TEACHING SYSTEM INTEGRATION OF MECHATRONIC SYSTEMS

Z. Fan, M. Detlef, M. M. Andreasen and L. Hein

Keywords: system integration, mechatronics, design education

1. Introduction

Mechatronics has been an emerging subject in industry with the introduction of a large volume of mechatronics product to the market, ranging from the large scale products such as hybrid/electric cars, industrial robots and CNC machines, to the smaller consumer products such as digital cameras, DVD players, and programmable sewing machines. In addition, it gives rise to tremendous challenge for education of mechatronic engineering, because mechatronics is an area encompassing multidisciplinary knowledge of mechanical engineering, electrical/electronics engineering, and information technology. How to integrate designs in various domains is a key of mechatronic designs. But many courses of mechatronic designs today treat teaching individual domians relatively independently. The topic of system integration is not emphasized in these courses, and therefore relationships of designs in different domains can not be seen clearly. This paper explains the efforts of the Technical University of Denmark (DTU) to embed the education of mechatronics in its educational program of ‘Design and Innovation’, and the use of a special type of Unified Modeling Language (UML), statechart, to teach the topic of system integration in the course of ‘Design of Mechatronic Systems’.

2. Background of the Course: Design and Innovation Program as a Framework

In 2002 the Technical University of Denmark introduced a radically new engineering education: Design & innovation. The five year programme leads to a B.Sc. after three years of study, and to a M.Sc. after a further two years. The education is designed to enable the student to master the creation of complex products and systems, in a mix of mechanical, electronical, software/IT, and service elements.

Throughout the education programme, each semester has its specific project for the students to work with. At the core of the programme there is a specific course in Design of Mechatronics, taught in the third semester. Because a course in design of Mechatronics must be supported by other courses to provide the students with supplementary competencies, the curriculum of the Design & innovation education serves as a framework for the course in Design of Mechatronics. [Hein et al. 2005]

The courses of the program falls into four categories of: projects, engineering topics, basics, and freely selected courses, and are shown in table 1.

Table 1. The Design & innovation bachelor curriculum matrix, with the course in Design of Mechatronics in the third semester (after [Boelskifte and Jørgensen 2005])

ECTS / Semester	5	10	15	20	25	30

Bachelor in Design & innovation	1	User oriented design	Visual communication	Use and design of products	Mechanics and materials	
	2	Product analysis and redesign		Technology analysis	Objects and programs / Electronics	Linear and differential mathematical models
	3	Design of Mechatronics	Industrial design	(freely selected)	Dynamics and oscillations	
	4	Workspace design	Design of work processes	(freely selected)	Electronics / Objects and programs	Thermo-dynamic modeling
	5	Product-service systems	Product life and environment	(freely selected)	Light and materials	Signals and transformations
	6	Bachelors project		Scenarios and concepts	(freely selected)	Fields and fluid mechanics

3. The Course of Design of Mechatronic Systems

3.1 Course characteristics

The course in design of Mechatronics is a combination of lectures, exercises, and project work in an approximate proportion of 1:1:8, at a sum total of 200 hours per student. The projects are carried out in teams of 3 – 5 students who are evaluated on the basis of a collective report and a collective presentation of the projects process and results. The lectures and exercises are intensive in the first half of the period, and then tapers out to make more space for the project activities.
Learning objectives of the course is stated as below.
After following this course, you will be able to:

- Define design specifications given design requirements
- Obtain alternative design concepts given design specifications
- Decide a proper functional structure given design concepts
- Select components within proper domains given a functional structure
- Integrate components into a mechatronic system and analyze the behavior of the integrated system
- Present and report your design as a member of a design team

In summary, we have two core elements in this course. One is related to technical skills including product specification, conceptualization, synthesis, analysis, and so on. The other relates to soft skills such as communication, presentation, team work spirit, and so on. Accordingly, we use two forms of assessment. One is project evaluation, which focuses on assessing the technical skills of students. Although only the final report are evaluated, students are required to submit also a proposal report, and three milestone reports. They form a good base for writing the final report. The final report accounts for 66.6% of the total score. On the other hand, oral examination accounts for 33.3% of the total score, in which students are required to make posters for their designs and present their designs in groups. An external censor, the research manager of Danfoss, helps us to evaluate performance of the students.

3.2 Course elements

The course is a combination of elements which together supports the students in their project work, i.e. Mechanical Design (ME) domain, Electronic/Electrical (EE) design domain, and Information Technology (IT) design domain.
In the project report, the quality of choice of solution and synthesis will be evaluated in the four dimensions:

- Product synthesis
  System requirements, user preference, product specification, Function-Means tree reasoning.

- Mechanical design
  Drafts and/or CAD models for all mechanical structures embedded with machinery components.
- Electronics/Electrical design
  Selection and verification of sensors and actuators, signal conditioning and driver circuitry etc.
- IT design
  Statechart, selection and verification of implementation techniques such as microcontroller, or Field Programmable Gate Array (FPGA) etc.

Because a mechatronic system involves designs in various domains, it is a challenge to cover all the domains in one course in one semester. In addition, it is important that different domains of mechatronics engineering should NOT be taught separately –system integration of them is a key of mechatronic designs. Actually, the philosophy of system integration and optimization is the driving force for the forming of this emerging area of mechatronics in the late 70s. However, many courses of mechatronic designs today still follow a very traditional pedagogical way to teach individual domians relatively independently. The topic of system integration is not sufficiently emphasized in these courses, and therefore relationships of designs in different domains can not be seen clearly by the students. The following section will discuss our experience of teaching system integration using a special tool of UML Unified Modeling Language) – statechart [Priestly 2003].

4. Teaching System Integration Using Statechart

4.1 Why can statecharts be used to integrate designs of mechatronic systems

If we look at a mechatronic product, the first thing we may notice is the outer appearance, or the mechanical infrastructure. Later on, we may notice that at some places of the product, a sensor or an actuator is installed here and there, embedded within the mechanical infrastructure. These electrical components and the mechanical infrastructure compose the 'flesh' and 'bones' of the product. But what is the 'soul' to connect them into an integrated whole?
If we are lucky enough and can open the cover of the product, we may see some wiring connecting the sensors and actuators to a printer circuit board. This board usually has a embedded chip (or more) that defines the 'soul' of the system, i.e. the logic behaviors of the mechatronic product.
This 'soul' cannot be seen by our eyes, but can be described by statecharts. Statechart is one type of UML, which includes a cluster of tools to facilitate communication among programmers without the need of going into details of coding. Statechart is one of them specially tailored to describe the logic behavior of an embedded system, which is almost an alias of a mechatronic system. That is why we chose statechart as a tool to teach IT design of mechatronic systems. Many experienced industrial practitioner of mechatronic designs also confirmed that they believe statechart is one of the best tools to facilitate communications among members of a mechatronics design team.
In brief, a statechart has a number of states. Between each two states there may exist a transition. Accompanying each transition from one state to another, two types of state variables can be defined: messages and actions. Messages define the conditions to trigger the transition, and can be related to actual signals taken from the outputs of sensors; actions define the actions to be taken, and are actually executed by the actuators (see Figure 1.). While statecharts define the logic behaviors of a mechatronic system, the physical carriers of them may be microprocessors/microcontrollers, or FPGA chips, and so on. In essence, they are chips with input pins and output pins. The input pins connect to the output of sensors through wirings, and the output pins on the other hand connect to the input of actuators through wirings. In this way, the core design in the IT domain, which is illustrated through statecharts and embedded in e.g. the microprocessor, can be extended to the design in the electrical domain, which may include the selection of suitable sensors and actuators and their proper connections to the microprocessor board. To do the right selection, some engineering calculations are also needed (e.g. calculation of the torques and speeds the motors have to be run at, etc), and understanding of some physical characteristics is also essential (e.g. what is the working range of the photoelectric sensor? Is it in the right range to suit the application purpose? etc).

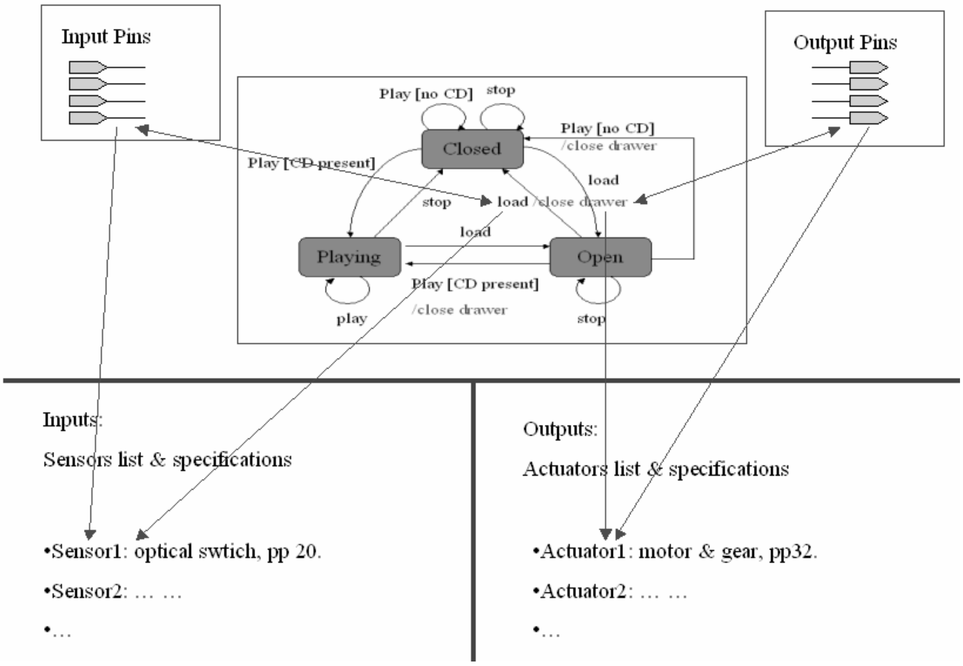
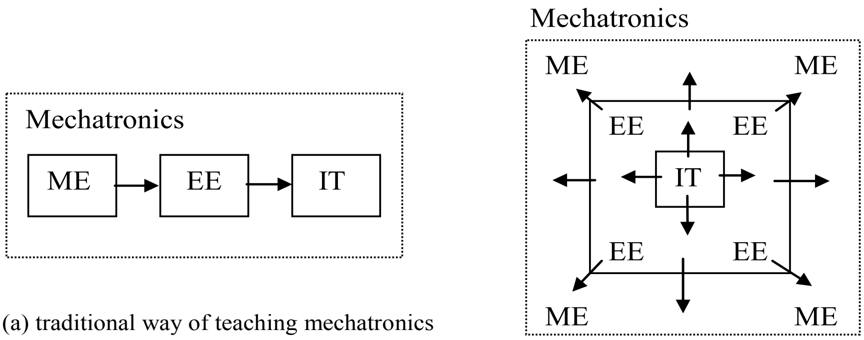


Figure 1. An example of an extended statechart for the CD player

The sensors and actuators shall be embedded somewhere in the mechanical infrastructure, which will be further defined by the design in the mechanical domain. The mechanical design also specify moving mechanisms and the overall structure of the system. Now you can see that from this analysis, we start with the core logic of the mechatronic system (defined in the IT design part), and then expand it to the design in the electrical domain, and finally to the design in the mechanical domain. This perspective, starting from the innermost core, and expanding step by step to the outer appearance, integrates the design of mechatronic systems better than the traditional perspective. A comparison of them is shown in Figure 2.



(a) traditional way of teaching mechatronics

(b) teaching mechatronics in a system integration perspective

Figure 2. Comparison of two different ways to teach mechatronics

In the course the students are required to deliver an A2/A3 page in their reports describing system integration of IT and EE using statechart, with good reference to disucssions of EE components (e.g. sensors, actuators, etc) in other pages. This page serves as the core technical page in the report, from which the readers will start to trace the other technical details of the design. Figure 3 shows an example from the students' milestone report.

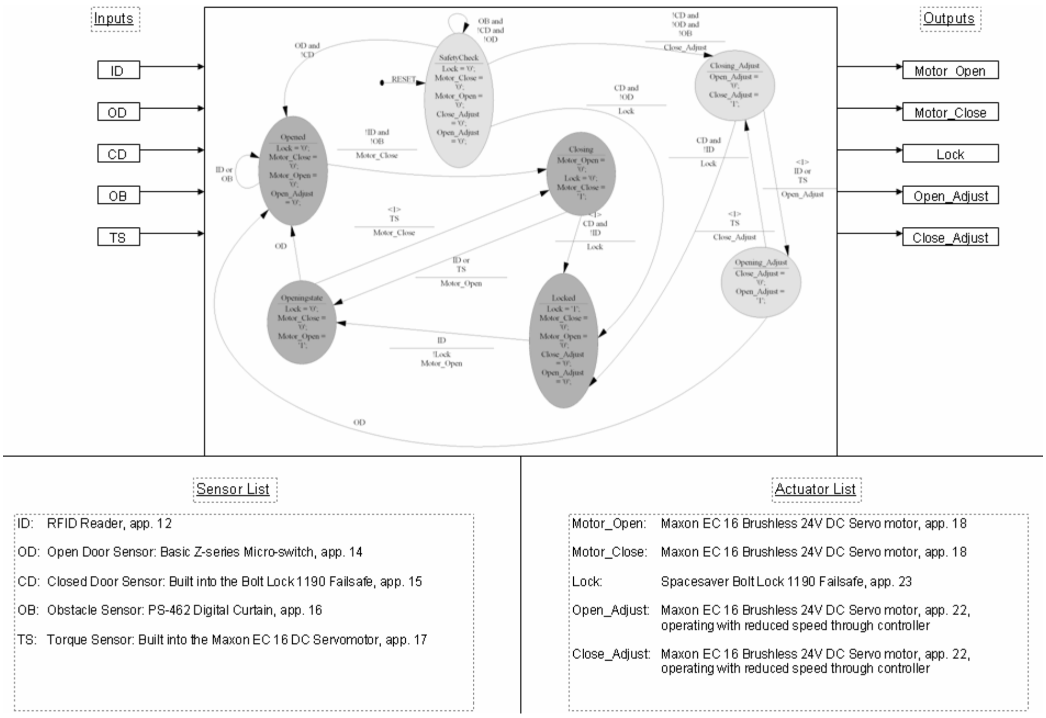


Figure 3. An extended statechart for the course project taken from a milestone report

4.2 Simulation and verification of statecharts using StateCAD

The use of software tools was also advocated in the course. Two particular tools used in the course include proEngineer, a CAD tool to sketch the drawings of mechanical structures, and StateCAD, a software to draw and simulate statecharts. StateCAD is a software tool provided by Xilinx Inc, in its package of Xilinx ISE 7.0 [Parnell and Mehta 2004].

Use of StateCAD proves to be very helpful. It was surprising to see how many logic errors can be produced when students generate statecharts without using the StateCAD previously. It turns out that people's imagination about systems' logic behaviors can be very erroneous, full of logic contradictions and imperfections. An advantage of using StateCAD is that it can detect logic errors, give out warnings, and refuse to compile if the errors are not corrected. After preliminary frustrations, and trial and error for some time, most students happily accepted the software, not only because it helped them to clean the logic errors, but also because it made them to understand how difficult it could be to design an error-free logic for their mechatronic products. Secondly, inputs and outputs pins are listed clearly in the simulation file of the StateCAD, which prompts the students to consider what sensors they should use to generate the inputs respectively, and what actuators they should use to take the 'orders' of the outputs of the statecharts. From my experience, pushing students to consider these questions can help them to obtain a better overview of the overall system to a large extent. Finally, after the statechart is free of logic errors, the software can compile the statechart, and convert it automatically into programming codes. This can free the designers from the tedious work of writing and debugging codes, and enable them to focus more on the creative work of logic design using the higher level modeling tool of statechart.

In the Xilinx ISE 7.0 environment, a proved statechart can be finally converted into bit streams and downloaded to the FPGA chip, which is the physical carrier of the design logic and can be readily used in the hardware implementation of the mechatronic system (the process is shown in Figure 4). Therefore, if we want to change the logic behaviors of the design, we just need to modify the statechart, and then repeat the process in Figure 4.

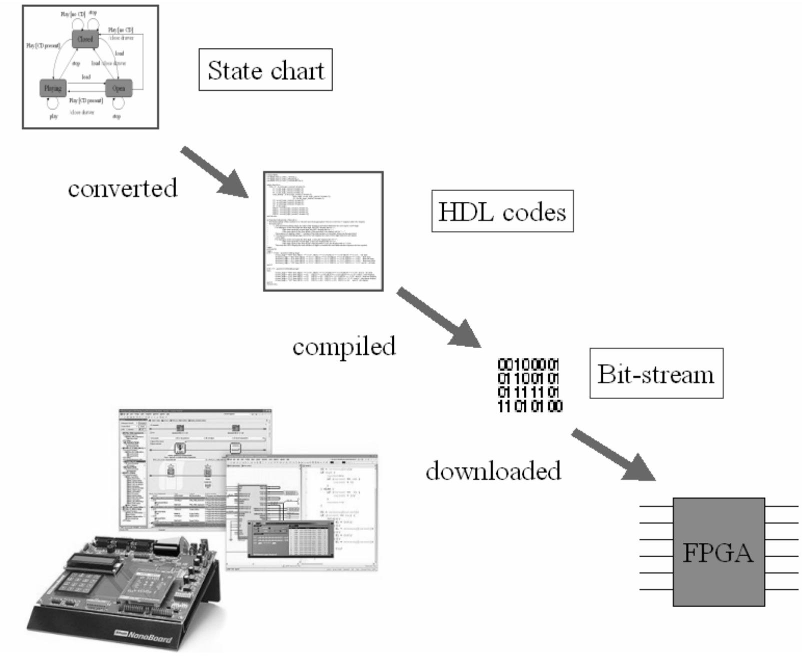


Figure 4. The process from designing of a statechart to its hardware implementation

5. Project assignment and results from the students

Design is practice-oriented and it is very important to cultivate practical design skills of students in course project work. The students were divided into 12 groups to carry out the assigned project in teams. They have abundant time to do the project: each Thursday morning in the 13- weeks period, and all days in the 3-weeks period. Teachers provide help to them in the classroom during all the project sessions of the course.

5.1 Project assignment

The project assigned in this course is about development of an intelligent door. A more detailed description is the following:

"The company NN develops and sells aids to be installed and used in private homes and at nursing homes, for surveillance, client- and staff assistance in connection with the elderly, handicapped or the retarded.

The company has asked for an evaluation of the prospects of an intelligent door, i.e. a door which opens and closes as required by the clients and the staff, while fulfilling the requirements for functionality, logical operation, privacy and safety.

The automatics of the door must offer an open door when a client passes with a rollator, a rolling walker or in a wheelchair, or when the staff passes with their hands full."

5.2 Deliverables from the students

A number of creative and high-quality designs were observed in the course, the following figures (Figure 5 to Figure 7) come from three examplar projects from students.

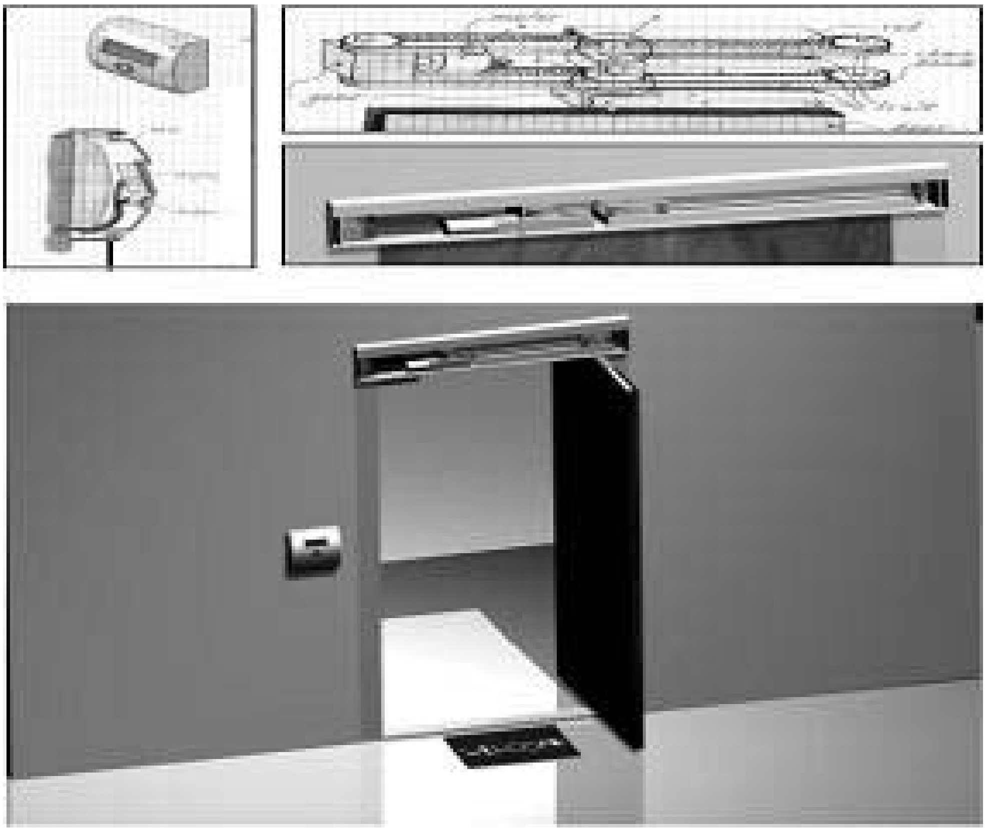


Figure 5. Drafts and CAD model of mechaical design of an intelligent door

Fig. 5 shows examples of drafts and CAD models for the mechanical design part of an intelligent door from a group of students within the course. It demonstrates that the students are able to investigate great details of the mechanical design.

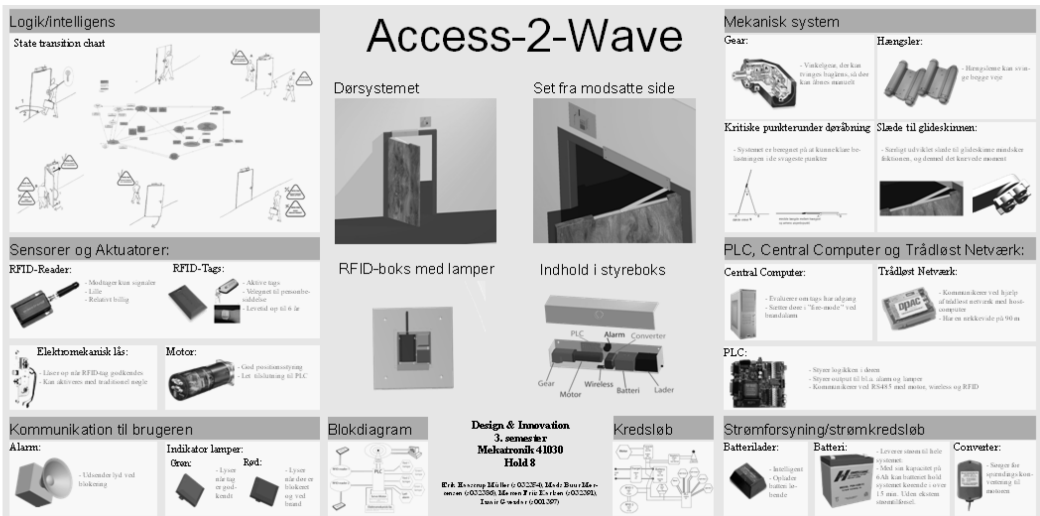
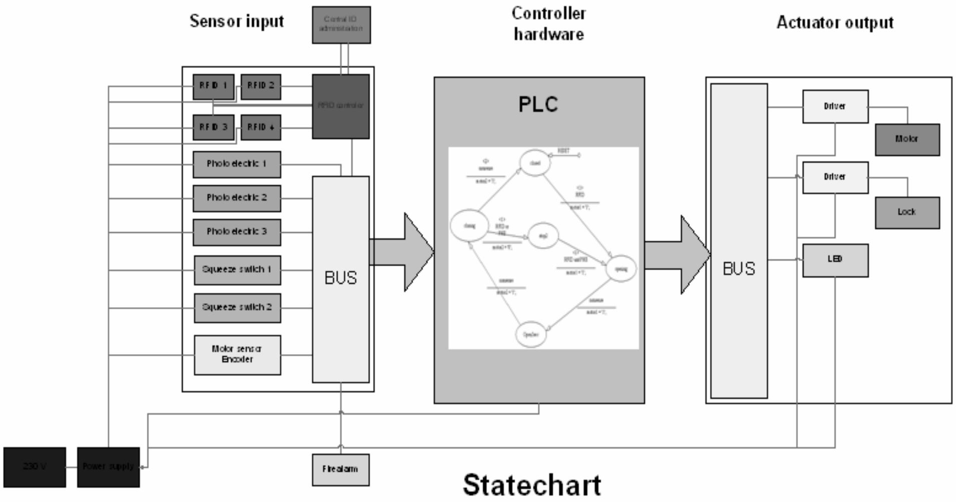


Figure 6. Overall electrical design of an intelligent door



Sensorlist

Sensortype	Manufacturer and type	No.	pp.
RFID	Unitek, T370	1	12
RFID	Unitek, T370	2	12
RFID	Unitek, T370	3	12
RFID	Unitek, T370	4	12
Photo electricsensor	Omron, E3S-C	5	15
Photo electricsensor	Omron, E3S-C	6	15
Photo electricsensor	Omron, E3S-C	7	15
Squeeze switch	Heavens doors special	8	16
Motor sensor	Encoder in the servo motor system	9	18
Firealarm	To be specified	10	20

Actuatorlist

Actuatortype	Manufacturer and type	No.	pp.
Servomotor	Maxon, Planetary Gearhead, GP 42 C. Ø 42mm	1	23
Lock	Lien Kuan, EM250 M 600 lbs	2	29
LED display	to be specified	3	30

Figure 7. IT design and statechart of an intelligetn door

Fig. 6 shows a poster for an overall electrical design of another intelligent door from a different group. It demonstrates that the students can articulate various technical aspects of the electrical design. Figure 7 gives a third group's design of a statechart for the intelligent door, which illustrates the integration of the system well and represnts a good example of design in the IT domain.

6. Reflections and discussions

Based on my impression from students' milestone reports and discussions with them, I can see that they obtained a much better understanding and overview of mechatronic designs after being taught system integration of mechatronic systems. They were also more active to ask me questions, and many of them are very in-depth. I have a feeling that the more technical details I touched in the course, the more else the students want to know, and more questions they will ask me about. In other words, they have found a key clue to investigate their mechatronic designs and self-motivedly identify knowledge to learn in the needed technical domains.

In this course we focused on logic integration of mechatronic systems. Another major topic of system integration is the integration of dynamical systems of mechatronic designs, which shall be added to the course syllabus later.

Another concern about the course is, now quite some students are not satisfied that they had no hand-on experiences on building up a real world mechatronic system. Even though I have shown them various components and devices in the lectures so that they can have more real-world impressions, and even demonstrated a complete motor control system (in which I actually run the motor and showed them how to change the speed and direction of the motor), they simply wanted to build and run the system by themselves.

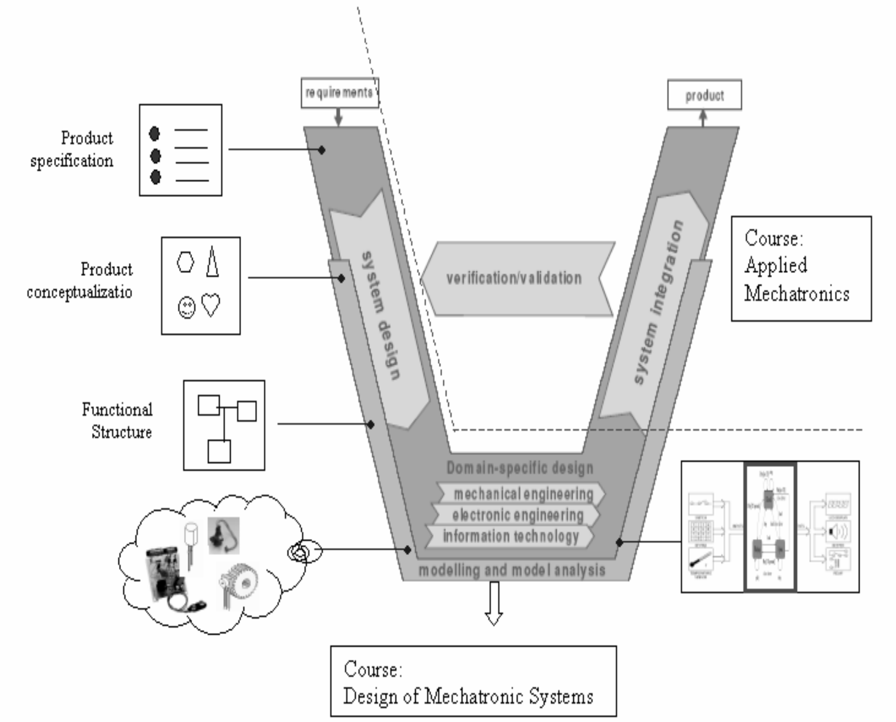


Figure 8. The two courses related to the V-model

Actually this expectation from the students is not without reasons. As we know, V-model is a principle used at the level of the product development process [Gausemeier and Michels 2004]. From the V-model shown in Figure 8, development of mechatronic systems has two paths, one is top-down (on the left side), the other is bottom up (on the right side). After conceptualising the design in a top-down 'system conceptual design' process, a bottom up 'physical system integration' process is followed to verify that the conceptual design can be physically realized as expected. Currently, this course only covers the 'system conceptual design' process, while the 'physical system integration' process is take care of by another course 'Applied Mechatronics'. The arrangement is as follows [Fan et al. 2005] :

"The first course, called 'Design of Mechatronic Systems', is focused on the conceptual design side and taught to the third-semester students. The main purpose of the course is to give students an overall vision of designing mechatronic systems, rather than to plunge them into technical details in any certain area. With a system view of designing mechatronic systems, students better understand connections in the content of the courses in their following education and help them to proceed more self-motivatedly to obtain knowledge in detailed subjects.

The second course, 'Applied Mechatronics', is focused on physical implementations of the conceptual designs and taught to the seventh-semester students. It gives students hand-on practices on how to build mechatronic systems."

Although the two courses balance the tasks of conceptual prototype and physical realization well, the current arrangement has the following problems. First, the two courses are too far away from each other in the curriculum. The students having taken the first course will have to wait for another two years to take the second one. Their interest may be lowered down dramatically by that time. Secondly, the project topics in the second course are not necessarily the same with the first one. Therefore, the students may not have a chance to verify their conceptual design considerations in the second course.

Solutions to the problems, which we have begun to discuss, involve introducing some laboratory experiments (e.g. experiments of FPGA development boards from Xilinx Inc.) in the first course, and moving the second course closer to the first one in the curriculum. All the discussions and reflections will lead to an updated syllabus and curriculum in the future.

References

Boelskifte P., and Jørgensen U. "Design and Innovation: Developing a Curriculum for Future's Design Engineers at the Technical University of Denmark," Engineering and Product Design Education Conference, Napier University, Edinburgh, UK 2005

Fan Z., Sørensen T., Conrad F., Andreasen M. M, Christensen G. K., Hein L., "Mechatronics Education: Conceptual Prototype vs. Physical Realization", 6th International Workshop on Research and Education in Mechatronics (REM) 2005, pp 397 – 402

Gausemeier J., and Michels J. S., "From Mechatronics to Self-Optimization," In: Proceedings of the O.M.P. - International Trade Fair for Optical and Microtechnology Products, 2004.

Hein L., Fan Z., Andreasen M. M., "Design & Innovation – a New Framework for Teaching Mechatronics", Chinese Mechanical Engineering Society Annual Conference,, Chongqing, China, 2005

Parnell K. and Mehta N., "Programmable Logic Design Quick Start Handbook", 2004, Xilinx, Inc.

Priestly, M.: Practical object oriented design with UML. McGraw-Hill, ISBN 00.77.103993, 2. ed., 2003

Zhun Fan, PhD
Department of Mechanical Engineering
Section for Engineering Design
Technical University of Denmark
Building 404, DTU
DK – 2800 Kgs. Lyngby, Denmark
Tel.: +45 45256271
Email: zf@mek.dtu.dk