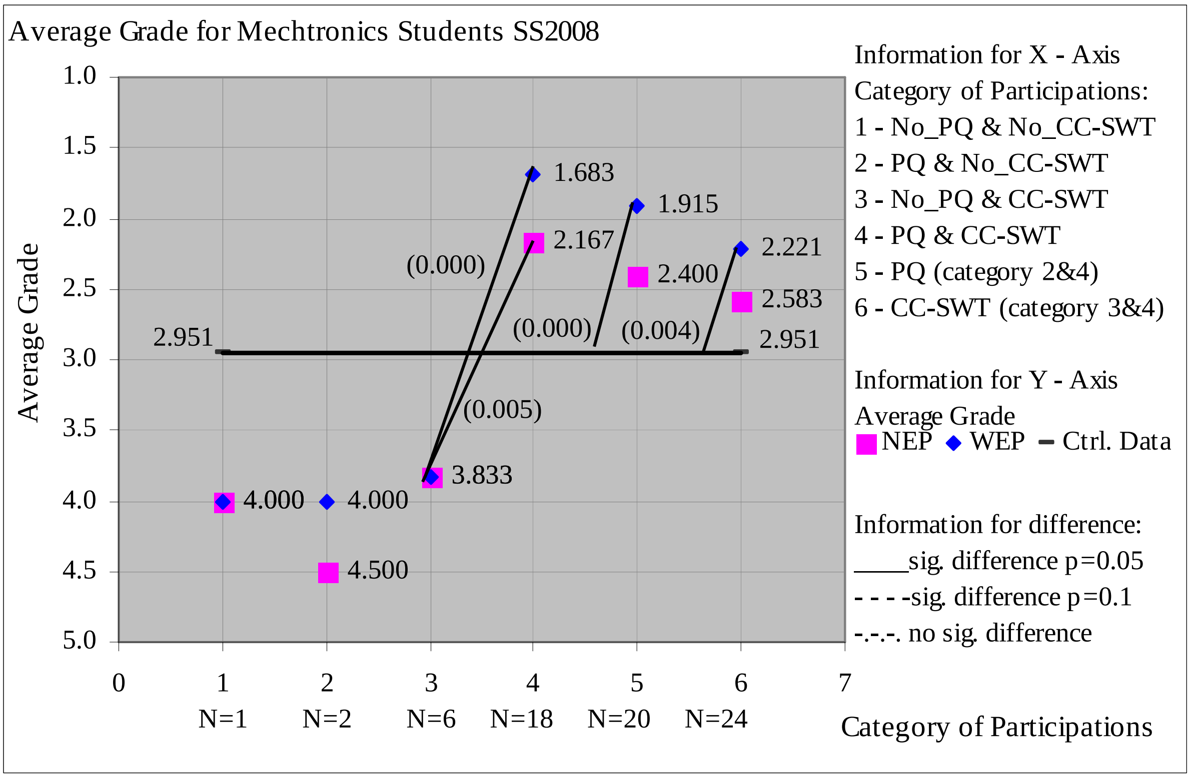


Figure 7.4 - Overview of Mechatronics students result for summer semester 2008

The impact of participating in both CC SWT SS2009 and pop quiz (category 4) is the most positive. The WEP and NEP average grade is significantly better than those who only participated in CC SWT SS2008 and not in pop quiz (category 3). This shows that the impact of pop quizzes alone without CC SWT SS2009 does not positively impact the students. However, when observing the data on the whole, the impact of students who participated in pop quizzes (category 5) and CC SWT SS2008 (category 6) are both significantly better than control data. This is due to the influence from the average grade in category 4.

As compared to the semester before in summer semester 2007, the Mechatronics students do better than control data. Both NEP and WEP average grade are better than the average control grade. During the summer semester 2007, even though the difference is not significant, students in category 5 and 6 have poorer NEP and WEP average grade as compared to control data. This shows that the implementation of pop quiz and CC SWT SS2008 is a good combination.

7.1.3 Comparison for Summer Semester 2009

Computer Science Students:

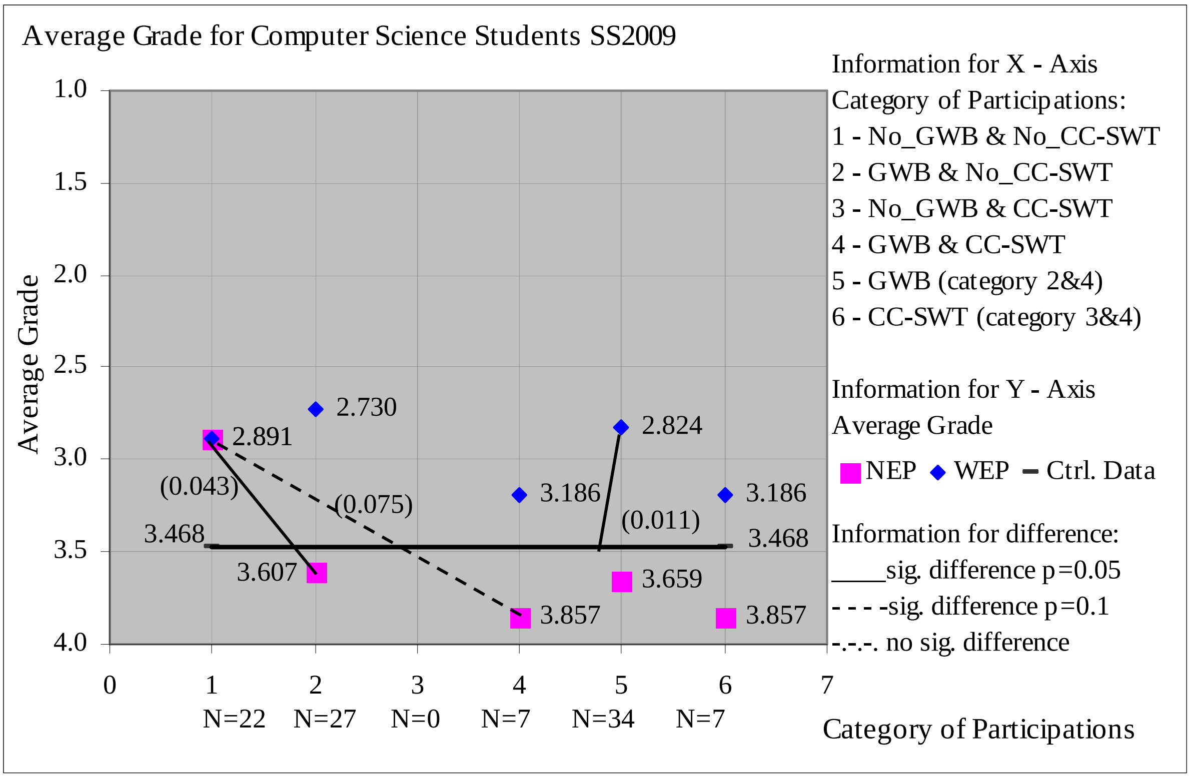


Figure 7.5 - Overview of Computer Science students result for summer semester 2009

During summer semester 2009, the same course coupling with SWT is conducted (CC SWT SS2009). In this semester group work with implementation of whole life cycle process (GWB) is conducted. Unlike summer semester 2007, where only 9 students did not take part in GWA and LC SWT; during summer semester 2009, 22 students did not take part in GWB and CC SS2009 (category 1). This is the only exceptional case where students in category 1 have better average grade than control data. The NEP average grade is significantly better than those in category 2 and 4. This shows that Computer Science students are able to do well by “just studying”.

GWB have more impact than CC SWT SS2009, the students who took part in GWB but not CC SWT SS2009 (category 2) have better average grade than those who took part in CC SWT SS2009 but not in GWB (category 4). However, when considering the social and practical impact of GWA that is reflected in the extra points, then the average grade for GWB is at par with category 1.

By comparing the improvement of average grade due to extra points in category 2 and 4, it can be deduced that participation in CC SWT do not equate to better deliverance of GWB documentations. The NEP average grades for categories 2, 4, 5, and 6 are similar to that in SS2007. They are all poorer than the average control grade. In fact, the only average grade that is better than the average control grade are those in category 1, those who did participate in neither GWA nor CC SWT SS2009.

Mechatronics Students:

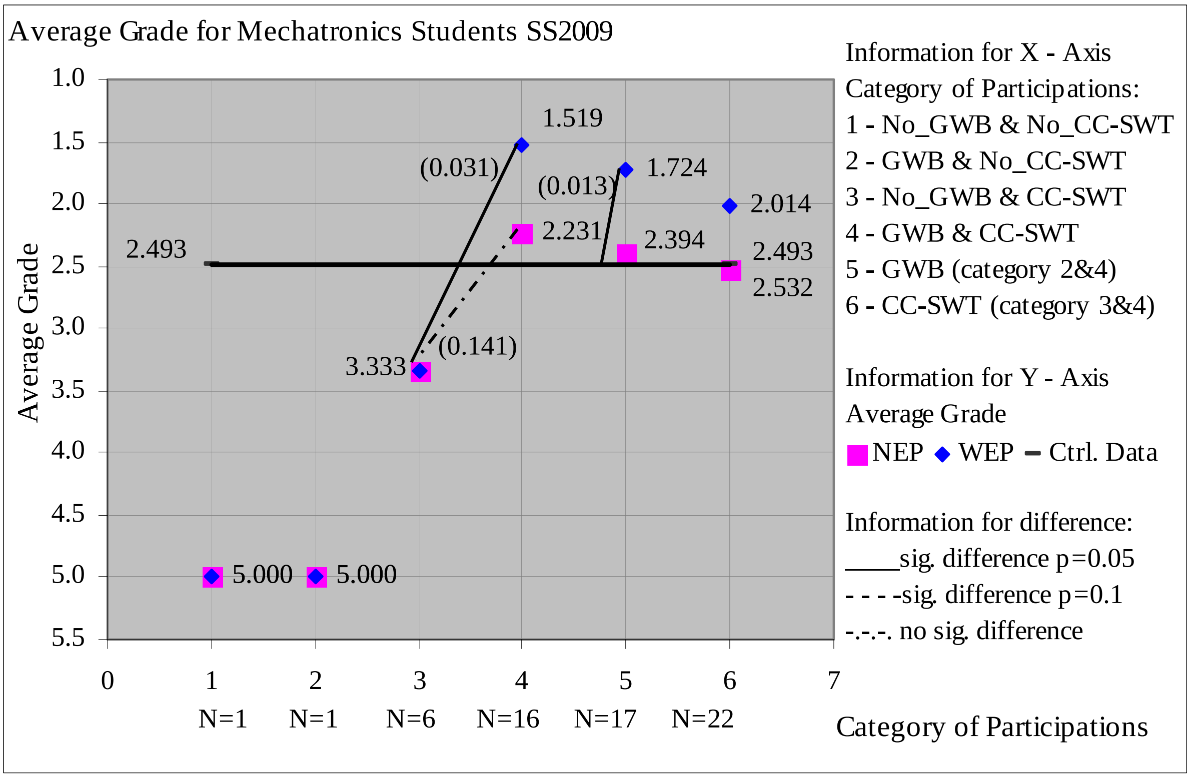


Figure 7.6 - Overview of Mechatronics students result for summer semester 2009

Students who participated in both GWB and CC SWT SS2009 (category 4) have the highest NEP and WEP average grade. Students who did not participate in GWB but in CC SWT SS2009 (category 3) tend to have poorer average grade but it is not significant. This shows that the impact of GWB may be stronger than CC SWT SS2009.

Mechatronics students who participated in GWB (category 5) have better NEP average grade than control data, but the opposite is true for Computer Science students. This leads to the conclusion that GWB is more beneficial for the Mechatronics students as compared to the Computer Science students.

7.2 General Conclusion

This section will relate the findings on the methods implemented to the disciplines and the efforts required. Firstly, Table 7.3 presents the overview of the hypothesis in chapter 6. The hypotheses are grouped to 3 categories, namely acceptable, not totally acceptable and not acceptable. The “+” sign means the analysis is significantly true and the “—” sign means the opposite findings to hypothesis is significant, and “O” means the findings are not significant.

Table 7.3 - Summary of Hypotheses

Hypotheses	Accept-able	not totally accept-able		not accept-able
	for both	CS	M	for both
Hypothesis 1 – Implementation of consistent bench scale equipment will help student to understand the course content	++			
Hypothesis 2 – Motivating lectures encourage students' attendance	++			
Hypothesis 3 – Motivating lectures will influence students' comprehensibility of lecture	++			
Hypothesis 4 – Motivating lectures will influence the usefulness of demonstration	++			
Hypothesis 5 – Pop quizzes will motivate students' attendance as compared to semesters without pop quizzes				--
Hypothesis 6 – Pop quizzes will prepare the students for the final exam	++			
Hypothesis 7 – Course feedback using 5IMPLe will motivate the students to provide feedback concerning the course content				--
Hypothesis 8 – GWA participants will have better average grade than non-participants		++	-	
Hypothesis 9 – GWB participants will have better average grade than non-participants		--	++	
Hypothesis 10 – Students who participated in more than 1 lifecycle phase will have better grade than those who only participated in 1 lifecycle phase.		+	+	
Hypothesis 11 – Closely coupled software tools course (CC SWT) has more positive impact as compared to loosely coupled software tools (LC SWT)		+	+	
Hypothesis 12 – As compared to textual description, students will find the modelling diagrams in SA/RT or SysML useful for system development				--
Hypothesis 13 – Group work encourages students to spend more time working on the course content on their own as compared to pop quizzes.		+	++	
Hypothesis 14 – As Mechatronics students are more practical based, they would appreciate to have more exercise to lecture ratio as compared to Computer Science students				--
Hypothesis 15 – Mechatronics are able to solve application questions better than Computer Science students		+	+	
Hypothesis 16 – The main motivation for students to revise the course content is to do well for the exam	++			
Hypothesis 17 – Students who attend the class more often will do better in the exam as compared to students who did not	++			

Legends: ++ significantly better, + better, -- significantly poorer, - poorer

Generally, during ES2 SS2007 the NEP average grade for both Computer Science and Mechatronics students are both poorer than the control data. This may due to the fact that the course contents are just modified from ES2 WS0607.

As described by Hypothesis 3 motivating lecture plays a role in helping the students to understand the course content better. As shown in Table 6.28, this requires the lowest effort. Whenever possible this method should be implemented. However from the available data, motivating lecture does not seem to be a factor when it comes to doing well for the final exam. The Computer Science students who did not participate in GWB or CC SWT SS2009 have the best NEP average grade in the semester. It is not possible to determine if the students who attended classes in motivating lecture have better grade than those attending lectures that are not motivating.

When comparing the impact of pop quizzes, group work and course coupling with software tools, the NEP average grade where pop quizzes are implemented in two consecutive semesters are better than the average control grade. This is elaborated in the discussion for Hypothesis 6. Another semester where the NEP average grade is better than the average control grade happens in summer semester 2009. The students who participated in both GWB and CC SWT SS2009 have slightly better NEP average grade as compared to the average control grade. As observed in Table 6.29, pop quizzes required 12 hours effort per semester in this research. This is between the effort needed for course coupling with software tools course and group work.

The effort for course coupling with software tools course is not as high as pop quizzes or group work, the impact for course coupling with software tools course on final exam result for ES2 is also not that evident. Computer Science students who participated in LC SWT SS2007, and CC SWT SS2009 have poorer NEP average grade than the average control grade. Mechatronics students who only participated in CC SWT SS2008 or CC SWT SS2009 have poorer NEP average grade as compared to the average control grade. Interestingly, this is opposite for the implementation of LC SWT SS2007. This shows that by only participating in a coupled course, it might not necessarily improve the grades in ES2. However, when the participation in a coupled course is also combined with either pop quizzes or group work, then the results improved to better than average control grade. Interestingly again, this is again opposite for Mechatronics students in summer semester 2007. As the participants who participated in only software tools are lower than those who participated in “pop quizzes or group work” and software tools, more weight will be given to those in latter category (category 4). With this the following 2 conclusions are drawn. Firstly, as discussed in Hypothesis 11, CC SWT is more effective than LC SWT. The difference of effort for LC SWT and CC SWT is 3.4 hours extra for CC SWT. The second conclusion is methods implemented in the class have more impact. This is visible in the average grade of Computer Science students. The students who did not participate in LC SWT or CC SWT did not do poorly.

From the 2 group works implemented, GWA is more suitable for Computer Science students and GWB for Mechatronics students. These to points are elaborated in Hypothesis 8 and Hypothesis 9 respectively. One reason for GWB positively impacting the Mechatronics students might be the same software that is required in GWB is also implemented in CC SWT SS2009.

The fact that GWB is implemented is to provide and opportunity for students to work through the whole life cycle process but according to the job distribution in each group, this objective is not achieved. Hypothesis 10 also proved that the benefit of students working in more than one phase of the lifecycle is not evident. As the hours spent for implementing GWB is higher than GWA, 44 hours as compared to 16 hours, it is perhaps more desirable to implement GWA than GWB. A possible work around is to use the idea from GWB, where a few groups will create the requirements of a new systems, and this information be passed to other groups for design, followed by another group for the development, and another for the testing. This way, the group size will be reduced to half, with perhaps 5 students in each group. This might also encourage higher participation from the Computer Science students.

Of all the methods implemented, exercise requires the highest effort. According to the discussion for Hypothesis 14, both Computer Science and Mechatronics students appreciate the implementation of exercise. However, the impact of exercise on the final exam grade cannot be evaluated as exercise is not optional. On top of that the schedule for lecture and exercise is not made officially known to the students. The implementation of 5IMPLe is not successful in this research.

The implementation of bench scale equipment tends to save class preparation time. In the case of using the same bench scale equipment in ES2 exercise and software tools course saves about 12 hours of preparation. As discussed in Hypothesis 1, the course evaluations showed that the bench scale equipment helps students to understand the course content better. However, this cannot be directly connected to the final exam’s grades.

The learning behaviour between Computer Science and Mechatronics students do differ. From the difference in summer semester 2009, the Computer Science students seem to be more “studious” where as the Mechatronics students are more engaged in class participation. The attendance for Mechatronics students is also higher than Computer Science students (Table 6.4). However, as discussed in section 6.6.7, the main objective for both groups of students is to do well for their final exam.

8 Summary and Future Work

This research work presented the different methods that can be implemented in a large interdisciplinary class environment. The three main challenges focused in this research are students from different back grounds have different cognitive mind set, the growing intake of students by institute of higher education resulting large class environment, and the limitation of resources to conduct the class. There are different researches conducted to tackle each of the challenges individually. However, in this research all these three challenges are the boundaries that need to be considered together.

Different works have also exposed on the pro and cons of implementing a certain method. However, it is also necessary to study methods that are specifically relevant to a group of discipline. This research compares the impact of methods implemented between Computer Science students and Mechatronics students. Motivating lecture is the main influence for students' attendance. One important role of the teaching instructor is to instil interest in the students, and interact with the students. Students who attend lecture more regularly also do better in exam as compared to those did not. Therefore, even though the steps to engage students in the class seem simple it has great impact. Comparing pop quiz, group work and course coupling, pop quiz is a winner for both Computer Science and Mechatronics students. This shows that the methods implemented in the class have more impact on the students as compared to the methods implemented partially outside the class or totally outside the class. When making a selection with limited resources in consideration, it is proposed that the methods implemented in the class should be given priority.

The impact of group work on the other hand varies between the two disciplines. The Computer Science students are less willing to participate in GWB as the effort required is higher and the perceived it as not worth it to put in the effort for the extra 15% point. Most Mechatronics students on the other hand participated in GWB. Course coupling in combination with other methods (group work or pop quizzes) did positively impact the grade of Mechatronics students. From this observation there is a tendency that the Computer Science students are more incline to study on their own, whereas the Mechatronics students are interested in work that involves interaction. These two points should be taken note for course design and should also be further investigated.

The growing class size and the growing focus of higher education to be a research institution leaves teaching resources very valuable [Bo04]. This research presented estimation of efforts required for the different methods. The efforts here are recorded working hours from the teaching assistant. The correct method that is suitable for either Computer Science or Mechatronics students and the method that meets the current efforts availability can be selected. With this information, a guideline on the efforts required to implement each method is provided. A reward system based on the effort required can be offered to teaching instructors who take the initiatives to improve learning environment for the students.

The course ES2 focuses on the development of an embedded system. It was expected that the Mechatronics students will do poorer than the Computer Science students as this involves the development of software. However, the Mechatronics students did better than the Computer Science students. This is observed through the grade improvement achieved by the two disciplines and the comparison with the control data. It is possible that the Mechatronics students having more experience in application situation are able to answer the exam questions that test on applications and not knowledge. More studies are required to identify if the application gap for Computer Science students by integrating more application concepts in the theoretical class. From the evaluation for lecture and exercise ratio, it is seen that the necessity to “practically” work out the questions in class, instead of just listening to the lecture is important to both group of students. For future work it might be interesting to find out if exercises really impact the students’ final exam grade.

This research aims to lay the ground work for identifying the differences between Computer Science and Mechatronics students, the comparison of various methods and its impacts, and a guideline on the effort needed to implement these methods.

Among the few aspects that can be investigated for future work are listed as below:

- The specific criteria needed to conduct a large interdisciplinary class, for example in this research it is shown that motivating lecture is an aspect that encourages students’ attendance.
- The refinement of 5IMPLe or muddy card implementations is it possible to improve students’ final grade and interest for the class by pure implementation of 5IMPLe.
- The direct impact of lecture or exercise to the students’ grades if no extra methods are implemented in the class; is it sufficient for the students to learn on their own instead of coming for classes.
- The impact of exercise on students’ final exam’s grade.
- The impact of practical courses on Computer Science students. This might provide an answer on how to encourage Computer Science students to actually develop a system.
- Comparison study if Computer Science students benefits from self study program.
- Comparison study if Mechatronics students are more inclined to interpersonal communication as compared to intrapersonal communication when learning.
- The impact of “perceived understanding” in class, students in this research mentioned that the understand ability where motivating lecture is not conducted is low. However, the average grade comparison to average control grade is not poorer than that in summer semester 2007. It is possible that the result here is also influenced by the other methods implemented. However, it is also possible that the

perceived understanding or usefulness plays only a minor role when it comes to doing well for the final exams.

These findings are not only useful for institution of higher education; it can also be used to improve training centres for students and professionals. Among the relevant policies that can be implemented based on this research are for example:

- providing basic training for teaching instructors and teaching assistants before undertaking a course to encourage motivating lecture, and
- providing additional resources or reward system for courses that implements extra methods that encourages students learning based of the effort required.

9 References

[AB05] Amey, M.J.; Brown, D.F.: *Academic Work: A Case Study of a University-Community Partnership*. In: New Directions for Teaching and Learning, 2005 - Vol. 2005(102), pp. 23-35.

[Ac97] Acar, M.: *Mechatronics challenge for the higher education world*. In: IEEE Transactions on Components, Packaging, and Manufacturing Technology - Part C, 1997 - Vol. 20(1), pp. 14-20.

[AD+08] Andrietti, V.; D´Addazio, R.; Velasco, C.: *Class Attendance and Academic Performance among Spanish Economics Students*. In: UC3M Working papers. Economics, 2008 - Vol. 09(38).

[AK+00] Anderson, L.W.; Krathwohl, D.R.; Airasian, P.W.; Cruikshank, K.A.; Mayer, R.E.; Pintrich, P.R.; Raths, J.; Wittrock, M.C.: *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Abridged Edition*. 2. Edition, Allyn & Bacon, 2000.

[AW04] Arias, J.; Walker, D.M.: *Additional Evidence on the Relationship between Class Size and Student Performance*. In: Journal of Economic Education, 2004 - Vol. Fall 2004, pp. 311-329.

[Ba04] Barrington, E.: *Teaching to Student Diversity in Higher Education: How Multiple Intelligence Theory Can Help*. In: Teaching in Higher Education, 2004 - Vol. 9(4), pp. 421-434.

[Ba09] Balzert, H.: *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering (Software Technic Textbook: Basic Concepts and Requirements of Engineering)*. 3. Edition, Spektrum Akademischer Verlag, 2009.

[Be05] Bender, K.: *Embedded Systems - qualitätsorientierte Entwicklung: Qualitätssicherung bei Embedded Software (Embedded Systems - Quality Oriented Development: Quality Control for Embedded System)*. 1. Edition, Springer, Berlin, 2005.

[BE91] Bonwell, C.C.; Eison, J.A.: *Active Learning: Creating Excitement in the Classroom*. ERIC Clearinghouse on Higher Education, George Washington University, Washington D.C., 1991.

[Bi99] Biggs, J.: *Teaching for Quality Learning at University*. Open University Press, 1999.

[Bl56] Bloom, B.S.: *Taxonomy of Educational Objectives, Handbook 1: Cognitive Domain*. 2. Edition, Addison Wesley Publishing Company, 1956.

[Bo04] Bourgeois, E.: *Higher Education and Research for the Era: Current Trends and Challenges for the Near Future - Final Report of the STRATA-ETAN Expert Group*. European Commission Research, 2004.

[Bo90] Boyer, E.L.: *Scholarship Reconsidered: Priorities of the Professoriate*. Carnagie Foundation for the Advancement of Teaching, Princeton, New Jersey, 1990.

[BS+04] Barnett, W.S.; Schulman, K.; Shore, R.: *Class Size: What's the Best Fit?*. In: National Institue for Early Education (NIEER Publications) - Preschool Policy Matters, 2004 - Vol. December 2004(9).

[BS+08] Brall, S.; Schneiders, M.; Grammatke, A.; Henning, K.; Isenhardt, I.: *A Symphony of Learning. Blended Learning for Large Student Groups*. In: Center for Learning and Knowldege Management and Department of Computer Science in Mechanical Engineering RWTH Aachen, 2008 - Vol. 2008.

[BS96] Bonwell, C.C.; Sutherland, T.E.: *The Active Learning Continuum: Choosing Activities to Engage Students in the Classroom*. In: New Directions for Teaching and Learning, 1996 - Vol. 1996(67), pp. 3-16.

[Bu97] Buttazzo, G.C.: *Hard real-time computing systems*. Springer, 1997.

[BvdB+98] Broy, M.; von der Beeck, M.; Krüger, I.: *SOFTBED: Problemanalyse für ein Großverbundprojekt "Systemtechnik Automobil - Software für eingebettete Systeme" (SOFTBED: Problem Analysis for a Large Interconnected Project "Automobile System Engineering - Software for Embedded System")*. Technische Universität München, 1998.

[Ca98] Carbone, E.L.: *Teaching large classes*. SAGE, 1998.

[Ca99] Carbone, E.: *Students Behaving Badly in Large Classes*. In: New Directions for Teaching and Learning, 1999 - Vol. 1999(77), pp. 35-43.

[CB89] Chi, M.T.H.; Bassok, M.: *Learning From Examples Via Self-Explanations*. In: Resnick, L.B. (Ed.):Knowing, Learning, and Instruction - Essays in Honor of Robert Glaser, Lawrence Erlbaum Associates Inc., 1989, pp. 251-282.

[CC02] Cooper, H.; Carlisle, C.; Gibbs, T.; Watkins, C.: *Developing an Evidence Base for Interdisciplinary Learning: A Systematic Review*. In: Blackwell Science Ltd, Journal of Advanced Nursing, 2002 - Vol. 35(2), pp. 228-237.

[CE+95] Cowan, M.A.; Ewell, B.C.; McConnell, P.: *Creating Conversations An Experiment in Interdisciplinary Team Teaching*. In: College Teaching, 1995 - Vol. 43.

[CENTUM] *Production Control Systems* - *Yokogawa*. Available at: http://www.yokogawa.com/dcs/dcs-index-en.htm , accessed 03/2010.

[Ch10] Chakraborty, S.: *Institute for Real-Time Computer Systems - Technische Universitaet Muenchen : Embedded Systems*. Available at: http://www.rcs.ei.tum.de/courses/ebs/ , accessed 05/2010.

[CHE] CHE: *Center for Higher Education Development*. Available at: http://www.che-concept.de/cms/?getObject=302&getLang=en , accessed 05/2010.

[CR00] Cooper, J.L.; Robinson, P.: *The Argument for Making Large Classes Seem Small*. In: New Directions for Teaching and Learning, 2000 - Vol. 2000(81), pp. 5-16.

[CS+05] Caspi, P.; Sangiovanni-Vincentelli, A.; Almeida, L.; Benveniste, A.; Bouyssounouse, B.; Buttazzo, G.; Crnkovic, I.; Damm, W.; Engblom, J.; Folher, G.; Garcia-Valls, M.; Kopetz, H.; Lakhnech, Y.; Laroussinie, F.; Lavagno, L.; Lipari, G.; Maraninchi, F.; Peti, P.; Puente, J.D.L.; Scaife, N.; Sifakis, J.; Simone, R.D.; Torngren, M.; Veríssimo, P.; Wellings, A.J.; Wilhelm, R.; Willemse, T.; Yi, W.: *Guidelines for a Graduate Curriculum on Embedded Software and Systems*. In: ACM Trans. Embed. Comput. Syst., 2005 - Vol. 4(3), pp. 587-611.

[CS99] Carlson, L.E.; Sullivan, J.F.: *Hands-on Engineering: Learning by Doing in the Integrated Teaching and Learning Program*. In: International Journal of Engineering Education, 1999 - Vol. 15(1), pp. 20-31.

[Da69] Dale, E.: *Audio-Visual Methods in Teaching*. International Thomson Publishing, 1969.

[Da95] Davis, J.R.: *Interdisciplinary Courses and Team Teaching*. American Council on Education, the Oryx Press, 1995.

[DD+06] Davern, M.; Davies, M.; Loi, A.: *Reports on Class Size*. Graduate Studies Committee, Faculty of Economics and Commerce, University of Melbourne, 2006.

[DD07] Davies, M.; Devlin, M.: *Interdisciplinary higher education: Implications for teaching and learning*. Center for the Study of Higher Education, 2007.

[DeltaV] Emerson: *Emerson Process Management - DeltaV, the Digital Automation System for Process Control*. Available at: http://www2.emersonprocess.com/EN-US/BRANDS/DELTAV/Pages/index.aspx , accessed 03/2010.

[DF96] Devadoss, S.; Foltz, J.: *Factors influencing student class attendance and performance*. In: International Advances in Economic Research, 1996 - Vol. 2(2), pp. 194-195.

[DIN44300] Information Technology and selected IT Applications Standards Committee: *DIN 44300 Information Processing - Concepts*. Beuth Verlag GmbH, 1981.

[DIN61131-1] Information Technology and selected IT Applications Standards Committee: *DIN 61131-1 Programmable Controllers - Part 1: General Information*. Beuth Verlag GmbH, 2004.

[DIN61131-3] Information Technology and selected IT Applications Standards Committee: *DIN 61131-3 Programmable Controllers - Part 3: Programming languages*. Beuth Verlag GmbH, 2003.

[DIN66201] Information Technology and selected IT Applications Standards Committee: *DIN 66201-1 Process Computing Systems - Terms*. Beuth Verlag GmbH, 1981.

[EB03] Erbe, H.; Bruns, F.W.: *Didactical Aspects of Mechatronics Education*. In: SICICA 2003 - Intelligent Components and Instruments for Control Alpplication, Elsevier Ltd., Oxford, 2003.

[EL+97] Edwards, S.; Lavagno, L.; Lee, E.A.; Sangiovanni-vincentelli, A.: *Design of Embedded Systems: Formal Models, Validation, and Synthesis*. In: Proceedings of the IEEE, 1997 - Vol. 85(3), pp. 366-385.

[EntArch] *Enterprise Architect - UML Design Tools and UML CASE tools for software development*. In: Enterprise Architect. Available at: http://www.sparxsystems.com.au/products/ea/index.html , accessed 10/2009.

[FB16+06] FB16, University Kassel: *Lehrveranstaltungshandbuch für den Bachelorstudiengang Informatik des Fachbereiches Elektrotechnik/Informatik der Universität Kassel*. 2006.

[Fo94] Follman, J.: *The Conundrum of Class Size at the College Level*. In: College Quaterly, 1994 - Vol. 2(1).

[FV07] Friedrich, D.; Vogel-Heuser, B.: *Benefit of system modelling in automation and control education*. In: American Control Conference (ACC), New York City, USA, 2007, pp. 11-13.

[Ga93] Gardner, H.E.: *Frames Of Mind: The Theory Of Multiple Intelligences*. 10. Edition, Basic Books, 1993.

[Ge92] Geske, J.: *Overcoming the Drawbacks of the Large Lecture Class*. In: College Teaching, 1992 - Vol. 40(4), pp. 151-154.

[GG+07] Gabriel, R.; Gersch, M.; Weber, P.; Venghaus, C.: *Blended Learning Engineering: Der Einfluss von Lernort und Lernmedium auf Lernerfolg und Lernzufriedenheit — Eine evaluationsgestützte Untersuchung (Blended Learning Engineering: The Influence of Learning Location and Medium on Learning Sucess and Satisfaction - an Evaluation Supported Research)*. In: Neue Trends im E-Learning, 2007, pp. 75-92.

[GG76] Goldschmid, B.; Goldschmid, M.L.: *Peer Teaching in Higher Education: A Review*. In: Higher Education, 1976 - Vol. 1976(5), pp. 9-33.

[GH01] Grimheden, M.; Hanson, M.: *What is Mechatronics? Proposing a Didactical Approach to Mechatronis*. In: 1st Baltic Sea Workshop on Education in Mechatronics, 2001.

[GH05] Grimheden, M.; Hanson, M.: *Mechatronics--the evolution of an academic discipline in engineering education*. In: Mechatronics, 2005 - Vol. 15(2), pp. 179-192.

[GL+96] Gibbs, G.; Lucas, L.; Simonite, V.: *Class size and student performance: 1984–94*. In: Studies in Higher Education, 1996 - Vol. 21(3), p. 261.

[GP06] Graber, G.C.; Pionke, C.D.: *A Team-Taught Interdisciplinary Approach to Engineering Ethics*. In: Science and Engineering Ethics, 2006 - Vol. 12(2), pp. 313-320.

[Gr88] Grunde, U.: *Petri-Netze: Die Grundideen (Petri Nets: The Basic Ideas)*.1988.

[GRS06] Gausemeier, J.; Rammig, F.; Schäfer, W.: *Selbstoptimierung im Maschinenbau (Self-Optimisation in Mechanical Engineering)*. In: Forschungs Forum Paderborn, 2006 - Vol. 9(2006), pp. 50-55.

[GT05] Grimheden, M.; Törngren, M.: *How Should Embedded Systems be Taught? Experiences and Snapshots from Swedish Higher Engineering Education*. In: SIGBED Rev., 2005 - Vol. 2(4), pp. 34-39.

[Ha03] Haberyan, K.A.: *Do Weekly Quizzes Improve Student Performance on General Biology Exams?*. In: The American Biology Teacher, 2003 - Vol. 65(2), pp. 110-114.

[Ha03] Hartmut Ernst: *Grundlagen und Konzepte der Informatik (Basics and Concepts of Computer Science)*. Viewveg Verlagsgesellschaft, 2003.

[Ha86] Hamilton, L.C.: *Microcomputer Graphics for Statistical Analysis*. In: Social Science Computer Review, 1986 - Vol. 4(2), pp. 181-193.

[HH+00] Hatley, D.J.; Hruschka, P.; Pirbhai, I.: *Process for System Architecture and Requirements Engineering*. Dorset House Publishing, 2000.

[HL+09] Heuer, A.; Lauenroth, K.; Stricker, V.; Pohl, K.: *Entiwicklung eingebetteter Software in einem Softwarepraktikum mit Lego Mindstorms (Development of Embedded System Software in a Software Practical Course using Lego Mindstorms)*. In: Software Engineering im Unterricht der Hochschulen, dpunkt.verlag, Heidelberg, 2009, pp. 115-130.

[HV05] Holleczek, P.; Vogel-Heuser, B.: *Echtzeitaspekte bei der Koordinierung Autonomer Systeme (Realtime Aspects in Co-ordination of an Autonomous System)*. Springer, 2005.

[HW+02] Hall, S.R.; Waitz, I.; Brodeur, D.R.; Soderholm, D.H.; Nasr, R.: *Adoption Of Active Learning In A Lecture-Based Engineering Class*. In: 32nd ASEE/IEEE Frontiers in Education Conference, Boston, MA, 2002, pp. T2A9-T2A15.

[IEEE802.3] IEEE 802.3 Working Group: *IEEE 802.3 ETHERNET*. Available at: http://www.ieee802.org/3/ , accessed 04/2010.

[ILC150a] Phoenix Contact: *PHOENIX CONTACT Components and System Automation*. Available at: http://www.phoenixcontact.de/news/194_22872.htm , accessed 02/2010.

[Ja07] James, R.: *Six educational issues facing universities and academic leaders*. Seminar Series on Mass Higher Education in UK and International Contexts, Surrey, 2007.

[JK+08] Jeschke, S.; Knipping, L.; Liebhardt, M.; Müller, F.; Vollmer, U.; Wilke, M.; Yan, X.: *Integrating Medical Robotics in the Robinson Program*. In: The 8th IEEE International Conference on Advanced Learning Technologies, IEEE Computer Society, Los Alamitos (CA), USA, 2008, pp. 524-528.

[KA+07] Kenny, S.S.; Alberts, B.; Booth, W.C.; Glaser, M.; Glassick, C.E.; Ikenberry, S.O.; Jameison, K.; O'Neil, R.M.; Reid-Wallace, C.; Tien, C.; Yang, C.N.: *Reinventing Undergraduate Education: A Blueprint for America's Research Universities*. The Boyer Commission on Educating Undergraduates in the Research University, State University of New York, Stony Brook, New York, 2007.

[KC+05] Koopman, P.; Choset, H.; Gandhi, R.; Krogh, B.; Marculescu, D.; Narasimhan, P.; Paul, J.M.; Rajkumar, R.; Siewiorek, D.; Smailagic, A.; Steenkiste, P.; Thomas, D.E.; Wang, C.: *Undergraduate embedded system education at Carnegie Mellon*. In: ACM Trans. Embed. Comput. Syst., 2005 - Vol. 4(3), pp. 500-528.

[KC05] Kennedy, G.E.; Cutts, Q.I.: *The association between students' use of an electronic voting system and their learning outcomes*. In: Journal of Computer Assisted Learning, 2005 - Vol. 21(4), pp. 260-268.

[KD+08] Kokkelenberg, E.C.; Dillon, M.; Christy, S.M.: *The effects of class size on student grades at a public university*. In: Economics of Education Review, 2008 - Vol. 27(2), pp. 221-233.

[Ke01] Kezar, A.: *Theory of Multiple Intelligences: Implications for Higher Education*. In: Innovative Higher Education, 2001 - Vol. 26(2), pp. 141-154.

[Kl96] Klein, J.T.: *Crossing Boundaries: Knowledge, Disciplinarities, and Interdisciplinarities*. Univ Pr of Virginia, 1996.

[KN02] Kessler, C.; Nadjm-Tehrani, S.: *Mid-term course evaluations with muddy cards*. In: Proceedings of the 7th annual conference on Innovation and technology in computer science education, Aarhus, Denmark, 2002, pp. 233-233.

[Ko10] Kowalewski, S.: *Einführung in eingebettete Systeme - Lehrstuhl Informatik 11 - Embedded Software Laboratory (Introduction to Embedded System - Chair Informatics 11 - Embedded Software Laboratory)*. Available at: http://embedded.rwth-aachen.de/doku.php?id=lehre:sose10:ites , accessed 05/2010.

[Ku07] Kuo, W.: *Editorial: How Reliable is Teaching Evaluation? The Relationship of Class Size to Teaching Evaluation Scores*. In: IEEE Transactions on Reliability, 2007 - Vol. 56(2), pp. 178-181.

[KV09] Katzke, U.; Vogel-Heuser, B.: *Vergleich der Anwendbarkeit von UML und UML-PA in der anlagennahen Softwareentwicklung der Automatisierungstechnik (Comparison on the applicability of UML and UML-PA in the Software Development of an Automation System)*. In: at-Automatisierungstechnik, 2009 - Vol. 57(7), pp. 332-340.

[LC06] Li, J.; Chok, P.: *Effective Software Practices To Leverage Software Development Process in Sand-Control Engineering Software-A Case Study*. In: Proceedings of SPE Annual Technical Conference and Exhibition, San Antonio, Texas, USA, 2006.

[Le04] Leohold, J.: *Prüfungsordnung für den Bachelorstudiengang Informatik des Fachbereiches Elektrotechnik/Informatik der Universität Kassel vom 9. November 2004*. 2004.

[Lego] *LEGO.com MINDSTORMS : Home*. In: Lego Mindstorms. Available at: http://mindstorms.lego.com/en-us/default.aspx , accessed 04/2010.

[LG99] Lauber, R.; Göhner, P.: *Prozeßautomatisierung. 1. Automatisierungssysteme und -strukturen, Computer- und Bussysteme für die Anlagen- und Produktautomatisierung, Echtzeitprogrammierung und Echtzeitbetriebssysteme, Zuverlässigkeits- und Sicherheitstechnik*. Springer, 1999.

[Li02] Liggesmeyer, P.: *Software-Qualität: Testen, Analysieren und Verifizieren von Software*. 2. Edition, Spektrum Akademischer Verlag, 2009.

[LP06] Lavagno, L.; Passerone, C.: *Design of Embedded Systems*. In: Zurawski, R. (Ed.):Embedded systems handbook, CRC Press, Boca Raton, Florida, 2006, pp. 3-1--3-24.

[Ma00] MacGregor, J.: *Restructuring Large Classes to Create Communities of Learners*. In: New Directions for Teaching and Learning, 2000 - Vol. 2000(81), pp. 25-46.

[Ma05] Marwedel, P.: *Towards laying common grounds for embedded system design education*. In: SIGBED Rev., 2005 - Vol. 2(4), pp. 25-28.

[Ma06] Marwedel, P.: *Embedded System Design*. Springer, Dordrecht, 2006.

[Magic Draw] No Magic: *UML, SysML, BPMN, UPDM, software modelling tool MagicDraw and requirements with storyboarding solution from No Magic*. Available at: http://www.magicdraw.com/ , accessed 02/2010.

[Mi08] Mitusi, H.; Kambe, H.; Endo, S.; Koizumi, H.: *A Student Experiment Method for Learning the Basics of Embedded Software Development Including HW/SW Co-design*. In: 22nd International Conference on Advanced Information Networking and Applications - Workshops, IEEE Computer Society, 2008, pp. 31-37.

[Mi99] Mills-Jones, A.: *Active Learning in IS Education: Choosing Effective Strategies for Teaching Large Classes in Higher Education*. In: 10th Australasian Conference, 1999, pp. 622-634.

[MID] *Innovator MID GmbH*. In: MID the modelling company. Available at: http://www.mid.de/MID-Homepage.startseite.0.html?&L=0 , accessed 03/2010.

[Mo89] Mosteller, F.: *The "Muddiest Point in the Lecture"as a Feedback Device*. In: On Teaching and Learning: The Journal of the Harvard-Danforth Center, 1989 - Vol. 3, pp. 10-21.

[MS93] Murata, S.; Stern, S.: *Technology Education in Japan*. In: Journal Technology of Education (JTE), 1993 - Vol. 5(1).

[Mu05] Muppala, J.K.: *Experience with an embedded systems software course*. In: SIGBED Rev., 2005 - Vol. 2(4), pp. 29-33.

[MW96] Mccabe, T.J.; Watson, A.H.: *Structured Testing: A testing methodology using cyclomatic complexity metric*. In: NIST Special Publication, 1996(500-235), pp. 235-500.

[Ne94] Newell, W.H.: *Designing Interdisciplinary Courses*. In: New Directions for Teaching and Learning, 1994 - Vol. 1994(58), pp. 35-51.

[OMGUML] OMG UML: *Object Management Group - UML*. 2009. Available at: http://www.uml.org/ , accessed 09/2009.

[Pa08] Patig, S.: *A practical guide to testing the understandability of notations*. In: Proceedings of the fifth on Asia-Pacific conference on conceptual modelling - Volume 79, Australian Computer Society, Inc., Wollongong, NSW, Australia, 2008, pp. 49-58.

[PCS7] Siemens: *Distributed Control System SIMATIC PCS 7 - Siemens*. Available at: http://www.automation.siemens.com/w2/automation-technology-distributed-control-system-simatic-pcs-7-1075.htm , accessed 03/2010.

[PD07] Pears, A.; Daniels, M.: *A national center for pedagogical development in technology education from a student oriented perspective*. In: Meeting the Growing Demand for Engineers and Their Educators 2010-2020 International Summit, 2007 IEEE, 2007, pp. 1-8.

[Po09] Polat, M.: *Is peer assessment worth trying?*. In: International Journal of Arts and Sciences, 2009.

[PR05] Pak, S.; Rho, E.; Chang, J.; Kim, M.H.: *Demand-driven curriculum for embedded system software in Korea*. In: SIGBED Rev., 2005 - Vol. 2(4), pp. 15-19.

[PROFIBUS] PI International: *PI - PROFIBUS & PROFINET International*. Available at: http://www.profibus.com/ , accessed 04/2010.

[PV09] Pasik-Duncan, B.; Verleger, M.: *Education and Qualification for Control and Automation*. In: Nof, S. (Ed.):Springer Handbook of Automation, 2009, pp. 767-778.

[Re09] Reichardt, J.: *Lehrbuch Digitaltechnik: Eine Einführung mit VHDL*. Oldenbourg, 2009.

[ReqPro] *IBM - Rational RequisitePro - Software*. In: IBM-Rational RequisitePro - Software. Available at: http://www-01.ibm.com/software/awdtools/reqpro/ , accessed 10/2009.

[Ro93] Romer, D.: *Do Students Go to Class? Should They?*. In: Journal of Economic Perspectives, 1993 - Vol. 7(3), pp. 167-74.

[RS+05] Ricks, K.G.; Stapleton, W.A.; Jackson, D.J.: *An embedded systems course and course sequence*. In: Proceedings of the 2005 workshop on Computer architecture education, ACM, Madison, Wisconsin, 2005, p. 8.

[RTOS-UH] RTOS-UH: *RTOS-UH*. Available at: http://www.rtos.irt.uni-hannover.de/rtos-gb.html , accessed 02/2010.

[SA03] Sanz, R.; Arzen, K.: *Trends in software and control*. In: Control Systems Magazine, IEEE, 2003 - Vol. 23(3), pp. 12-15.

[SB02] Sargison, J.; Bullen, F.; McCulloch, J.: *Development of a cross discipline, experiential based, flexible delivery unit*. In: International Conference on Engineering Education, Manchester, 2002, pp. 1-4.

[Sc05] Schulmeister, R.: *Lernplattformen für das virtuelle lernen (Learning Platforms for Virtual Learning)*. Oldenbourg Wissenschaftsverlag, 2005.

[Sc07] Schäfer, C.: *On the Modularity of Manufacturing Systems - The Challenges to Be Faced in a Future of Fundamental Change*. In: IEEE Industrial Electronics Magazine, 2007(Fall), pp. 20-27.

[Se05] Seviora, R.E.: *A curriculum for embedded system engineering*. In: ACM Trans. Embed. Comput. Syst., 2005 - Vol. 4(3), pp. 569-586.

[SE07] Sommers, D.R.; Engbretson, G.: *Teaching “Real-World” Design in the Classroom*. In: IEEE Education - Meeting the Growing Demand for Engineers and Their Educators 2010-2020 International Summit, 2007, pp. 1-8.

[SK07] Schmid, J.; Knoll, G.: *Neufassung der Prüfungsordnung für den gestuften Diplomstudiengang Mechatronik der Universität Kassel*. 2007.

[SL+09] Sim, T.Y.; Li, F.; Vogel-Heuser, B.: *Benefits of an Interdisciplinary Modular Concept in Automation of Machine and Plant Manufacturing*. In: 13th IFAC Symposium on Information Control Problems in Manufacturing Submission, Germany, 2009.

[SM02] Shooter, S.; McNeill, M.: *Interdisciplinary Collaborative Learning in Mechatronics at Bucknell University*. In: Journal of Engineering Education, 2002(July 2002), pp. 339-344.

[Sm96] Smith, K.A.: *Cooperative Learning: Making "Groupwork" Work*. In: New Directions for Teaching and Learning, 1996 - Vol. 1996(67), pp. 71-82.

[Sn06] Snooks, M.K.: *Practice Tests: a Practical Teaching Method*. In: Essays on Teaching Excellence - Toward the Best in the Academy, 2005 - Vol. 17(7).

[SP05] Sangiovanni-Vincentelli, A.L.; Pinto, A.: *Embedded system education: a new paradigm for engineering schools?*. In: SIGBED Rev., 2005 - Vol. 2(4), pp. 5-14.

[SPSS] SPSS.com: *SPSS, Data Mining, Statistical Analysis Software, Predictive Analysis, Predictive Analytics, Decision Support Systems*. Available at: http://www.spss.com/ , accessed 01/2010.

[SS+05] Smith, K.A.; Sheppard, S.D.; Johnson, D.W.; Johnson, R.T.: *Pedagogies of Engagement: Classroom-Based Practices*. In: Journal of Engineering Education, 2005 - Vol. 94(1), pp. 1-15.

[St74] Stanton, H.E.: *Teaching methods and student personality - The search for an elusive interaction*. In: Instructional Science, 1974 - Vol. 2(4), pp. 477-501.

[StZeit] Zeit: *Studiengänge - Was Studieren? Wo Studieren? Wie Studieren | ZEIT ONLINE*. Available at: http://studiengaenge.zeit.de/ , accessed 05/2010.

[SV09] Sim, T.Y.; Vogel-Heuser, B.: *Lessons learnt from Interdisciplinary Class between Mechatronics and Computer Science students*. In: International Journal of Arts and Sciences, 2009.

[SV10] Sim, T.Y.; Vogel-Heuser, B.: *Reviews and Findings on Implementing Active Learning in a Large Class Environment for Mechatronics and Computer Science students*. In: IEEE EDUCON Education Engineering 2010 - The Future of Global Learning Engineering Education, 2010.

[SW09] Schütz, D.; Wannagat, A.: *Domänenspezifische Modellierung für automatisierungstechnische Anlagen mit Hilfe der SysML*. In: Automatisierungstechnische Praxis, 2009 - Vol. 50(3), pp. 54-62.

[SysML] OMG SysML: *OMG SysML*. 2009. Available at: http://www.omgsysml.org/ , accessed 12/2009.

[Ta03] Tanenbaum, A.S.: *Computer networks*. Prentice Hall PTR, 2003.

[To02] Tomizuka, M.: *Mechatronics: from the 20th to 21st century*. In: Control Engineering Practice, 2002 - Vol. 10(8), pp. 877-886.

[TrySim] *Machine Simulation for PLC Programmers - TrySim*. In: Machine Simulation for PLC Programmers - TrySim 2009. Available at: http://www.cephalos.de/home.htm , accessed 09/2009.

[Va07] Valles-Barajas, F.: *A requirements engineering process for control engineering software*. In: Innovations in Systems and Software Engineering, 2007 - Vol. 3(4), pp. 217-227.

[VDI90] VDI: *Empfehlung des VDI zur Integration fachübergreifender Studieninhalte in das Ingenieurstudium (VDI Proposals for Interdisciplinary Course Integration for Engineering Study)*. VDI, Düsseldorf, Germany, 1990.

[Vo03] Vogel-Heuser, B.: *Systems Software Engineering. Angewandte Methoden des Systementwurfs für Ingenieure (Applied System Development Methods for Engineers)*. 1. Edition, Oldenbourg Wissenschaftsverlag, 2003.

[Vo08] Vogel-Heuser, B.: *Automation & Embedded Systems*. Oldenbourg Industrieverlag, 2008.

[Vo09] Nof, S.Y.; Vogel-Heuser, B. (Eds.): *Automation in Wood and Paper Industry*. In: Springer Handbook of Automation, Springer, Berlin, 2009, pp. 1095-1025.

[VON++] *Visual Object Net Analysis*. In: VON: Visual Object Net++. Available at: http://www.techfak.uni-bielefeld.de/~mchen/BioPNML/Intro/VON.html , accessed 10/2009.

[VW07] Vogel-Heuser, B.; Wannagat, A.: *Modulares Engineering und Wiederverwendung mit CoDeSys V3 (Modular Engineering and Reusability using CoDeSys V3)*. 1. Edition, Oldenbourg Industrieverlag, 2007.

[WB05] Wörn, H.; Brinkschulte, U.: *Echtzeitsysteme (Real-time System)*. Springer, 2005.

[We06] Weilkiens, T.: *System Engineering with SysML/UML*. 1. Edition, dpunkt Verlag, 2006.

[WT+01] Wikander, J.; Torngren, M.; Hanson, M.: *The science and education of mechatronics engineering*. In: Robotics & Automation Magazine, IEEE, 2001 - Vol. 8(2), pp. 20-26.

[WW+08] Weiker, N.; Weicker, K.; Korzenek, C.: *Bewertung von didaktischen Konzeptionen in der Informatiklehre (Evaluation of Didactic Concepts in Computer Science Teaching)*. In: Brinda, T.; Fothe, M.; Hubwieser, P.; Schülter, K. (Eds.):Didaktik der Informatik – Aktuelle Forschungsergebnisse 5. Workshop der GI-Fachgruppe "Didaktik der Informatik", Köllen Druck + Verlag GmbH, Bonn, 2008, pp. 97-106.

[YT+05] Yamamoto, M.; Tomiyama, H.; Takada, H.; Agusa, K.; Mase, K.; Kawaguchi, N.; Honda, S.; Kaneko, N.: *NEXCESS: Nagoya university extension courses for embedded software specialists*. In: SIGBED Rev., 2005 - Vol. 2(4), pp. 20-24.

[Zu06] Zurawski, R.: *Embedded systems handbook*. CRC Press, Boca Raton, Florida, 2006.

[ZVEI04] ZVEI: *Grundlegende Positionen des ZVEI zur Ingenieurausbildung der Zukunft (The Foundation Position of ZVEI towards Engineering Education for the Future)*. Zentralverband Elektrotechnik- und Elektronikindustrie e.V. 2004.

10 Attachment

10.1 Statistical Data – Final Exams' Grades

10.1.1 Descriptive Statistics

Descriptive Statistics for School's Grade

Descriptive Statistics Category = School; Discipline = Computer Science					
	N	Minimum	Maximum	Mean	Std. Deviation
WS0607	298	1.0	5.0	3.436	1.2232
SS2007	139	1.0	5.0	2.900	1.0476
WS0708	258	1.0	5.0	3.322	1.2917
SS2008	147	1.0	5.0	3.354	1.1637
WS0809	108	1.0	5.0	3.392	1.3670
SS2009	129	1.0	5.0	3.468	1.3187

Descriptive Statistics Category = School; Discipline = Mechatronics					
	N	Minimum	Maximum	Mean	Std. Deviation
WS0607	78	1.0	5.0	3.082	1.1320
SS2007	91	1.0	5.0	2.924	1.1261
WS0708	140	1.0	5.0	2.800	1.1000
SS2008	149	1.0	5.0	2.951	1.0771
WS0809	173	1.0	5.0	3.293	1.3332
SS2009	123	1.0	5.0	2.493	1.4392

Descriptive Statistics for Computer Science Students' Grade

Descriptive Statistics Category = Class; Discipline = Computer Science; Participants = No; Grade = NEP					
	N	Minimum	Maximum	Mean	Std. Deviation
WS0607	71	1.0	5.0	3.501	1.3818
SS2007	9	3.3	5.0	4.144	0.6966
WS0708	25	1.7	5.0	3.784	0.8601
SS2008	20	2.0	5.0	4.070	0.9251
WS0809	18	1.0	5.0	3.506	1.2027
SS2009	22	1.0	5.0	2.891	1.3140

Descriptive Statistics Category = Class; Discipline = Computer Science; Participants = Yes; Grade = NEP					
	N	Minimum	Maximum	Mean	Std. Deviation
WS0607					
SS2007	41	1.0	4.0	3.0290	0.8878
WS0708	27	1.3	5.0	3.2480	1.0024
SS2008	26	1.0	5.0	2.7420	1.1673
WS0809	29	1.0	5.0	3.0690	1.3696
SS2009	34	1.7	5.0	3.6590	1.0252

Descriptive Statistics Category = Class; Discipline = Computer Science; Participants = Yes; Grade = WEP					
	N	Minimum	Maximum	Mean	Std. Deviation
WS0607					
SS2007	41	1.0	4.0	2.593	0.8748
WS0708	27	1.0	5.0	3.052	1.0184
SS2008	26	1.0	5.0	2.162	1.1486
WS0809	29	1.0	5.0	2.638	1.4115
SS2009	34	1.0	5.0	2.824	1.1855

Descriptive Statisics for Mechatronics Students' Grade

Descriptive Statistics Category = Class; Discipline = Mechatronics; Participants = No; Grade = NEP					
	N	Minimum	Maximum	Mean	Std. Deviation
WS0607	25	1.0	5.0	3.932	1.3247
SS2007	5	1.0	4.0	2.660	1.1082
WS0708	14	2.7	5.0	4.193	0.7205
SS2008	7	2.7	5.0	3.857	0.8772
WS0809	14	2.3	5.0	4.114	0.9281
SS2009	7	1.0	5.0	3.571	1.5119

Descriptive Statistics Category = Class; Discipline = Mechatronics; Participants = Yes; Grade = NEP					
	N	Minimum	Maximum	Mean	Std. Deviation
WS0607					
SS2007	19	1.7	5.0	3.653	0.9845
WS0708	20	1.0	5.0	2.535	0.9366
SS2008	20	1.0	5.0	2.400	1.2703
WS0809	15	1.0	5.0	2.320	1.4047
SS2009	17	1.0	5.0	2.394	1.1448

Descriptive Statistics Category = Class; Discipline = Mechatronics; Participants = Yes; Grade = WEP					
	N	Minimum	Maximum	Mean	Std. Deviation
WS0607					
SS2007	19	1.3	4.0	3.158	0.8821
WS0708	20	1.0	5.0	2.260	1.0748
SS2008	20	1.0	5.0	1.915	1.2717
WS0809	15	1.0	5.0	1.973	1.3019
SS2009	17	1.0	5.0	1.724	1.0533

10.1.2 T-Test

T-Test between Computer Science Participants in Pop Quizzes or Group Work (using WEP average grade); and School - Control Data

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
Grade SS2007	WEP Average Grade	Class	41	2.593	0.8748	0.1366
	Average Control Grade	School	139	2.900	1.0476	0.0889
Grade WS0708	WEP Average Grade	Class	27	3.052	1.0184	0.1960
	Average Control Grade	School	258	3.322	1.2917	0.0804
Grade SS2008	WEP Average Grade	Class	26	2.162	1.1486	0.2253
	Average Control Grade	School	147	3.354	1.1637	0.0960
Grade WS0809	WEP Average Grade	Class	29	2.638	1.4115	0.2621
	Average Control Grade	School	108	3.392	1.3670	0.1315
Grade SS2009	WEP Average Grade	Class	34	2.824	1.1855	0.2033
	Average Control Grade	School	129	3.468	1.3187	0.1161

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
Grade SS2007	Eq. var. assumed	0.975	0.325	-1.710	178	0.089	-0.3073	0.1797	-0.6620	0.0474
	Eq. var. not assumed			-1.886	77.01	0.063	-0.3073	0.1630	-0.6318	0.0172
Grade WS0708	Eq. var. assumed	3.503	0.062	-1.051	283	0.294	-0.2699	0.2567	-0.7751	0.2354
	Eq. var. not assumed			-1.274	35.389	0.211	-0.2699	0.2119	-0.6998	0.1601
Grade SS2008	Eq. var. assumed	0.009	0.925	-4.825	171	0.000	-1.1922	0.2471	-1.6800	-0.7044
	Eq. var. not assumed			-4.869	34.706	0.000	-1.1922	0.2449	-1.6894	-0.6950
Grade WS0809	Eq. var. assumed	0.107	0.744	-2.618	135	0.010	-0.7537	0.2879	-1.323	-0.1844
	Eq. var. not assumed			-2.570	43.162	0.014	-0.7537	0.2933	-1.3451	-0.1624
Grade SS2009	Eq. var. assumed	3.658	0.058	-2.587	161	0.011	-0.6447	0.2492	-1.1367	-0.1526
	Eq. var. not assumed			-2.754	56.483	0.008	-0.6447	0.2341	-1.1136	-0.1758

T-Test between Mechatronics Participants in Pop Quizzes or Group Work (using WEP average grade); and School - Control Data

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
Grade SS2007	WEP Average Grade	Class	19	3.158	0.8821	0.2024
	Average Control Grade	School	91	2.924	1.1261	0.1180
Grade WS0708	WEP Average Grade	Class	20	2.260	1.0748	0.2403
	Average Control Grade	School	140	2.800	1.1000	0.0930
Grade SS2008	WEP Average Grade	Class	20	1.915	1.2717	0.2844
	Average Control Grade	School	149	2.951	1.0771	0.0882
Grade WS0809	WEP Average Grade	Class	15	1.973	1.3019	0.3362
	Average Control Grade	School	173	3.293	1.3332	0.1014
Grade SS2009	WEP Average Grade	Class	17	1.724	1.0533	0.2555
	Average Control Grade	School	123	2.493	1.4392	0.1298

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
Grade SS2007	Eq. var. assumed	0.924	0.339	0.851	108	0.397	0.2337	0.2747	-0.3109	0.7783
	Eq. var. not assumed			0.998	31.601	0.326	0.2337	0.2343	-0.2437	0.7112
Grade WS0708	Eq. var. assumed	0.584	0.446	-2.059	158	0.041	-0.5400	0.2622	-1.0579	-0.0221
	Eq. var. not assumed			-2.096	25.035	0.046	-0.5400	0.2577	-1.0707	-0.0093
Grade SS2008	Eq. var. assumed	0.859	0.355	3.951	167	0.000	1.0360	0.2622	0.5184	1.5536
	Eq. var. not assumed			3.480	22.808	0.002	1.0360	0.2977	0.4198	1.6522
Grade WS0809	Eq. var. assumed	0.442	0.507	-3.684	186	0.000	-1.3197	0.3582	-2.0264	-0.6130
	Eq. var. not assumed			-3.759	16.651	0.002	-1.3197	0.3511	-2.0617	-0.5778
Grade SS2009	Eq. var. assumed	6.995	0.009	-2.126	138.00	0.035	-0.7700	0.3622	-1.4862	-0.0537
	Eq. var. not assumed			-2.687	25.103	0.013	-0.7700	0.2865	-1.3600	-0.1800

T-Test between Computer Science Students who participated in Pop Quizzes or Group Work; and Non-Participants (Participants EP = 1; Non-Participants EP = 0)

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS2007	NEP Average Grade	0	9	4.144	0.6966	0.2322
	NEP Average Grade	1	41	3.029	0.8878	0.1386
NEP WS0708	NEP Average Grade	0	25	3.784	0.8601	0.1720
	NEP Average Grade	1	27	3.248	1.0024	0.1929
NEP SS2008	NEP Average Grade	0	20	4.070	0.9251	0.2069
	NEP Average Grade	1	26	2.742	1.1673	0.2289
NEP WS0809	NEP Average Grade	0	18	3.506	1.2027	0.2835
	NEP Average Grade	1	29	3.069	1.3696	0.2543
NEP SS2009	NEP Average Grade	0	22	2.891	1.3140	0.2801
	NEP Average Grade	1	34	3.659	1.0252	0.1758

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
NEP SS2007	Eq. var. assumed	1.418	0.240	3.527	48	0.001	1.1152	0.3162	0.4795	1.7508
	Eq. var. not assumed			4.123	14.356	0.001	1.1152	0.2704	0.5365	1.6939
NEP WS0708	Eq. var. assumed	0.954	0.333	2.061	50	0.045	0.5359	0.2600	0.0136	1.0581
	Eq. var. not assumed			2.073	49.726	0.043	0.5359	0.2585	0.0166	1.0551
NEP SS2008	Eq. var. assumed	1.536	0.222	4.174	44	0.000	1.3277	0.3181	0.6866	1.9688
	Eq. var. not assumed			4.303	43.944	0.000	1.3277	0.3085	0.7058	1.9495
NEP WS0809	Eq. var. assumed	0.989	0.325	1.111	45	0.272	0.4366	0.3928	-0.3545	1.2277
	Eq. var. not assumed			1.146	39.747	0.258	0.4366	0.3808	-0.3333	1.2065
NEP SS2009	Eq. var. assumed	2.222	0.142	-2.449	54	0.018	-0.7679	0.3136	-1.3967	-0.1392
	Eq. var. not assumed			-2.322	37.134	0.026	-0.7679	0.3307	-1.4380	-0.0978

T-Test between Mechatronics Students who participated in Pop Quizzes or Group Work; and Non-Participants (Participants EP = 1; Non-Participants EP = 0)

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS2007	NEP Average Grade	0	5	2.660	1.1082	0.4956
	NEP Average Grade	1	19	3.653	0.9845	0.2259
NEP WS0708	NEP Average Grade	0	14	4.193	0.7205	0.1926
	NEP Average Grade	1	20	2.535	0.9366	0.2094
NEP SS2008	NEP Average Grade	0	7	3.857	0.8772	0.3316
	NEP Average Grade	1	20	2.400	1.2703	0.2840
NEP WS0809	NEP Average Grade	0	14	4.114	0.9281	0.2480
	NEP Average Grade	1	15	2.320	1.4047	0.3627
NEP SS2009	NEP Average Grade	0	7	3.571	1.5119	0.5714
	NEP Average Grade	1	17	2.394	1.1448	0.2777

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
NEP SS2007	Eq. var. assumed	0.003	0.955	-1.959	22	0.063	-0.9926	0.5067	-2.0435	0.0582
	Eq. var. not assumed			-1.823	5.779	0.120	-0.9926	0.5446	-2.3377	0.3525
NEP WS0708	Eq. var. assumed	0.435	0.514	5.562	32	0.000	1.6579	0.2981	1.0507	2.2650
	Eq. var. not assumed			5.827	31.646	0.000	1.6579	0.2845	1.0781	2.2376
NEP SS2008	Eq. var. assumed	1.370	0.253	2.793	25	0.010	1.4571	0.5217	0.3827	2.5315
	Eq. var. not assumed			3.338	15.417	0.004	1.4571	0.4366	0.5287	2.3855
NEP WS0809	Eq. var. assumed	2.331	0.138	4.027	27	0.000	1.7943	0.4456	0.8800	2.7086
	Eq. var. not assumed			4.084	24.408	0.000	1.7943	0.4394	0.8882	2.7003
NEP SS2009	Eq. var. assumed	1.104	0.305	2.088	22	0.049	1.1773	0.5639	0.0079	2.3467
	Eq. var. not assumed			1.853	8.98	0.097	1.1773	0.6353	-0.2604	2.6150

T-Test between SWT Participants (Computer Science) and Control Data

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP SS2007	WEP Average Grade	Class	10	2.700	0.7318	0.2314
	Average Control Grade	School	139	2.900	1.0476	0.0889
WEP SS2008	WEP Average Grade	Class	2	3.000	2.8284	2.0000
	Average Control Grade	School	147	3.354	1.1637	0.0960
WEP SS2009	WEP Average Grade	Class	7	3.186	1.4265	0.5391
	Average Control Grade	School	129	3.468	1.3187	0.1161

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
Grade SS2007	Eq. var. assumed	1.484	0.225	-0.592	147	0.554	-0.2000	0.3376	-0.8671	0.4671
	Eq. var. not assumed			-0.807	11.833	0.436	-0.2000	0.2479	-0.7410	0.3410
Grade SS2008	Eq. var. assumed	4.978	0.027	-0.420	147	0.675	-0.3537	0.8421	-2.0180	1.3105
	Eq. var. not assumed			-0.177	1.005	0.889	-0.3537	2.0023	-25.521	24.8136
Grade SS2009	Eq. var. assumed	0.007	0.935	-0.550	134	0.583	-0.2825	0.5137	-1.2985	0.7335
	Eq. var. not assumed			-0.512	6.569	0.625	-0.2825	0.5515	-1.6042	1.0392

T-Test between SWT Participants (Mechatronics) and Control Data

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP SS2007	WEP Average Grade	Class	22	2.968	0.9245	0.1971
	Average Control Grade	School	91	2.924	1.1261	0.1180
WEP SS2008	WEP Average Grade	Class	24	2.221	1.4213	0.2901
	Average Control Grade	School	149	2.951	1.0771	0.0882
WEP SS2009	WEP Average Grade	Class	22	2.014	1.2353	0.2634
	Average Control Grade	School	123	2.493	1.4392	0.1298

Independent Samples Test

		Levene's Test for		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Diffe-	Std. Error Diffe-	95% Confidence	
									Lower	Upper
Grade SS2007	Eq. var. assumed	1.049	0.308	0.170	111	0.865	0.0440	0.2592	-0.4695	0.5575
	Eq. var. not assumed			0.192	37.638	0.849	0.0440	0.2297	-0.4212	0.5092
Grade SS2008	Eq. var. assumed	5.117	0.025	-2.939	171	0.004	-0.7302	0.2484	-1.2206	-0.2398
	Eq. var. not assumed			-2.408	27.416	0.023	-0.7302	0.3032	-1.3519	-0.1084
Grade SS2009	Eq. var. assumed	2.730	0.101	-1.469	143	0.144	-0.4799	0.3266	-1.1255	0.1658
	Eq. var. not assumed			-1.634	32.108	0.112	-0.4799	0.2936	-1.0778	0.1181

T-Test between SWT Participants (1) and Non-Participants (0)

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS2007	NEP Average NEP	1	10	3.200	0.7409	0.2343
	NEP Average NEP	0	40	3.238	1.0081	0.1594
NEP SS2008	NEP Average NEP	1	2	3.150	2.6163	1.8500
	NEP Average NEP	0	44	3.327	1.2138	0.1830
NEP SS2009	NEP Average NEP	1	7	3.857	1.0830	0.4093
	NEP Average NEP	0	49	3.286	1.2057	0.1722

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
NEP SS2007	Eq. var. assumed	1.003	0.322	-0.110	48	0.913	-0.0375	0.3407	-0.7225	0.6475
	Eq. var. not assumed			-0.132	18.352	0.896	-0.0375	0.2834	-0.6320	0.5570
NEP SS2008	Eq. var. assumed	3.368	0.073	-0.194	44	0.847	-0.1773	0.9132	-2.0177	1.6631
	Eq. var. not assumed			-0.095	1.02	0.939	-0.1773	1.8590	-22.749	22.3944
NEP SS2009	Eq. var. assumed	0.002	0.962	1.186	54	0.241	0.5714	0.4819	-0.3948	1.5376
	Eq. var. not assumed			1.287	8.281	0.233	0.5714	0.4441	-0.4466	1.5895

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS2007	NEP Average NEP	1	22	3.350	1.0541	0.2247
	NEP Average NEP	0	2	4.500	0.7071	0.5000
NEP SS2008	NEP Average NEP	1	24	2.583	1.2761	0.2605
	NEP Average NEP	0	3	4.333	0.5774	0.3333
NEP SS2009	NEP Average NEP	1	22	2.532	1.2029	0.2565
	NEP Average NEP	0	2	5.000	0.0000	0.0000

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
NEP SS2007	Eq. var. assumed	0.770	0.390	-1.496	22	0.149	-1.1500	0.7687	-2.7443	0.4443
	Eq. var. not assumed			-2.098	1.442	0.218	-1.1500	0.5482	-4.6403	2.3403
NEP SS2008	Eq. var. assumed	2.334	0.139	-2.314	25	0.029	-1.7500	0.7562	-3.3074	-0.1926
	Eq. var. not assumed			-4.137	5.025	0.009	-1.7500	0.4230	-2.8358	-0.6642
NEP SS2009	Eq. var. assumed	3.396	0.079	-2.844	22	0.009	-2.4682	0.8680	-4.2683	-0.6681
	Eq. var. not assumed			-9.624	21	0.000	-2.4682	0.2565	-3.0015	-1.9348

T-Test between Winter Semester and Summer Semester within a Batch for Control Data

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP	WS0607	1	71	3.501	1.3818	0.1640
	SS2008	1	41	2.593	0.8748	0.1366
WEP	WS0708	1	27	3.052	1.0184	0.1960
	SS2008	1	26	2.162	1.1486	0.2253
WEP	WS0809	1	29	2.638	1.4115	0.2621
	SS2009	1	34	2.824	1.1855	0.2033

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP	Eq. var. assumed	17.246	0.000	3.791	110	0.000	0.9087	0.2397	0.4337	1.3837
	Eq. var. not assumed			4.258	109	0.000	0.9087	0.2134	0.4857	1.3317
WEP	Eq. var. assumed	0.650	0.424	2.989	51	0.004	0.8903	0.2979	0.2922	1.4884
	Eq. var. not assumed			2.982	49.76	0.004	0.8903	0.2986	0.2905	1.4901
WEP	Eq. var. assumed	2.538	0.116	-0.567	61	0.573	-0.1856	0.3271	-0.8397	0.4685
	Eq. var. not assumed			-0.559	54.951	0.578	-0.1856	0.3317	-0.8504	0.4792

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP	WS0607	1	25	3.932	1.3247	0.2649
	SS2008	1	19	3.158	0.8821	0.2024
WEP	WS0708	1	20	2.260	1.0748	0.2403
	SS2008	1	20	1.915	1.2717	0.2844
WEP	WS0809	1	15	1.973	1.3019	0.3362
	SS2009	1	17	1.724	1.0533	0.2555

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP	Eq. var. assumed	4.042	0.051	2.200	42	0.033	0.7741	0.3518	0.0641	1.4841
	Eq. var. not assumed			2.322	41.39	0.025	0.7741	0.3334	0.1010	1.4472
WEP	Eq. var. assumed	1.084	0.304	0.927	38	0.360	0.3450	0.3723	-0.4087	1.0987
	Eq. var. not assumed			0.927	36.973	0.360	0.3450	0.3723	-0.4094	1.0994
WEP	Eq. var. assumed	1.234	0.275	0.600	30	0.553	0.2498	0.4165	-0.6009	1.1005
	Eq. var. not assumed			0.592	26.969	0.559	0.2498	0.4222	-0.6165	1.1161

T-Test between Winter Semester and Summer Semester within a Batch for Participants (1) in Pop Quizzes or Group Work using NEP Average Grade

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP	WS0607	1	71	3.501	1.3818	0.1640
	SS2008	1	41	3.029	0.8878	0.1386
WEP	WS0708	1	27	3.248	1.0024	0.1929
	SS2008	1	26	2.742	1.1673	0.2289
WEP	WS0809	1	29	3.069	1.3696	0.2543
	SS2009	1	34	3.659	1.0252	0.1758

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP	Eq. var. assumed	16.655	0.000	1.964	110	0.052	0.4721	0.2404	-0.0042	0.9485
	Eq. var. not assumed			2.199	108.67	0.030	0.4721	0.2147	0.0465	0.8978
WEP	Eq. var. assumed	0.840	0.364	1.695	51	0.096	0.5058	0.2985	-0.0934	1.1051
	Eq. var. not assumed			1.690	49.239	0.097	0.5058	0.2994	-0.0957	1.1074
WEP	Eq. var. assumed	3.980	0.051	-1.95	61	0.056	-0.5899	0.3022	-1.1942	0.0145
	Eq. var. not assumed			-1.91	51.229	0.062	-0.5899	0.3092	-1.2105	0.0308

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP	WS0607	1	25	3.932	1.3247	0.2649
	SS2008	1	19	3.653	0.9845	0.2259
WEP	WS0708	1	20	2.535	0.9366	0.2094
	SS2008	1	20	2.400	1.2703	0.2840
WEP	WS0809	1	15	2.320	1.4047	0.3627
	SS2009	1	17	2.394	1.1448	0.2777

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP	Eq. var. assumed	2.503	0.121	0.771	42	0.445	0.2794	0.3624	-0.4521	1.0108
	Eq. var. not assumed			0.802	41.99	0.427	0.2794	0.3481	-0.4232	0.9820
WEP	Eq. var. assumed	2.554	0.118	0.383	38	0.704	0.1350	0.3529	-0.5794	
	Eq. var. not assumed			0.383	34.944	0.704	0.1350	0.3529	-0.5815	
WEP	Eq. var. assumed	0.815	0.374	-0.164	30	0.871	-0.0741	0.4509	-0.9949	0.8466
	Eq. var. not assumed			-0.162	27.08	0.872	-0.0741	0.4568	-1.0112	0.8630

T-Test between Winter Semester and Summer Semester within a Batch for Non-Participants (0) in Pop Quizzes or Group Work using NEP Average Grade

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP	WS0607	0	71	3.501	1.3818	0.1640
	SS2008	0	9	4.144	0.6966	0.2322
WEP	WS0708	0	25	3.784	0.8601	0.1720
	SS2008	0	20	4.070	0.9251	0.2069
WEP	WS0809	0	18	3.506	1.2027	0.2835
	SS2009	0	22	2.891	1.3140	0.2801

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP	Eq. var. assumed	8.937	0.004	-1.369	78	0.175	-0.6430	0.4698	-1.5784	0.2923
	Eq. var. not assumed			-2.262	17.473	0.037	-0.6430	0.2843	-1.2416	-0.0445
WEP	Eq. var. assumed	0.284	0.597	-1.072	43	0.290	-0.2860	0.2668	-0.8241	0.2521
	Eq. var. not assumed			-1.063	39.433	0.294	-0.2860	0.2690	-0.8300	0.2580
WEP	Eq. var. assumed	0.417	0.522	1.528	38	0.135	0.6146	0.4022	-0.1995	1.4288
	Eq. var. not assumed			1.542	37.48	0.131	0.6146	0.3985	-0.1925	1.4218

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP	WS0607	0	25	3.932	1.3247	0.2649
	SS2008	0	5	2.660	1.1082	0.4956
WEP	WS0708	0	14	4.193	0.7205	0.1926
	SS2008	0	7	3.857	0.8772	0.3316
WEP	WS0809	0	14	4.114	0.9281	0.2480
	SS2009	0	7	3.571	1.5119	0.5714

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP	Eq. var. assumed	0.878	0.357	2.004	28	0.055	1.2720	0.6349	-0.0285	2.5725
	Eq. var. not assumed			2.264	6.524	0.061	1.2720	0.5620	-0.0767	2.6207
WEP	Eq. var. assumed	0.365	0.553	0.938	19	0.360	0.3357	0.358	-0.4137	1.0851
	Eq. var. not assumed			0.876	10.195	0.401	0.3357	0.3834	-0.5164	1.1878
WEP	Eq. var. assumed	3.025	0.098	1.024	19	0.319	0.5429	0.5301	-0.5666	1.6523
	Eq. var. not assumed			0.871	8.337	0.408	0.5429	0.6229	-0.8836	1.9693

T-Test between Winter Semester and Summer Semester within a Batch for participants (1) of SWT during Summer Semester

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP	WS0607	.	8	2.525	1.3285	0.4697
	SS2008	1	8	2.500	0.6698	0.2368
WEP	WS0708	.	1	3.300	--	--
	SS2008	1	1	1.000	--	--
WEP	WS0809	.	5	3.000	1.5811	0.7071
	SS2009	1	5	3.200	1.1576	0.5177

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference Lower	Upper
WEP	Eq. var. assumed	2.193	0.161	0.048	14	0.963	0.0250	0.5260	-1.1032	1.1532
	Eq. var. not assumed			0.048	10.342	0.963	0.0250	0.5260	-1.1418	1.1918
WEP	Eq. var. assumed	.	.	.	0	.	2.3000	.	.	.
	Eq. var. not assumed			.	.	.	2.3000	.	.	.
WEP	Eq. var. assumed	0.399	0.545	-0.228	8	0.825	-0.2000	0.8764	-2.2209	1.8209
	Eq. var. not assumed			-0.228	7.331	0.826	-0.2000	0.8764	-2.2534	1.8534

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP	WS0607	.	16	3.662	1.3446	0.3361
	SS2008	1	16	2.894	1.0070	0.2517
WEP	WS0708	.	21	2.405	1.1698	0.2553
	SS2008	1	21	1.986	1.3275	0.2897
WEP	WS0809	.	20	2.345	1.3189	0.2949
	SS2009	1	20	1.815	1.0604	0.2371

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference Lower	Upper
WEP	Eq. var. assumed	0.738	0.397	1.831	30	0.077	0.7687	0.4200	-0.0889	1.6264
	Eq. var. not assumed			1.831	27.8	0.078	0.7687	0.4200	-0.0918	1.6293
WEP	Eq. var. assumed	0.296	0.589	1.085	40	0.284	0.4190	0.3861	-0.3613	1.1994
	Eq. var. not assumed			1.085	39.377	0.284	0.4190	0.3861	-0.3617	1.1998
WEP	Eq. var. assumed	2.257	0.141	1.401	38	0.169	0.5300	0.3784	-0.2361	1.2961
	Eq. var. not assumed			1.401	36.325	0.170	0.5300	0.3784	-0.2372	1.2972

T-Test between GWB Participants Who Worked in a Phase and Those Who Worked in More Phases

Group Statistics

Category (0=Single Phase, 1=More than 1 Phase)			N	Mean	Std. Deviation	Std. Error Mean
CS	NEP Average Grade	0	24	3.708	1.0434	0.2130
	NEP Average Grade	1	10	3.540	1.0244	0.3239
M	NEP Average Grade	0	12	2.583	1.2074	0.3486
	NEP Average Grade	1	5	1.940	0.9317	0.4167

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
CS	Eq. var. assumed	0.054	0.817	0.431	32	0.669	0.1683	0.3907	-0.6275	0.9642
	Eq. var. not assumed			0.434	17.205	0.670	0.1683	0.3877	-0.6489	0.9855
M	Eq. var. assumed	0.280	0.605	1.060	15	0.306	0.6433	0.6070	-0.6505	1.9372
	Eq. var. not assumed			1.184	9.81	0.264	0.6433	0.5432	-0.5702	1.8569

T-Test between "No Extra Methods & No SWT" (00) participation and "With Extra Methods & No SWT" participation (10) using WEP Average Grade

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP SS07	No GWA & No SWT	00	9	4.144	0.6966	0.2322
	GWA & No SWT	10	31	2.558	0.9244	0.1660
WEP SS08	No PQ & No SWT	00	19	4.021	0.9235	0.2119
	PQ & No SWT	10	25	2.208	1.1471	0.2294
WEP SS09	No GWB & No SWT	00	22	2.891	1.3140	0.2801
	GWB & No SWT	10	27	2.730	1.1269	0.2169

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP SS07	Eq. var. assumed	2.064	0.159	4.754	38	0.000	1.5864	0.3337	0.9108	2.2620
	Eq. var. not assumed			5.557	17.08	0.000	1.5864	0.2855	0.9843	2.1884
WEP SS08	Eq. var. assumed	1.565	0.218	5.635	42	0.000	1.8131	0.3217	1.1638	2.4623
	Eq. var. not assumed			5.806	41.828	0.000	1.8131	0.3123	1.1828	2.4433
WEP SS09	Eq. var. assumed	1.206	0.278	0.463	47	0.646	0.1613	0.3487	-0.5402	0.8628
	Eq. var. not assumed			0.455	41.635	0.651	0.1613	0.3543	-0.5539	0.8764

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP SS07	No GWA & No SWT	00	0	.	.	.
	GWA & No SWT	10	2	4.000	0	0.0000
WEP SS08	No PQ & No SWT	00	1	4.000	.	.
	PQ & No SWT	10	2	4.000	0	0.0000
WEP SS09	No GWB & No SWT	00	1	5.000	.	.
	GWB & No SWT	10	1	5.000	.	.

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP SS07	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.
WEP SS08	Eq. var. assumed	.	.	.	1	.	0.0000	0.0000	0.0000	0.0000
	Eq. var. not assumed			.	.	.	0.0000	.	.	.
WEP SS09	Eq. var. assumed	.	.	.	0	.	0.0000	.	.	.
	Eq. var. not assumed			.	.	.	0.0000	.	.	.

T-Test between "No Extra Methods" (00) participation and "Extra Methods & No SWT" participation (10) using NEP Average Grade

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS07	No GWA & No SWT	00	9	4.144	0.6966	0.2322
	GWA & No SWT	10	31	2.974	0.9345	0.1678
NEP SS08	No PQ & No SWT	00	19	4.021	0.9235	0.2119
	PQ & No SWT	10	25	2.800	1.1529	0.2306
NEP SS09	No GWB & No SWT	00	22	2.891	1.3140	0.2801
	GWB & No SWT	10	27	3.607	1.0247	0.1972

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
NEP SS07	Eq. var. assumed	1.893	0.177	3.474	38	0.001	1.1703	0.3369	0.4883	1.8522
	Eq. var. not assumed			4.084	17.285	0.001	1.1703	0.2865	0.5665	1.7740
NEP SS08	Eq. var. assumed	1.490	0.229	3.782	42	0.000	1.2211	0.3228	0.5696	1.8725
	Eq. var. not assumed			3.899	41.854	0.000	1.2211	0.3131	0.5891	1.8531
NEP SS09	Eq. var. assumed	2.202	0.145	-2.145	47	0.037	-0.7165	0.3340	-1.3884	-0.0446
	Eq. var. not assumed			-2.091	39.194	0.043	-0.7165	0.3426	-1.4093	-0.0237

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS07	No GWA & No SWT	00	0	.	.	.
	GWA & No SWT	10	2	4.500	0.7071	0.5000
NEP SS08	No PQ & No SWT	00	1	4.000	.	.
	PQ & No SWT	10	2	4.500	0.7071	0.5000
NEP SS09	No GWB & No SWT	00	1	5.000	.	.
	GWB & No SWT	10	1	5.000	.	.

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
NEP SS07	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.
NEP SS08	Eq. var. assumed	.	.	-0.577	1	0.667	-0.5000	0.8660	-11.504	10.5039
	Eq. var. not assumed			.	.	.	-0.5000	.	.	.
NEP SS09	Eq. var. assumed	.	.	.	0	.	0.0000	.	.	.
	Eq. var. not assumed			.	.	.	0.0000	.	.	.

T-Test between "No Extra Methods & No SWT" (00) participation and "No Extra Methods & SWT" participation (01) using WEP Average Grade

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP SS07	No GWA & No SWT	00	9	4.144	0.6966	0.2322
	No GWA & SWT	01	0	.	.	.
WEP SS08	No PQ & No SWT	00	19	4.021	0.9235	0.2119
	No PQ & SWT	01	1	5.000	.	.
WEP SS09	No GWB & No SWT	00	22	2.891	1.3140	0.2801
	No GWB & SWT	01	0	.	.	.

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference Lower	Upper
WEP SS07	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.
WEP SS08	Eq. var. assumed	.	.	-1.03	18	0.315	-0.9789	0.9475	-2.9696	1.0117
	Eq. var. not assumed			.	.	.	-0.9789	.	.	.
WEP SS09	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP SS07	No GWA & No SWT	00	0	.	.	.
	No GWA & SWT	01	5	2.660	1.1082	0.4956
WEP SS08	No PQ & No SWT	00	1	4.000	.	.
	No PQ & SWT	01	6	3.833	0.9585	0.3913
WEP SS09	No GWB & No SWT	00	1	5.000	.	.
	No GWB & SWT	01	6	3.333	1.5055	0.6146

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference Lower	Upper
WEP SS07	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.
WEP SS08	Eq. var. assumed	.	.	0.161	5	0.878	0.1667	1.0353	-2.4946	2.8279
	Eq. var. not assumed			.	.	.	0.1667	.	.	.
WEP SS09	Eq. var. assumed	.	.	1.025	5	0.352	1.6667	1.6262	-2.5135	5.8469
	Eq. var. not assumed			.	.	.	1.6667	.	.	.

T-Test between "No Extra Methods & No SWT" (00) participation and "No Extra Methods & SWT" participation (01) using NEP Average Grade

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS07	No GWA & No SWT	00	9	4.144	0.6966	0.2322
	No GWA & SWT	01	0	.	.	.
NEP SS08	No PQ & No SWT	00	19	4.021	0.9235	0.2119
	No PQ & SWT	01	1	5.000	.	.
NEP SS09	No GWB & No SWT	00	22	2.891	1.3140	0.2801
	No GWB & SWT	01	0	.	.	.

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference Lower	Upper
NEP SS07	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.
NEP SS08	Eq. var. assumed	.	.	-1.033	18	0.315	-0.9789	0.9475	-2.9696	1.0117
	Eq. var. not assumed			.	.	.	-0.9789	.	.	.
NEP SS09	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS07	No GWA & No SWT	00	0	.	.	.
	No GWA & SWT	01	5	2.660	1.1082	0.4956
NEP SS08	No PQ & No SWT	00	1	4.000	.	.
	No PQ & SWT	01	6	3.833	0.9585	0.3913
NEP SS09	No GWB & No SWT	00	1	5.000	.	.
	No GWB & SWT	01	6	3.333	1.5055	0.6146

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference Lower	Upper
NEP SS07	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.
NEP SS08	Eq. var. assumed	.	.	0.161	5	0.878	0.1667	1.0353	-2.4946	2.8279
	Eq. var. not assumed			.	.	.	0.1667	.	.	.
NEP SS09	Eq. var. assumed	.	.	1.025	5	0.352	1.6667	1.6262	-2.5135	5.8469
	Eq. var. not assumed			.	.	.	1.6667	.	.	.

T-Test between "No Extra Methods & No SWT" (00) participation and "Extra Methods & SWT" participation (11) using WEP Average Grade

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP SS07	No GWA & No SWT	00	9	4.144	0.6966	0.2322
	GWA & SWT	11	10	2.700	0.7318	0.2314
WEP SS08	No PQ & No SWT	00	19	4.021	0.9235	0.2119
	PQ & SWT	11	1	1.000	.	.
WEP SS09	No GWB & No SWT	00	22	2.891	1.3140	0.2801
	GWB & SWT	11	7	3.186	1.4265	0.5391

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP SS07	Eq. var. assumed	0.103	0.752	4.394	17	0.000	1.4444	0.3287	0.7509	2.1380
	Eq. var. not assumed			4.406	16.934	0.000	1.4444	0.3278	0.7526	2.1363
WEP SS08	Eq. var. assumed	.	.	3.188	18	0.005	3.0211	0.9475	1.0304	5.0117
	Eq. var. not assumed			.	.	.	3.0211	.	.	.
WEP SS09	Eq. var. assumed	0.096	0.759	-0.507	27	0.616	-0.2948	0.5814	-1.4877	0.8981
	Eq. var. not assumed			-0.485	9.48	0.639	-0.2948	0.6076	-1.6587	1.0691

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP SS07	No GWA & No SWT	00	0	.	.	.
	GWA & SWT	11	17	3.059	0.8811	0.2137
WEP SS08	No PQ & No SWT	00	1	4.000	.	.
	PQ & SWT	11	18	1.683	1.1132	0.2624
WEP SS09	No GWB & No SWT	00	1	5.000	.	.
	GWB & SWT	11	16	1.519	0.6504	0.1626

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP SS07	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.
WEP SS08	Eq. var. assumed	.	.	2.026	17	0.059	2.3167	1.1437	-0.0962	4.7296
	Eq. var. not assumed			.	.	.	2.3167	.	.	.
WEP SS09	Eq. var. assumed	.	.	5.193	15	0.000	3.4813	0.6704	2.0524	4.9101
	Eq. var. not assumed			.	.	.	3.4813	.	.	.

T-Test between “No Extra Methods & No SWT” (00) participation and “Extra Methods & SWT” participation (11) using NEP Average Grade

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS07	No GWA & No SWT	00	9	4.144	0.6966	0.2322
	GWA & SWT	11	10	3.200	0.7409	0.2343
NEP SS08	No PQ & No SWT	00	19	4.021	0.9235	0.2119
	PQ & SWT	11	1	1.300	.	.
NEP SS09	No GWB & No SWT	00	22	2.891	1.3140	0.2801
	GWB & SWT	11	7	3.857	1.0830	0.4093

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
NEP SS07	Eq. var. assumed	0.097	0.76	2.853	17	0.011	0.9444	0.3310	0.2461	1.6428
	Eq. var. not assumed			2.863	16.958	0.011	0.9444	0.3299	0.2484	1.6405
NEP SS08	Eq. var. assumed	.	.	2.872	18	0.010	2.7211	0.9475	0.7304	4.7117
	Eq. var. not assumed			.	.	.	2.7211	.	.	.
NEP SS09	Eq. var. assumed	0.177	0.677	-1.758	27	0.090	-0.9662	0.5495	-2.0937	0.1613
	Eq. var. not assumed			-1.948	12.174	0.075	-0.9662	0.4960	-2.0452	0.1128

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS07	No GWA & No SWT	00	0	.	.	.
	GWA & SWT	11	17	3.553	0.9792	0.2375
NEP SS08	No PQ & No SWT	00	1	4.000	.	.
	PQ & SWT	11	18	2.167	1.0944	0.2579
NEP SS09	No GWB & No SWT	00	1	5.000	.	.
	GWB & SWT	11	16	2.231	0.9576	0.2394

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
NEP SS07	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.
NEP SS08	Eq. var. assumed	.	.	1.631	17	0.121	1.8333	1.1244	-0.5389	4.2055
	Eq. var. not assumed			.	.	.	1.8333	.	.	.
NEP SS09	Eq. var. assumed	.	.	2.805	15	0.013	2.7688	0.9871	0.6649	4.8726
	Eq. var. not assumed			.	.	.	2.7688	.	.	.

T-Test between "Extra Methods & No SWT" (10) participation and "No Extra Methods & SWT" participation (01) using WEP Average Grade

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP SS07	GWA & No SWT	10	31	2.558	0.9244	0.1660
	No GWA & SWT	01	0	.	.	.
WEP SS08	PQ & No SWT	10	25	2.208	1.1471	0.2294
	No PQ & SWT	01	1	5.000	.	.
WEP SS09	GWB & No SWT	10	27	2.730	1.1269	0.2169
	No GWB & SWT	01	0	.	.	.

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP SS07	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.
WEP SS08	Eq. var. assumed	.	.	-2.387	24	0.025	-2.7920	1.1698	-5.2063	-0.3777
	Eq. var. not assumed			.	.	.	-2.7920	.	.	.
WEP SS09	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP SS07	GWA & No SWT	10	2	4.000	0.0000	0.0000
	No GWA & SWT	01	5	2.660	1.1082	0.4956
WEP SS08	PQ & No SWT	10	2	4.000	0.0000	0.0000
	No PQ & SWT	01	6	3.833	0.9585	0.3913
WEP SS09	GWB & No SWT	10	1	5.000	.	.
	No GWB & SWT	01	6	3.333	1.5055	0.6146

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP SS07	Eq. var. assumed	2.830	0.153	1.616	5	0.167	1.3400	0.8293	-0.7917	3.4717
	Eq. var. not assumed			2.704	4	0.054	1.3400	0.4956	-0.0360	2.7160
WEP SS08	Eq. var. assumed	5.650	0.055	0.233	6	0.823	0.1667	0.7144	-1.5814	1.9147
	Eq. var. not assumed			0.426	5	0.688	0.1667	0.3913	-0.8392	1.1725
WEP SS09	Eq. var. assumed	.	.	1.025	5	0.352	1.6667	1.6262	-2.5135	5.8469
	Eq. var. not assumed			.	.	.	1.6667	.	.	.

T-Test between "Extra Methods & No SWT" (10) participation and "No Extra Methods & SWT" participation (01) using NEP Average Grade

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS07	GWA & No SWT	10	31	2.974	0.9345	0.1678
	No GWA & SWT	01	0	.	.	.
NEP SS08	PQ & No SWT	10	25	2.800	1.1529	0.2306
	No PQ & SWT	01	1	5.000	.	.
NEP SS09	GWB & No SWT	10	27	3.607	1.0247	0.1972
	No GWB & SWT	01	0	.	.	.

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
NEP SS07	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.
NEP SS08	Eq. var. assumed	.	.	-1.871	24	0.074	-2.2000	1.1757	-4.6266	0.2266
	Eq. var. not assumed			.	.	.	-2.2000	.	.	.
NEP SS09	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS07	GWA & No SWT	10	2	4.500	0.7071	0.5000
	No GWA & SWT	01	5	2.660	1.1082	0.4956
NEP SS08	PQ & No SWT	10	2	4.500	0.7071	0.5000
	No PQ & SWT	01	6	3.833	0.9585	0.3913
NEP SS09	GWB & No SWT	10	1	5.000	.	.
	No GWB & SWT	01	6	3.333	1.5055	0.6146

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
NEP SS07	Eq. var. assumed	0.411	0.550	2.114	5	0.088	1.8400	0.8704	-0.3976	4.0776
	Eq. var. not assumed			2.614	3.166	0.075	1.8400	0.7040	-0.3354	4.0154
NEP SS08	Eq. var. assumed	0.721	0.429	0.886	6	0.410	0.6667	0.7523	-1.1741	2.5074
	Eq. var. not assumed			1.050	2.419	0.387	0.6667	0.6349	-1.6586	2.9919
NEP SS09	Eq. var. assumed	.	.	1.025	5	0.352	1.6667	1.6262	-2.5135	5.8469
	Eq. var. not assumed			.	.	.	1.6667	.	.	.

T-Test between "Extra Methods & No SWT" (10) participation and "Extra Methods & SWT" participation (11) using WEP Average Grade

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP SS07	GWA & No SWT	10	31	2.558	0.9244	0.1660
	GWA & SWT	11	10	2.700	0.7318	0.2314
WEP SS08	PQ & No SWT	10	25	2.208	1.1471	0.2294
	PQ & SWT	11	1	1.000	.	.
WEP SS09	GWB & No SWT	10	27	2.730	1.1269	0.2169
	GWB & SWT	11	7	3.186	1.4265	0.5391

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP SS07	Eq. var. assumed	1.406	0.243	-0.442	39	0.661	-0.1419	0.3214	-0.7920	0.5081
	Eq. var. not assumed			-0.498	19.129	0.624	-0.1419	0.2848	-0.7378	0.4539
WEP SS08	Eq. var. assumed	.	.	1.033	24	0.312	1.2080	1.1698	-1.2063	3.6223
	Eq. var. not assumed			.	.	.	1.2080	.	.	.
WEP SS09	Eq. var. assumed	1.149	0.292	-0.905	32	0.372	-0.4561	0.5042	-1.4832	0.5710
	Eq. var. not assumed			-0.785	8.05	0.455	-0.4561	0.5811	-1.7947	0.8826

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP SS07	GWA & No SWT	10	2	4.000	0.0000	0.0000
	GWA & SWT	11	17	3.059	0.8811	0.2137
WEP SS08	PQ & No SWT	10	2	4.000	0.0000	0.0000
	PQ & SWT	11	18	1.683	1.1132	0.2624
WEP SS09	GWB & No SWT	10	1	5.000	.	.
	GWB & SWT	11	16	1.519	0.6504	0.1626

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP SS07	Eq. var. assumed	6.860	0.018	1.473	17	0.159	0.9412	0.6390	-0.4070	2.2893
	Eq. var. not assumed			4.404	16	0.000	0.9412	0.2137	0.4882	1.3942
WEP SS08	Eq. var. assumed	2.656	0.121	2.873	18	0.010	2.3167	0.8063	0.6226	4.0107
	Eq. var. not assumed			8.830	17	0.000	2.3167	0.2624	1.7631	2.8702
WEP SS09	Eq. var. assumed	.	.	5.193	15	0.000	3.4813	0.6704	2.0524	4.9101
	Eq. var. not assumed			.	.	.	3.4813	.	.	.

T-Test between "Extra Methods & No SWT" (10) participation and "Extra Methods & SWT" participation (11) using NEP Average Grade

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS07	GWA & No SWT	10	31	2.974	0.9345	0.1678
	GWA & SWT	11	10	3.200	0.7409	0.2343
NEP SS08	PQ & No SWT	10	25	2.800	1.1529	0.2306
	PQ & SWT	11	1	1.300	.	.
NEP SS09	GWB & No SWT	10	27	3.607	1.0247	0.1972
	GWB & SWT	11	7	3.857	1.0830	0.4093

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
NEP SS07	Eq. var. assumed	1.267	0.267	-0.695	39	0.491	-0.2258	0.3250	-0.8831	0.4315
	Eq. var. not assumed			-0.784	19.1	0.443	-0.2258	0.2882	-0.8288	0.3772
NEP SS08	Eq. var. assumed	.	.	1.276	24	0.214	1.5000	1.1757	-0.9266	3.9266
	Eq. var. not assumed			.	.	.	1.5000	.	.	.
NEP SS09	Eq. var. assumed	0.476	0.495	-0.568	32	0.574	-0.2497	0.4393	-1.1447	0.6452
	Eq. var. not assumed			-0.550	8.996	0.596	-0.2497	0.4544	-1.2776	0.7781

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS07	GWA & No SWT	10	2	4.500	0.7071	0.5000
	GWA & SWT	11	17	3.553	0.9792	0.2375
NEP SS08	PQ & No SWT	10	2	4.500	0.7071	0.5000
	PQ & SWT	11	18	2.167	1.0944	0.2579
NEP SS09	GWB & No SWT	10	1	5.000	.	.
	GWB & SWT	11	16	2.231	0.9576	0.2394

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
NEP SS07	Eq. var. assumed	0.867	0.365	1.312	17	0.207	0.9471	0.7216	-0.5755	2.4696
	Eq. var. not assumed			1.711	1.497	0.269	0.9471	0.5535	-2.3916	4.2857
NEP SS08	Eq. var. assumed	0.475	0.499	2.908	18	0.009	2.3333	0.8024	0.6476	4.0191
	Eq. var. not assumed			4.147	1.596	0.077	2.3333	0.5626	-0.7796	5.4463
NEP SS09	Eq. var. assumed	.	.	2.805	15	0.013	2.7688	0.9871	0.6649	4.8726
	Eq. var. not assumed			.	.	.	2.7688	.	.	.

T-Test between "No Extra Methods & SWT" (01) participation and "Extra Methods & SWT" participation (11) using WEP Average Grade

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP SS07	No GWA & SWT	01	0	.	.	.
	GWA & SWT	11	10	2.700	0.7318	0.2314
WEP SS08	No PQ & SWT	01	1	5.000	.	.
	PQ & SWT	11	1	1.000	.	.
WEP SS09	No GWB & SWT	01	0	.	.	.
	GWB & SWT	11	7	3.186	1.4265	0.5391

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP SS07	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.
WEP SS08	Eq. var. assumed	.	.	.	0	.	4.0000	.	.	.
	Eq. var. not assumed			.	.	.	4.0000	.	.	.
WEP SS09	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
WEP SS07	No GWA & SWT	01	5	2.660	1.1082	0.4956
	GWA & SWT	11	17	3.059	0.8811	0.2137
WEP SS08	No PQ & SWT	01	6	3.833	0.9585	0.3913
	PQ & SWT	11	18	1.683	1.1132	0.2624
WEP SS09	No GWB & SWT	01	6	3.333	1.5055	0.6146
	GWB & SWT	11	16	1.519	0.6504	0.1626

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
WEP SS07	Eq. var. assumed	0.040	0.843	-0.842	20	0.410	-0.3988	0.4736	-1.3868	0.5891
	Eq. var. not assumed			-0.739	5.578	0.490	-0.3988	0.5397	-1.7440	0.9464
WEP SS08	Eq. var. assumed	0.034	0.854	4.223	22	0.000	2.1500	0.5091	1.0942	3.2058
	Eq. var. not assumed			4.564	9.917	0.001	2.1500	0.4711	1.0991	3.2009
WEP SS09	Eq. var. assumed	5.088	0.035	4.032	20	0.001	1.8146	0.4501	0.8758	2.7534
	Eq. var. not assumed			2.854	5.715	0.031	1.8146	0.6358	0.2399	3.3893

T-Test between "No Extra Methods & SWT" (01) participation and "Extra Methods & SWT" participation (11) using NEP Average Grade

Discipline = Computer Science

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS07	No GWA & SWT	01	0	.	.	.
	GWA & SWT	11	10	3.200	0.7409	0.2343
NEP SS08	No PQ & SWT	01	1	5.000	.	.
	PQ & SWT	11	1	1.300	.	.
NEP SS09	No GWB & SWT	01	0	.	.	.
	GWB & SWT	11	7	3.857	1.0830	0.4093

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
NEP SS07	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.
NEP SS08	Eq. var. assumed	.	.	.	0	.	3.7000	.	.	.
	Eq. var. not assumed			.	.	.	3.7000	.	.	.
NEP SS09	Eq. var. assumed	.	.	.	.	.	.	.	.	.
	Eq. var. not assumed			.	.	.	.	.	.	.

Discipline = Mechatronics

Group Statistics

Category			N	Mean	Std. Deviation	Std. Error Mean
NEP SS07	No GWA & SWT	01	5	2.660	1.1082	0.4956
	GWA & SWT	11	17	3.553	0.9792	0.2375
NEP SS08	No PQ & SWT	01	6	3.833	0.9585	0.3913
	PQ & SWT	11	18	2.167	1.0944	0.2579
NEP SS09	No GWB & SWT	01	6	3.333	1.5055	0.6146
	GWB & SWT	11	16	2.231	0.9576	0.2394

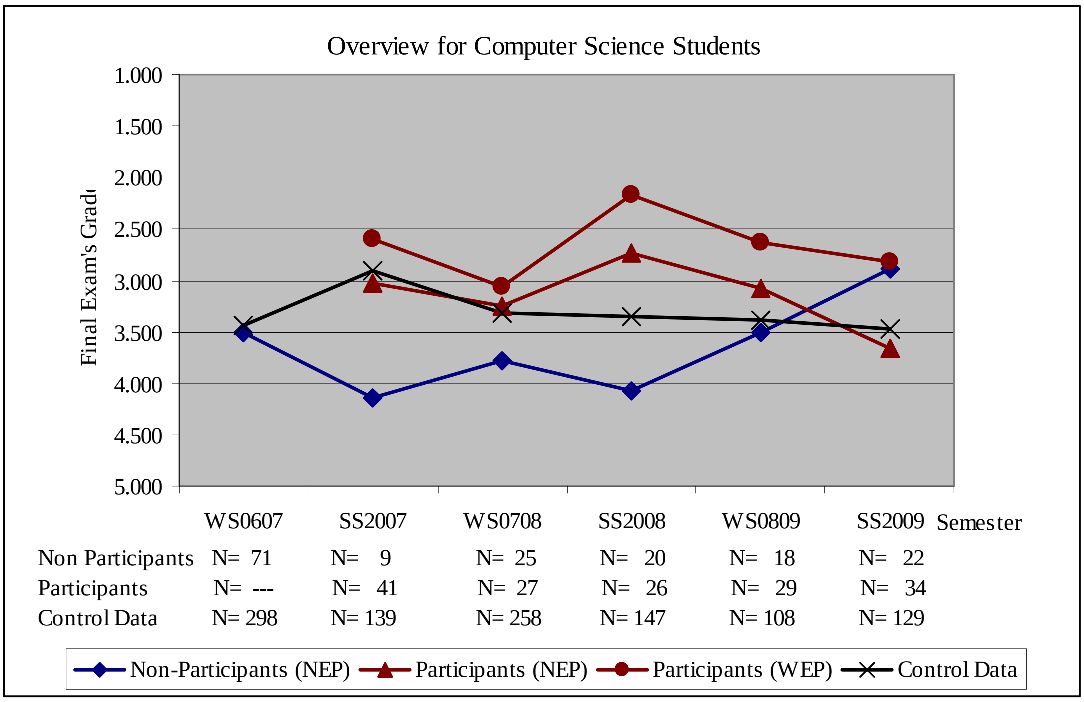
Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
NEP SS07	Eq. var. assumed	0.005	0.945	-1.744	20	0.096	-0.8929	0.5120	-1.9609	0.1750
	Eq. var. not assumed			-1.625	5.97	0.156	-0.8929	0.5496	-2.2393	0.4534
NEP SS08	Eq. var. assumed	0.040	0.844	3.320	22	0.003	1.6667	0.5020	0.6255	2.7079
	Eq. var. not assumed			3.556	9.748	0.005	1.6667	0.4687	0.6188	2.7146
NEP SS09	Eq. var. assumed	1.170	0.292	2.056	20	0.053	1.1021	0.5362	-0.0163	2.2205
	Eq. var. not assumed			1.671	6.582	0.141	1.1021	0.6596	-0.4780	2.6821

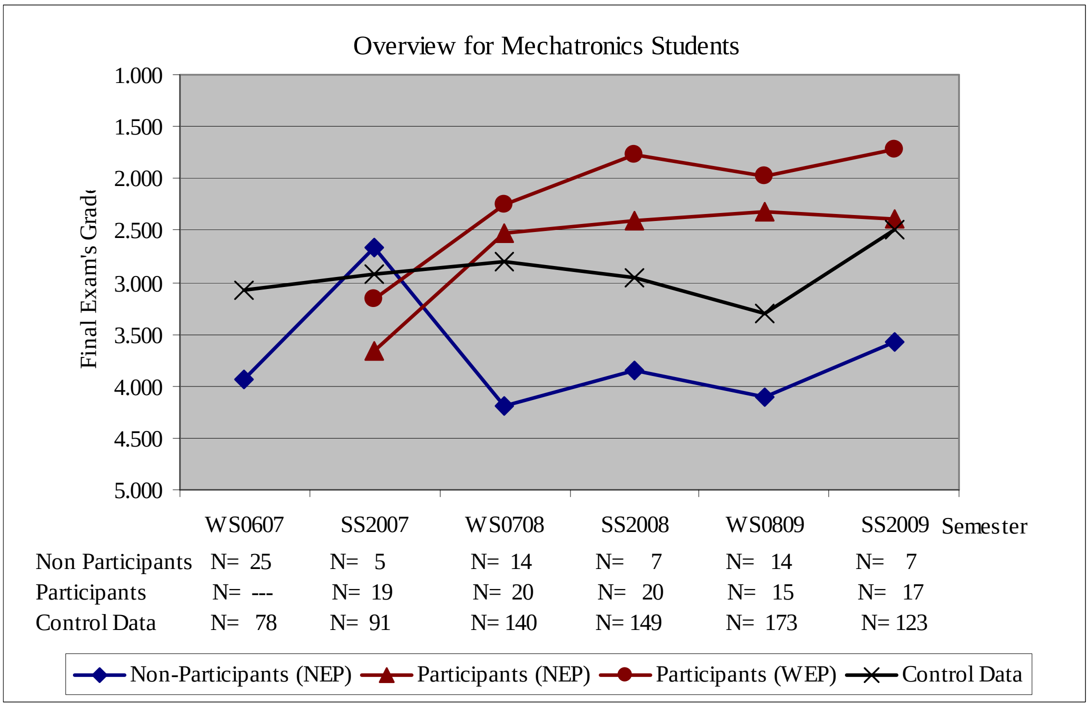
10.2 Graphs

10.2.1 Overview across 6 Semesters

Computer Science Students



Mechatronics Students



10.3 Statistical data – Evaluations' Result

10.3.1 T-Test

T-Test between Motivating Presentation to Comprehensibility of Lecture, and Usefulness of Demonstration

Group Statistics

Category	Sem.	N	Mean	Std. Deviation	Std. Error Mean
L-Motivating Presentation (L-MP)	SS2007	42	3.429	1.0393	0.1604
	SS2008	34	2.735	0.9312	0.1597
	WS0809	51	2.745	0.7961	0.1115
	SS2009	20	3.600	1.1877	0.2656
L-Comprehensibility of Lecture (L-CL)	SS2007	44	2.568	1.0432	0.1573
	SS2008	35	2.457	0.8168	0.1381
	WS0809	51	2.471	0.9456	0.1324
	SS2009	21	3.048	1.0235	0.2234
L-Usefulness of Demonstration (L-UD)	SS2007	34	2.618	1.1014	0.1889
	SS2008	26	2.846	1.0077	0.1976
	WS0809	40	2.500	0.9871	0.1561
	SS2009	16	3.000	1.1547	0.2887

Independent Samples Test

T- Test between SS2009 and SS2007		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
L-MP	Eq. var. assumed	0.650	0.423	0.580	60	0.564	0.1714	0.2957	-0.4201	0.7629
	Eq. var. not assumed			0.553	33.329	0.584	0.1714	0.3102	-0.4595	0.8024
L-CL	Eq. var. assumed	0.000	0.990	1.743	63	0.086	0.4794	0.2750	-0.0702	1.0291
	Eq. var. not assumed			1.755	40.157	0.087	0.4794	0.2732	-0.0726	1.0315
L-DU	Eq. var. assumed	0.420	0.520	1.128	48	0.265	0.3824	0.3390	-0.2993	1.0640
	Eq. var. not assumed			1.108	28.24	0.277	0.3824	0.3450	-0.3240	1.0887

Independent Samples Test

T- Test between SS2009 and SS2008		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
L-MP	Eq. var. assumed	2.650	0.110	2.972	52	0.004	0.8647	0.2909	0.2809	1.4485
	Eq. var. not assumed			2.790	32.761	0.009	0.8647	0.3099	0.2341	1.4954
L-CL	Eq. var. assumed	1.401	0.242	2.380	54	0.021	0.5905	0.2481	0.0930	1.0880
	Eq. var. not assumed			2.249	35.184	0.031	0.5905	0.2626	0.0575	1.1235
L-DU	Eq. var. assumed	0.002	0.965	0.455	40	0.652	0.1538	0.3385	-0.5302	0.8379
	Eq. var. not assumed			0.440	28.587	0.663	0.1538	0.3498	-0.5621	0.8698

Independent Samples Test

T- Test between SS2009 and WS0809		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
L-MP	Eq. var. assumed	7.388	0.008	3.520	69	0.001	0.8549	0.2429	0.3703	1.3395
	Eq. var. not assumed			2.968	25.979	0.006	0.8549	0.2880	0.2629	1.4469
L-CL	Eq. var. assumed	0.071	0.791	2.298	70	0.025	0.5770	0.2511	0.0762	1.0779
	Eq. var. not assumed			2.222	34.808	0.033	0.5770	0.2597	0.0498	1.1043
L-DU	Eq. var. assumed	0.159	0.691	1.631	54	0.109	0.5000	0.3066	-0.1146	1.1146
	Eq. var. not assumed			1.524	24.254	0.141	0.5000	0.3282	-0.1769	1.1769

T-Test between Computer Science and Mechatronics Students' Respond to Lecture's and Exercise's Structure and Comprehensibility

Semesters = SS2007

Group Statistics

Category	Discipline	N	Mean	Std. Deviation	Std. Error Mean
L-Structure (L-S)	CS	29	2.517	1.0896	0.2023
	M	15	2.533	0.7432	0.1919
L-Comprehensibility of Lecture (L-CL)	CS	29	2.621	1.0828	0.2011
	M	15	2.467	0.9904	0.2557
E-Structure (E-S)	CS	26	3.346	1.0175	0.1996
	M	15	3.267	0.7988	0.2063
E-Comprehensibility of Exercise (E-CE)	CS	26	3.423	0.9021	0.1769
	M	15	3.267	0.7988	0.2063

Independent Samples Test

		Levene's Test for Equality of Variances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
L-S	Eq. var. assumed	1.989	0.166	-0.051	42	0.959	-0.0161	0.3141	-0.6500	0.6179
	Eq. var. not assumed			-0.058	38.586	0.954	-0.0161	0.2789	-0.5803	0.5481
L-CL	Eq. var. assumed	0.212	0.648	0.460	42	0.648	0.1540	0.3349	-0.5218	0.8298
	Eq. var. not assumed			0.473	30.779	0.639	0.1540	0.3253	-0.5096	0.8177
E-S	Eq. var. assumed	2.171	0.149	0.259	39	0.797	0.0795	0.3064	-0.5402	0.6992
	Eq. var. not assumed			0.277	35.204	0.783	0.0795	0.287	-0.5030	0.6620
E-CE	Eq. var. assumed	0.159	0.693	0.557	39	0.581	0.1564	0.2809	-0.4118	0.7247
	Eq. var. not assumed			0.576	32.369	0.569	0.1564	0.2717	-0.3969	0.7097

Semesters = SS2008

Group Statistics

Category	Dis-cipline	N	Mean	Std. Devia-tion	Std. Error Mean
L-Structure (L-S)	CS	16	1.938	0.6801	0.1700
	M	19	2.632	1.0651	0.2444
L-Comprehensibility of Lecture (L-CL)	CS	16	2.125	0.5000	0.1250
	M	19	2.737	0.9335	0.2142
E-Structure (E-S)	CS	16	2.688	1.0145	0.2536
	M	17	2.471	0.6243	0.1514
E-Comprehensibility of Exercise (E-CE)	CS	16	3.375	1.5000	0.3750
	M	17	3.294	0.9196	0.2230

Independent Samples Test

		Levene's Test for Equality of Va-riances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Diffe-rence	Std. Error Diffe-rence	95% Confidence Interval of the Difference	
									Lower	Upper
L-S	Eq. var. assumed	4.646	0.039	-2.247	33	0.031	-0.6941	0.3089	-1.3226	-0.0655
	Eq. var. not assumed			-2.332	30.944	0.026	-0.6941	0.2977	-1.3013	-0.0869
L-CL	Eq. var. assumed	9.138	0.005	-2.350	33	0.025	-0.6118	0.2604	-1.1416	-0.0821
	Eq. var. not assumed			-2.467	28.399	0.020	-0.6118	0.2480	-1.1194	-0.1042
E-S	Eq. var. assumed	2.639	0.114	0.745	31	0.462	0.2169	0.2912	-0.3771	0.8109
	Eq. var. not assumed			0.734	24.66	0.470	0.2169	0.2954	-0.3919	0.8257
E-CE	Eq. var. assumed	5.998	0.020	0.188	31	0.852	0.0809	0.4302	-0.7964	0.9582
	Eq. var. not assumed			0.185	24.602	0.854	0.0809	0.4363	-0.8184	0.9802

Semesters = WS0809

Group Statistics

Category	Dis-cipline	N	Mean	Std. Devia-tion	Std. Error Mean
L-Structure (L-S)	CS	28	2.893	0.9940	0.1879
	M	20	2.800	0.9515	0.2128
L-Comprehensibility of Lecture (L-CL)	CS	31	2.355	0.8774	0.1576
	M	20	2.650	1.0400	0.2325
E-Structure (E-S)	CS	31	2.323	0.7911	0.1421
	M	19	2.526	0.9643	0.2212
E-Comprehensibility of Exercise (E-CE)	CS	31	2.613	0.8032	0.1443
	M	19	2.474	1.1723	0.2689

		Levene's Test for Equality of Va-riances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Diffe-rence	Std. Error Diffe-rence	95% Confidence Interval of the Difference	
									Lower	Upper
L-S	Eq. var. assumed	0.006	0.940	0.325	46	0.747	0.0929	0.2859	-0.4827	0.6684
	Eq. var. not assumed			0.327	42.147	0.745	0.0929	0.2838	-0.4799	0.6656
L-CL	Eq. var. assumed	0.805	0.374	-1.090	49	0.281	-0.2952	0.2707	-0.8391	0.2488
	Eq. var. not assumed			-1.051	35.691	0.300	-0.2952	0.2809	-0.8651	0.2747
E-S	Eq. var. assumed	0.840	0.364	-0.813	48	0.420	-0.2037	0.2506	-0.7076	0.3001
	Eq. var. not assumed			-0.775	32.586	0.444	-0.2037	0.2629	-0.7389	0.3314
E-CE	Eq. var. assumed	2.161	0.148	0.499	48	0.620	0.1392	0.2792	-0.4222	0.7007
	Eq. var. not assumed			0.456	28.436	0.652	0.1392	0.3052	-0.4855	0.7639

Semesters = SS2009

Group Statistics

Category	Dis-cipline	N	Mean	Std. Devia-tion	Std. Error Mean
L-Structure (L-S)	CS	11	2.545	0.8202	0.2473
	M	9	3.000	0.7071	0.2357
L-Comprehensibility of Lecture (L-CL)	CS	11	2.455	0.6876	0.2073
	M	9	3.667	1.0000	0.3333
E-Structure (E-S)	CS	11	2.091	0.7006	0.2113
	M	9	2.556	0.8819	0.2940
E-Comprehensibility of Exercise (E-CE)	CS	11	2.000	0.6325	0.1907
	M	9	2.556	0.8819	0.2940

Independent Samples Test

		Levene's Test for Equality of Va-riances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Diffe-rence	Std. Error Diffe-rence	95% Confidence Interval of the Difference	
									Lower	Upper
L-S	Eq. var. assumed	1.244	0.279	-1.310	18	0.207	-0.4545	0.3470	-1.1835	0.2744
	Eq. var. not assumed			-1.331	17.928	0.200	-0.4545	0.3416	-1.1725	0.2634
L-CL	Eq. var. assumed	1.620	0.219	-3.207	18	0.005	-1.2121	0.3779	-2.0062	-0.4181
	Eq. var. not assumed			-3.088	13.741	0.008	-1.2121	0.3925	-2.0555	-0.3687
E-S	Eq. var. assumed	1.133	0.301	-1.315	18	0.205	-0.4646	0.3535	-1.2072	0.2779
	Eq. var. not assumed			-1.284	15.161	0.219	-0.4646	0.3620	-1.2355	0.3062
E-CE	Eq. var. assumed	3.309	0.086	-1.640	18	0.118	-0.5556	0.3387	-1.2672	0.1561
	Eq. var. not assumed			-1.585	14.145	0.135	-0.5556	0.3504	-1.3064	0.1953

T-Test between Hours Spent Revising during GWB and Pop Quiz Implementation

Group Statistics

Hours		N	Mean	Std. Devia-tion	Std. Error Mean
Computer Science (CS)	GWB	27	2.440	1.0860	0.2090
	PQ	19	2.110	1.3700	0.3140
Mechatronics (M)	GWB	11	2.090	0.9440	0.2850
	PQ	11	1.180	0.4050	0.1220

Independent Samples Test

Hours		Levene's Test for Equality of Va-riances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Diffe-rence	Std. Error Diffe-rence	95% Confidence Interval of the Difference	
									Lower	Upper
CS	Eq. var. assumed	0.584	0.449	0.936	44	0.354	0.3390	0.3620	-0.3910	1.0700
	Eq. var. not assumed			0.899	32.974	0.375	0.3390	0.3770	-0.4290	1.1070
M	Eq. var. assumed	3.517	0.075	2.936	20	0.008	0.9090	0.3100	0.2630	1.5550
	Eq. var. not assumed			2.936	13.554	0.011	0.9090	0.3100	0.2430	1.5750

T-Test between Computer Science and Mechatronics Students on Lecture Exercise Ratio

Group Statistics

Hours	N	Mean	Std. Devia-tion	Std. Error Mean
Computer Science (CS)	25	3.240	1.0520	0.2100
Mechatronics (M)	11	3.180	0.7510	0.2260

Independent Samples Test

Hours		Levene's Test for Equality of Va-riances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Diffe-rence	Std. Error Diffe-rence	95% Confidence Interval of the Difference	
									Lower	Upper
Hours	Eq. var. assumed	0.963	0.333	0.165	34	0.870	0.0580	0.3520	-0.6570	0. 7740
	Eq. var. not assumed			0.188	26.501	0.852	0.0580	0.3090	-0.5760	0. 6930

T-Test between Computer Science and Mechatronics Students on Theoretical and Application Questions

Semesters = WS0809

Group Statistics

Hours		N	Mean	Std. Deviation	Std. Error Mean
Question 3	CS	47	4.787	1.8846	0.2749
	M	29	4.252	2.3721	0.4405
Question 4	CS	47	1.479	1.8765	0.2737
	M	29	2.069	1.8647	0.3463

Independent Samples Test

Hours		Levene's Test for Equality of Va-riances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Diffe-rence	Std. Error Diffe-rence	95% Confidence Interval of the Difference	
									Lower	Upper
Question 3	Eq. var. assumed	0.315	0.576	1.089	74	0.280	0.5355	0.4918	-0.4443	0.6179
	Eq. var. not assumed			1.031	49.487	0.307	0.5355	0.5192	-0.5077	0.5481
Question 4	Eq. var. assumed	0.003	0.957	-1.335	74	0.186	-0.5902	0.4421	-1.4711	0.8298
	Eq. var. not assumed			-1.337	59.728	0.186	-0.5902	0.4414	-1.4732	0.8177

Semesters = WS0809

Group Statistics

Hours		N	Mean	Std. Deviation	Std. Error Mean
Question 4	CS	56	7.087	2.5232	0.3372
	M	24	7.533	2.9333	0.5988
Question 5	CS	56	8.319	4.0774	0.5449
	M	24	9.967	4.6346	0.9460
Question 6	CS	56	5.844	2.8921	0.3865
	M	24	5.704	3.2730	0.6681

Independent Samples Test

Hours		Levene's Test for Equality of Va-riances (Eq. Var.)		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Diffe-rence	Std. Error Diffe-rence	95% Confidence Interval of the Difference	
									Lower	Upper
Question 4	Eq. var. assumed	0.543	0.464	-0.690	78	0.492	-0.4464	0.6467	-1.7339	0.8412
	Eq. var. not assumed			-0.650	38.29	0.520	-0.4464	0.6872	-1.8371	0.9444
Question 5	Eq. var. assumed	0.130	0.719	-1.590	78	0.116	-1.6479	1.0367	-3.7119	0.4161
	Eq. var. not assumed			-1.509	38.996	0.139	-1.6479	1.0917	-3.8562	0.5603
Question 6	Eq. var. assumed	0.096	0.757	0.190	78	0.850	0.1396	0.7342	-1.3222	1.6013
	Eq. var. not assumed			0.181	39.135	0.857	0.1396	0.7718	-1.4214	1.7006

10.3.2 Correlations

Correlations between Motivating Presentation to Comprehensibility of Lecture, and Usefulness of Demonstration

Discipline = Computer Science, Mechatronics, Others

Correlations

		L-Comprehensibility of Lecture	L-Usefulness of Demon-stration	L-Motivating Presentation
L-Motivating Presentation SS2007	Pearson Correlation	0.459	0.337	1
	Sig. (2-tailed)	0.002	0.055	
	N	42	33	42
L-Motivating Presentation SS2008	Pearson Correlation	0.275	0.286	1
	Sig. (2-tailed)	0.115	0.157	
	N	34	26	34
L-Motivating Presentation WS0809	Pearson Correlation	0.561	0.462	1
	Sig. (2-tailed)	0.000	0.003	
	N	51	40	51
L-Motivating Presentation SS2009	Pearson Correlation	0.523	0.542	1
	Sig. (2-tailed)	0.018	0.030	
	N	20	16	20

Discipline = Computer Science

Correlations

		L-Comprehensibility of Lecture	L-Usefulness of Demon-stration	L-Motivating Presentation
L-Motivating Presentation SS2007	Pearson Correlation	0.491	0.405	1
	Sig. (2-tailed)	0.009	0.069	
	N	27	21	27
L-Motivating Presentation SS2008	Pearson Correlation	-0.262	0.200	1
	Sig. (2-tailed)	0.346	0.579	
	N	15	10	15
L-Motivating Presentation WS0809	Pearson Correlation	0.631	0.272	1
	Sig. (2-tailed)	0.000	0.178	
	N	31	26	31
L-Motivating Presentation SS2009	Pearson Correlation	0.452	0.866	1
	Sig. (2-tailed)	0.189	0.012	
	N	10	7	10

Discipline = Mechatronics

Correlations

		L-Comprehensibility of Lecture	L-Usefulness of Demonstration	L-Motivating Presentation
L-Motivating Presentation SS2007	Pearson Correlation	0.352	0.330	1
	Sig. (2-tailed)	0.199	0.295	
	N	15	12	15
L-Motivating Presentation SS2008	Pearson Correlation	0.593	0.414	1
	Sig. (2-tailed)	0.007	0.111	
	N	19	16	19
L-Motivating Presentation WS0809	Pearson Correlation	0.490	0.788	1
	Sig. (2-tailed)	0.028	0.001	
	N	20	14	20
L-Motivating Presentation SS2009	Pearson Correlation	0.401	0.513	1
	Sig. (2-tailed)	0.284	0.193	
	N	9	8	9

Correlations between Students' Grade with Attendance

Correlations

		Attendance	ES1 Grade
ES1 Grade	Pearson Correlation	-0.434	1
	Sig. (2-tailed)	0.082	
	N	17	17

10.4 Evaluation Form

10.4.1 WS0809 Evaluation Form

Fragebogen zur Beurteilung der Lehre im Fachbereich
Elektrotechnik/Informatik
an der Universität Kassel
Wintersemester 2008/09

Liebe Kommilitonin, lieber Kommilitone,

dieser Fragebogen soll eine Basis für die Diskussion der Form, des Umfangs und der Qualität der Lehre zwischen Studierenden und Dozenten bilden. Ihr Dozent führt diese Erhebung freiwillig durch. Die Auswertung der Daten erfolgt *anonym* und die Fragebögen verbleiben bei Ihrem Dozenten.

Zutreffende „Note“ bitte ankreuzen.

0. Persönliche Angaben:

0.1 Geschlecht: ☐ männlich ☐ weiblich

0.2 Wie alt sind Sie bei Ihrem letzten Geburtstag geworden: __________Jahre

0.3 Fachrichtung: ______________________________ 0.4 Semester: ___________

0.5 Studienziel: ☐ Informatik Bachelor ☐ Mechatronik DI ☐ ETechnik DII
☐ anderes ________________________

0.6 Haben Sie an den folgende Lehrveranstaltung teilgenommen? Wenn Ja, wann?

0.61 Pflichtfächer für Informatik /ETechnik — Wenn Ja, wann? (Bitte nennen Sie jedes Semester, falls Sie die Lehrveranstaltung mehr als einmal besucht haben)

Betriebssysteme	☐ Ja	☐ Nein	☐ WS0809	☐ Anderes Sem.:____
Rechnernetze	☐ Ja	☐ Nein	☐ WS0809	☐ Anderes Sem.:____
Theoretische Informatik	☐ Ja	☐ Nein	☐ WS0809	☐ Anderes Sem.:____
Diskrete Strukturen II	☐ Ja	☐ Nein	☐ WS0809	☐ Anderes Sem.:____
Grundwissen der Elektronik	☐ Ja	☐ Nein	☐ WS0809	☐ Anderes Sem.:____

0.62 Pflichtfächer für Mechatronik — Wenn Ja, wann? (Bitte nennen Sie jedes Semester, falls Sie die Lehrveranstaltung mehr als einmal besucht haben)

Mathematik 3	☐ Ja	☐ Nein	☐ WS0809	☐ Anderes Sem.:____
Techn. Mechanik 2	☐ Ja	☐ Nein	☐ WS0809	☐ Anderes Sem.:____
Fertigungstechnik	☐ Ja	☐ Nein	☐ WS0809	☐ Anderes Sem.:____
Elektrische Messtechnik	☐ Ja	☐ Nein	☐ WS0809	☐ Anderes Sem.:____
Digitaltechnik	☐ Ja	☐ Nein	☐ WS0809	☐ Anderes Sem.:____
Werkstoffe der Elektrotechnik	☐ Ja	☐ Nein	☐ WS0809	☐ Anderes Sem.:____

0.7 Haben Sie Praxiserfahrung ___________ Wenn Ja, für wie lange (in **Monaten**)?

a) Studentische Hilfskraft ☐ Ja ☐ Nein	1 - 4	5 - 8	9 -12	13 - 16	> 17
b) Ausbildung ☐ Ja ☐ Nein	1 - 4	5 - 8	9 -12	13 - 16	> 17
c) Praktikum ☐ Ja ☐ Nein	1 - 4	5 - 8	9 -12	13 - 16	> 17
d) andere Berufliche Tätigkeit ☐ Ja ☐ Nein	1 - 4	5 - 8	9 -12	13 - 16	> 17

1. Motivation:

1.1 Entsprachen die Inhalte der Vorlesung Ihrem persönlichen fachlichen Interesse?	immer	1	2	3	4	5	nie
1.2 Wurde Ihr Interesse an speziellen Fragestellungen des Fachs gefördert?	sehr gut	1	2	3	4	5	nicht ausreichend
1.3 Wurde in den zugehörigen Übungen die Anwendungen der Lehrinhalte verdeutlicht?	sehr gut	1	2	3	4	5	nicht ausreichend
1.4 Wurde in dem zugehörigen Praktikum die Anwendungen der Lehrinhalte verdeutlicht?	sehr gut	1	2	3	4	5	nicht ausreichend

1.5 Anmerkungen: ____

____

2. Umfang und Schwierigkeitsgrad der Vorlesung

2.1 Der Stoff erschien mir persönlich eher	knapp	-2	-1	0	1	2	zu umfangreich
2.2 Der Schwierigkeitsgrad war für mich eher	knapp	-2	-1	0	1	2	zu umfangreich
2.3 War das für die Vorlesung notwendige Vorwissen aus anderen Veranstaltungen bekannt?	sehr gut	1	2	3	4	5	nicht ausreichend
2.4 Wie bewerten Sie die stoffliche Überlappung zu anderen Lehrveranstaltungen? (*)	zu klein	-2	-1	0	1	2	zu groß

(*) Bitte Beispiele angeben: ____

____

2.5 Anmerkungen: ____

____

3. Durchführung und Aufbau der Vorlesung

3.1 Gliederung	sehr gut	1	2	3	4	5	nicht aus-reichend
3.2 Verständlichkeit der Erklärungen	sehr gut	1	2	3	4	5	nicht aus-reichend
3.3 Fehlerhäufigkeit	fast nie	1	2	3	4	5	störend oft
3.4 Tafelbild	sehr gut	1	2	3	4	5	nicht aus-reichend
3.5 Beamernutzung							
Folienqualität	sehr gut	1	2	3	4	5	schlecht
Folieneinsatz	sehr nützlich	1	2	3	4	5	unwirksam
3.6 Versuche/Demonstrationen	zu selten	-2	-1	0	1	2	zu häufig
Einsatz:	sehr nützlich	1	2	3	4	5	unwirksam
3.7 Gibt es sonstige Medien, die in der Vorlesung genutzt wurden?	☐ Ja				☐ Nein		
Wenn Ja, was? ____________________							
Einsatz sonstiger Medien	sehr nützlich	1	2	3	4	5	unwirksam
3.8 Präsentation des Stoffes	lebendig	1	2	3	4	5	langatmig
	motivie-rend	1	2	3	4	5	nicht mo-tivierend
	zu lang-sam	-2	-1	0	1	2	zu schnell
3.9 Werden Zwischenfragen klar und verständlich beantwortet?	immer	1	2	3	4	5	nie
3.10 Haben Sie versucht nach der Vorlesung ein Gespräch mit einem Dozenten zu führen?	☐ Ja				☐ Nein		
Wenn Ja, wie ist die Gesprächsbereitschaft?	sehr gut	1	2	3	4	5	schlecht
3.11 Waren Sie schon einmal in der Sprechstunde?	☐ Ja				☐ Nein		
Wenn Ja, wie ist die Gesprächsbereitschaft?	sehr gut	1	2	3	4	5	schlecht
3.12 Wie gut ist die Abstimmung von Vorlesung und Übung?	sehr gut	1	2	3	4	5	nicht aus-reichend

3.13 Anmerkungen: ____________________

4. Durchführung und Aufbau der Übung

4.1 Strukturierung	sehr gut	1	2	3	4	5	nicht ausreichend
4.2 Verständlichkeit von Erklärungen	sehr gut	1	2	3	4	5	nicht ausreichend
4.3 Qualität der Ubungsaufgaben	sehr gut	1	2	3	4	5	nicht ausreichend
4.4 Fehlerhäufigkeit	fast nie	1	2	3	4	5	störend oft
4.5 Geforderte Mitarbeit	zu wenig	-2	-1	0	1	2	zu viel
4.6 Behandlung des Stoffes	lebendig	1	2	3	4	5	langatmig
	motivierend	1	2	3	4	5	nicht motivierend
	zu langsam	-2	-1	0	1	2	zu schnell
4.7 Waren Sie schon einmal in der Sprechstunde?	☐ Ja				☐ Nein		
Wenn Ja, wie ist die Gesprächsbereitschaft?	sehr gut	1	2	3	4	5	schlecht

4.8 Anmerkungen:____

____

5. Praktikum – nicht Vorhanden für ES1

6. Äußere Beeinträchtigungen

6.1 Sitze/Tische	sehr gut	1	2	3	4	5	schlecht
6.2 Tafel	sehr gut	1	2	3	4	5	schlecht
6.3 Projektion	sehr gut	1	2	3	4	5	schlecht
6.4 Licht	zu klein	1	2	3	4	5	schlecht
6.5 Akustik	sehr gut	1	2	3	4	5	schlecht
6.6 Temperatur	sehr gut	1	2	3	4	5	schlecht
6.7 Luft	sehr gut	1	2	3	4	5	schlecht
6.8 Lärm	sehr niedrig	1	2	3	4	5	zu hoch

6.9 Anmerkungen:____

7. Eigener Einsatz

7.1 Wie oft haben Sie diese Veranstaltung in diesem Semester besucht? Bitte die konkrete Anzahl ankreuzen	1-3	4-6	7-9	10-12	13-15

7.2 Wie oft arbeiten Sie die Vorlesung nach?

generell	immer	1	2	3	4	5	nie
anhand des Skriptes	immer	1	2	3	4	5	nie
anhand der Mitschrift	immer	1	2	3	4	5	nie
anhand der Literatur	immer	1	2	3	4	5	nie

7.3 Wie viele Stunden pro Woche wenden Sie für diese Lehrveranstaltung auf? Std.

7.4 Wie viele Stunden pro Woche wenden Sie für das Studium auf? Std.

7.5 Wie viele Stunden pro Woche wenden Sie zur Finanzierung des Studiums auf? Std.

7.6 Anmerkungen: ______________________________

____

Form Nummer: 1

Form Nummer: _____1____________

Liebe Studentinnen und Studenten,

bitte behalten Sie diesen Teil der Evaluation für Ihre eigenen Unterlagen.

Für eine laufende Studie über die Zusammenhänge zwischen dem „Fragebogen zur Beurteilung der Lehre im Fachbereich“ der Studenten und deren Prüfungsnote sind wir auf Ihre Hilfe angewiesen.

Nachdem Sie das endgültige Ergebnis für Eingebettete Systeme I erhalten haben, wären wir Ihnen dankbar, wenn Sie eine E-Mail an Frau Sim (sim@uni-kassel.de) mit den folgenden Informationen.

1) Form Nummer (siehe oben)
2) Prüfungsnote für Eingebettete Systeme I

schicken würden. Nur mit Hilfe Ihrer Form Nummer und Prüfungsnote sind wir in der Lage, die notwendigen Zusammenhänge, welche für diese Studie benötigt werden, zu ziehen.

Bitte beachten Sie, dass diese Studie in **keiner Weise** Einfluss auf Ihre Prüfung hat.

Vielen Dank für Ihre Unterstützung.

Mit freundlichen Grüßen,

Fachgebiet Eingebettete Systeme

10.4.2 SS2009 Evaluation Form

Fragebogen zur Beurteilung der Lehre im Fachbereich
Elektrotechnik/Informatik an der Universität Kassel
Sommer Semester 2009

Liebe Kommilitonin, lieber Kommilitone,

dieser Fragebogen soll eine Basis für die Diskussion der Form, des Umfangs und der Qualität der Lehre zwischen Studierenden und Dozenten bilden. Ihr Dozent führt diese Erhebung freiwillig durch. Die Auswertung der Daten erfolgt *anonym* und die Fragebögen verbleiben bei Ihrem Dozenten.

Zutreffende „Note" bitte ankreuzen.

0. Persönliche Angaben:

0.1 Geschlecht: ☐ männlich ☐ weiblich

0.2 Wie alt sind Sie bei Ihrem letzten Geburtstag geworden: ________Jahre

0.3 Fachrichtung: ______________________________ 0.4 Semester: _________

0.5 Studienziel: ☐ Informatik Bachelor ☐ Mechatronik DI ☐ ETechnik DII

☐ anderes ______________________

1. Motivation:

1.1	Entsprachen die Inhalte der Vorlesung Ihrem persönlichen fachlichen Interesse?	immer	1	2	3	4	5	nie
1.2	Wurde Ihr Interesse an speziellen Fragestellungen des Fachs gefördert?	sehr gut	1	2	3	4	5	nicht ausreichend
1.3	Wurde in der zugehörigen Übung die Anwendungen der Lehrinhalte verdeutlicht?	sehr gut	1	2	3	4	5	nicht ausreichend

1.4 Anmerkungen: ____

2. Umfang und Schwierigkeitsgrad der Vorlesung

2.1	Der Stoff erschien mir persönlich eher	knapp	-2	-1	0	1	2	zu umfangreich
2.2	Der Schwierigkeitsgrad war für mich eher	knapp	-2	-1	0	1	2	zu umfangreich
2.3	War das für die Vorlesung notwendige Vorwissen aus anderen Lehrveranstaltungen bekannt?	sehr gut	1	2	3	4	5	nicht ausreichend
2.4	Wie bewerten Sie die stoffliche Überlappung zu anderen Lehrveranstaltungen? (*)	zu klein	-2	-1	0	1	2	zu groß

(*) Bitte Beispiele angeben: ____

2.5 Anmerkungen: ____

3. Durchführung und Aufbau der Vorlesung

3.1	Gliederung	sehr gut	1	2	3	4	5	nicht ausreichend
3.2	Verständlichkeit der Erklärungen	sehr gut	1	2	3	4	5	nicht ausreichend

3.4	Tafelbild	sehr gut	1	2	3	4	5	nicht ausreichend
3.5	Beamernutzung							
	Folienqualität	sehr gut	1	2	3	4	5	schlecht
	Folieneinsatz	sehr nützlich	1	2	3	4	5	unwirksam
3.6	Versuche/Demonstrationen	zu selten	-2	-1	0	1	2	zu häufig
	Einsatz:	sehr nützlich	1	2	3	4	5	unwirksam
3.7	Präsentation des Stoffes	lebendig	1	2	3	4	5	langatmig
		motivierend	1	2	3	4	5	nicht motivierend
		zu langsam	-2	-1	0	1	2	zu schnell
3.8	Werden Zwischenfragen klar und verständlich beantwortet?	immer	1	2	3	4	5	nie
3.9	Haben Sie versucht nach der Vorlesung ein Gespräch mit dem Dozenten zu führen?	☐ Ja				☐ Nein		
	Wenn Ja, wie ist die Gesprächsbereitschaft?	sehr gut	1	2	3	4	5	schlecht
3.10	Wie gut ist die							
	Zeitliche Abstimmung von Vorlesung und Übung?	sehr gut	1	2	3	4	5	nicht ausreichend
	Inhaltliche Abstimmung von Vorlesung und Übung?	sehr gut	1	2	3	4	5	nicht ausreichend

3.11 Anmerkungen: ____________________

4. Durchführung und Aufbau der Übung

4.1	Strukturierung	sehr gut	1	2	3	4	5	nicht ausreichend
4.2	Verständlichkeit von Erklärungen	sehr gut	1	2	3	4	5	nicht ausreichend
4.3	Qualität der Übungsaufgaben	sehr gut	1	2	3	4	5	nicht ausreichend
4.4	Fehlerhäufigkeit	fast nie	1	2	3	4	5	störend oft
4.5	Geforderte Mitarbeit	zu wenig	-2	-1	0	1	2	zu viel
4.6	Behandlung des Stoffes	lebendig	1	2	3	4	5	langatmig
		motivierend	1	2	3	4	5	nicht motivierend
		zu langsam	-2	-1	0	1	2	zu schnell
4.7	Waren Sie schon einmal in der Sprechstunde?	☐ Ja				☐ Nein		
	Wenn Ja, wie ist die Gesprächsbereitschaft?	sehr gut	1	2	3	4	5	schlecht

4.8 Anmerkungen: ____________________

5. Praktikum – Softwaretools

5.1 Wurde in dem zugehörigen Praktikum die Anwendungen der Lehrinhalte verdeutlicht?	sehr gut	1	2	3	4	5	nicht aus-reichend
5.2 Bedeutung des Praktikums							
Vertiefung des Vorlesungsstoffes	sehr gut	1	2	3	4	5	nicht aus-reichend
Erweiterung des Vorlesungsstoffes	sehr gut	1	2	3	4	5	nicht aus-reichend
5.3 Anspruch an Vorbereitung und Ausarbeitung	eher gering	-2	-1	0	1	2	eher zu viel
5.4 Erforderlicher Zeitaufwand für Ausarbeitung	eher gering	-2	-1	0	1	2	eher zu viel

5.5 Anmerkungen: ______________________________

6. Außere Beeinträchtigungen

6.1 Sitze/Tische	sehr gut	1	2	3	4	5	schlecht
6.2 Tafel	sehr gut	1	2	3	4	5	schlecht
6.3 Projektion	sehr gut	1	2	3	4	5	schlecht
6.4 Licht	zu klein	1	2	3	4	5	schlecht

6.5 Anmerkungen: ______________________________

7. Eigener Einsatz

7.1 Wie oft haben Sie diese Veranstaltung in diesem Semester besucht? Bitte die konkrete Anzahl ankreuzen	1-3	4-6	7-9	10-12	13-15

7.2 Wie oft arbeiten Sie die Vorlesung nach?							
generell	immer	1	2	3	4	5	nie
anhand des Skriptes	immer	1	2	3	4	5	nie
anhand der Mitschrift	immer	1	2	3	4	5	nie
anhand der Literatur	immer	1	2	3	4	5	nie

7.3 Was ist Ihre Motivation diese Lehrveranstaltung nach zu arbeiten?

☐ Lerngruppe ☐ Ubungsaufgabe zur erledigen ☐ Bearbeiten der Gruppenarbeit

☐ Das Thema interessiert mich ☐ Klausur vorzubereiten ☐ Andere ____________________

7.4 Anmerkungen: ______________________________

Form Nummer: 1

Form Nummer: _____1__________

Liebe Studentinnen und Studenten,

bitte behalten Sie diesen Teil der Evaluation für Ihre eigenen Unterlagen.

Für eine laufende Studie über die Zusammenhänge zwischen dem „Fragebogen zur Beurteilung der Lehre im Fachbereich" der Studenten und deren Prüfungsnote sind wir auf Ihre Hilfe angewiesen.

Nachdem Sie das endgültige Ergebnis für Eingebettete Systeme II erhalten haben, wären wir Ihnen dankbar, wenn Sie eine E-Mail an Frau Sim (sim@uni-kassel.de) mit den folgenden Informationen.

1) Form Nummer (siehe oben)
2) Prüfungsnote für Eingebettete Systeme II

schicken würden. Nur mit Hilfe Ihrer Form Nummer und Prüfungsnote sind wir in der Lage, die notwendigen Zusammenhänge, welche für diese Studie benötigt werden, zu ziehen.

Bitte beachten Sie, dass diese Studie in **keiner Weise** Einfluss auf Ihre Prüfung hat.

Vielen Dank für Ihre Unterstützung.

Mit freundlichen Grüßen,

Fachgebiet Eingebettete Systeme

10.4.3 5IMPLe Form for SS2009

Halbzeit Feedback zur ES2 Sommer Semester 2009

Name: ____

Matrikelnummer: ______________________________

Liebe Studenten,

wir haben fast Halbzeit des Semesters. Bitte schreiben Sie in den folgenden Tabellen die Themen, die Sie für sich für wichtig halten oder nicht verstanden haben. Wir werden versuchen die Fragen zu beantworten.

Danke fürs mitmachen!

Gruß,
FG ES

Kapitel 1a: Bussysteme (Kapitel 6 ES 1)

Wichtige Themen in diesem Kapitel	Die Themen, die ich nicht verstanden habe

Kapitel 1: Requirements Engineering

Wichtige Themen in diesem Kapitel	Die Themen, die ich nicht verstanden habe

Kapitel 2a: System Modellierung – Strukturierte Analyse

Wichtige Themen in diesem Kapitel	Die Themen, die ich nicht verstanden habe

Kapitel 2b: System Modellierung – SysML

Wichtige Themen in diesem Kapitel	Die Themen, die ich nicht verstanden habe

Seite 1 von 1

ES2 Sommer Semester 2009

Name: ____

Matrikelnummer: ______________________________

Liebe Studenten,

Bitte schreiben Sie in den folgenden Tabellen die Themen, die Sie für sich für wichtig halten oder nicht verstanden haben. Wir werden versuchen die Fragen zu beantworten.

Danke fürs mitmachen!

Gruß,
FG ES

Kapitel 3: Automatisierungstechnik

Wichtige Themen in diesem Kapitel	Die Themen, die ich nicht verstanden habe

Kapitel 4: Programmierung von Eingebettete Systeme

Wichtige Themen in diesem Kapitel	Die Themen, die ich nicht verstanden habe

Kapitel 5: Verifikation, Validierung und Test

Wichtige Themen in diesem Kapitel	Die Themen, die ich nicht verstanden habe

Seite 1 von 1

10.4.4 Group Work B Evaluation Form

ES 2: Gruppenarbeit Teil G

Name:

Gruppennummer:

	stimme ganz zu				stimme überhaupt nicht zu
Wie effizient hat ihre Gruppe zusammen gearbeitet, um die Gruppenarbeit zu lösen?	☐	☐	☐	☐	☐

Bewerten Sie die Mitarbeit von Ihren Gruppenmitgliedern.
(Bitte die eigenen Mitarbeit nicht bewerten)

Vorname	Nachname					
		☐	☐	☐	☐	☐
		☐	☐	☐	☐	☐
		☐	☐	☐	☐	☐
		☐	☐	☐	☐	☐
		☐	☐	☐	☐	☐
		☐	☐	☐	☐	☐
		☐	☐	☐	☐	☐
		☐	☐	☐	☐	☐
		☐	☐	☐	☐	☐
		☐	☐	☐	☐	☐

Geben Sie ein spezifisches Beispiel an, bei dem Sie von ihren Gruppenmitgliedern bzw. der Gruppenarbeit gelernt haben und das Sie ohne Gruppenarbeit nicht gelernt hätten.
.

Geben Sie ein Beispiel an, was die anderen Mitglieder von Ihnen gelernt haben.

	stimme ganz zu				stimme überhaupt nicht zu
Der Inhalt der Vorlesung ist relevant für die Gruppenarbeit.	☐	☐	☐	☐	☐
Der Inhalt der Übung ist relevant für die Gruppenarbeit.	☐	☐	☐	☐	☐
Die Benutzung des gleichen Anlagebeispiels (Stempel / Sortieranlage) hat mir geholfen die Lehrinhalte besser zu verstehen.	☐	☐	☐	☐	☐
Die Inhalte der Lehrveranstaltung reichteaus, um mit der Gruppenarbeit anzufangen.	☐	☐	☐	☐	☐
Die Termine für Teilaufgabe A bis C haben mir geholfen die Aufgabe besser zu verstehen und zu lösen.	☐	☐	☐	☐	☐

	4V:2Ü	3V:2Ü	2V:2Ü	2V:3Ü	2V:4Ü
Um den Inhalt der Lehrveranstaltung zu verstehen, sollte der Anteil von Vorlesung (V) zur Übung (Ü) ………. sein.	☐	☐	☐	☐	☐

	stimme ganz zu				stimme überhaupt nicht zu
Die Gruppenarbeit hat mich herausgefordert, mehr über die Entwicklung einer Anlage zu lernen.	☐	☐	☐	☐	☐
Durch die Gruppenarbeit, habe ich die Lehrinhalte besser kennen gelernt.	☐	☐	☐	☐	☐

	<1Std.	1≤Std. <2	2≤Std. <3	3≤Std. <4	> 4 Std.
Wie viele Stunden (Std.) haben Sie pro Woche für die Gruppenarbeit eingesetzt?	☐	☐	☐	☐	☐

	<1Std.	1≤Std. <2	2≤Std. <3	3≤Std. <4	> 4 Std.
Wenn Sie letztes Semester ES1 gehört haben, wie viele Stunden (Std.) pro Woche haben Sie sich für das Popquiz vorbereitet?	☐	☐	☐	☐	☐

Was hat Ihnen an der Gruppenarbeit besonderes gefallen?

Was können an der Gruppenarbeit verbessert werden?

Bitte Antworten Sie nur bei der Teilaufgabe bei der Sie beteiligt waren.

Teilaufgabe A

1 Nennen Sie zwei wichtige Punkte über Requirements Engineering, die Sie durch die Gruppenarbeit gelernt haben.

a

b

2 Nennen Sie zwei Herausforderungen auf die Sie während des Requirements Engineering gestoßen sind.

a

b

Teilaufgabe B und Teilaufgabe C

3 Nennen Sie zwei Zwecke von Modellierungsansätzen für SA/RT oder SysML.

a

b

Teilaufgabe D und Teilaufgabe E

4 Nennen Sie zwei Vorteile von "Requirements Engineering" und Systemmodellierung zur Implementierung der Anlage.

a

b

Teilaufgabe F

5 Nennen Sie zwei Punkte die Sie beim Test gelernt haben.

a

b

ISBN 978-3-89958-970-2