Chapter 47
How to Introduce Mathematical Modelling in Industrial Design Education?

Geert Langereis, Jun Hu, and Loe Feijs

Abstract With competency based learning in a project driven environment, we are facing a different perspective of how students perceive mathematical modelling. In this chapter, a model is proposed where conventional education is seen as a process from mathematics to design, while competency driven approaches tend to have an inverted sequence. We assume there is a virtual barrier for on-demand learning with regards to the mathematical modelling layer under the layer of technical skills. Several successful attempts were done in the past to remove the technology skill from the chain in order to make the opportunities of modelling visible. After experiencing the modelling competency in such a setting, students can beneficially deploy it for technology. We evaluated this model based on a learning activity which was changed from traditional education into competency centred learning.

1 Introduction

The Department of Industrial Design at the Eindhoven University of Technology distinguishes itself by a focus on the design of intelligent systems, products, and related services. The user-focused application area is continued in the research and educational system by means of a competency-centred learning approach. The associated reflective transformative design process (Hummels and Frens 2009) is based on learning and developing "from doing" and "by doing" and is, as such, highly dependent on action research. This is where it distinguishes fundamentally from classical design engineering approaches. In classical engineering education, a foundation of mathematical tools and thinking is taught first. Secondly, a scientific

G. Langereis (✉) • J. Hu • L. Feijs
Department of Industrial Design, Eindhoven University of Technology, Eindhoven, The Netherlands
e-mail: geert.langereis@gmail.com

G.A. Stillman et al. (eds.), *Teaching Mathematical Modelling: Connecting to Research and Practice*, International Perspectives on the Teaching and Learning of Mathematical Modelling, DOI 10.1007/978-94-007-6540-5_47,

notion is taught, depending on the department being physics, chemistry, electronics, computer science, or any other area. After having acquired these two layers of background, the engineer can tackle practical problems in a design engineering approach. With our competency centred educational system, we start with engineering, to discover the science behind it by playing and exploring. The consequence is that the mathematical background is experienced as a deep third layer, which is not evidently a skill noted by our students as a next investment in their self directed competency development.

This chapter will explain the competency based learning system first. Next the aspects of modelling required for academic design students are discussed. In a subsequent section, some examples of teaching modelling skills are explained. In a section about balancing design engineering and modelling, a vision of the consequence of the educational system on the learning behaviour is given. With this model we can propose how teaching activities can be improved to optimise the learning outcome. The theory and proposal is evaluated based on our experience with a specific assignment on microcontrollers.

2 Competency Based Learning as Optimised for Design Education

Industrial Design (ID) is one of the nine departments of the Eindhoven University of Technology (TU/e), and was established in 2001. In consultation with industry and government, ID focuses on the development and design of user-friendly interfaces for intelligent systems, products and related services in multimedia environments. Current research topics include Human-Computer-Interaction, multi-modal interaction, perceptive user interfaces and aware environments (also known as “ambient intelligence”), as well entertainment computing research. The research aims to provide generic models and frameworks in the domains of perception, cognition, interaction and communication to the extent that these fields are relevant to the design of technical products and services.

The focal area of the Department of Industrial Design, being the creation of intelligent systems, products and related services, has resulted in an educational system that differs significantly from the other departments. For designing intelligent products at the Industrial Design Department, we are facing the problem that theory for interaction with humans is unpredictable, or at least complex. Therefore, in contrast to the knowledge driven approach of the other departments, Industrial Design has implemented an educational system that is based on action research. Education starts with practical work before students discover which scientific foundation is involved. This learning model applies particularly well for industrial designers because the industrial designer has to develop contexts of use, actively explore concepts, evaluate alternative solutions, and bring new products to the world.

The educational model is in agreement with social constructivism (Vygotsky 1978), the sociological theory of knowledge. In education, social constructivism stresses interaction over observation. As a consequence, students in our competency based system act and interact with objects, teachers and other students to learn. They need to become experts who create, apply and disseminate knowledge, and continuously construct and reconstruct their expertise in a process of life-long learning. Meeting the goals of education requires a high consistency between instruction, learning and assessment. With the new social constructivism insights in instruction and learning, the paradigm of assessment had to be changed as well (Birenbaum 2003; Segers et al. 2003). Although similar systems of Competency-Centered Learning programs are seen (e.g., Pereira et al. 2009), at the Industrial Design Department of Eindhoven University of Technology, it is implemented in every layer of the organisation, and already optimised over the past 10 years.

To assess the quality of the students, and to monitor their progress, ten competency areas are defined. The tenth competency area is *Descriptive and mathematical modelling*. This makes clear that the ability to describe models and to find the mathematical models behind concepts is seen as one of the skills of an industrial designer. The other competency areas are *Self-directed and continuous learning*, *Integrating technology*, *Ideas and concepts*, *Form and senses*, *User focus and perspective*, *Social cultural awareness*, *Designing business process*, *Design and research processes* and *Teamwork and communication*. In practice, it is expected from the students that they can integrate all ten competences into a single design process. The ten competencies are primarily learned from projects (3 days per week), and taught on-demand in classes (2 days a week). It is essential to understand that none of the classes are mandatory. Until the academic year 2012/2013, there was no class available for basic mathematics. Students who had to apply mathematics in their project, had to learn from an expert or external resources. Nevertheless, awareness of the potentials of mathematics is an intended learning outcome.

3 Interpretation of the Modelling Competency for Design

The definition of the competence "Descriptive and Mathematical Modelling" is

> Being able to create and apply descriptive and mathematical models by using formal and mathematical tools, in order to justify design decisions and support the design of complex, highly dynamic and intelligent systems.

Understanding and mastering methods and tools for descriptive modelling enables students to describe relationships between parameters resulting in system behaviour. It is the foundation of simulation and optimisation. There is a strong link to the ability to analyse complex problems: to identify structures before tackling partial problems and to work towards a solution structurally.

When mathematical modelling is put into practice, it can be subdivided into four skills for gaining system insights. The first is where models are used for *analysing a complex problem* by breaking it into pieces. In this case, the model can be a state diagram or a flow chart, and does not necessarily have to be finalised into a numerical model. It is a method of communication about the problem with others or with oneself lowering the cognitive load by drawing systems on paper. The model can be the first step to translate the problem into solutions. Based on the state diagram, a designer can evaluate options and process flow.

A second modelling deployment for design is to *identify behaviour and dynamics* of systems. The notion of feedback systems, second order dynamic systems and, for example, phenomena like friction, are typical engineering skills, almost a craftsmanship, resulting in a predictive design process. A designer can prevent oscillations in a system by identifying mass-spring systems, for example.

The *predictive power* of models becomes strongest when a numerical mathematical model is implemented. The model can consist of closed form equations, or of a finite element simulation. Closed form equations normally can be solved to find design criteria or to exclude options. However, this type of modelling, which is highly dependent on calculus and numerical mathematics, is normally not seen from our students because of the absence of basic mathematical education in our offered classes.

The fourth skill is to eliminate options before building them. This is part of *evidence based design* where calculations are used to underpin design choices. To do this, estimations or calculations are made of an envisioned implementation in order to prove that the chosen solution is correct or that the design choice is the optimum solution.

In the next section, these four skills of modelling are identified with some examples of educational innovations at our Industrial Design Department. It is worth noting, that in practice the competency "descriptive and mathematical modelling" is closely related to the competency "integrating technology". Although developing technology is not the core business of the department, hardware is needed as a substrate to explore intangible concepts.

4 Some Implementations of the Modelling Competency

Previously, a teaching method (Hu et al. 2007) was presented to teach students to understand object-oriented design principles and formal software specification methods up to a level suitable for communication with software experts. The method was based on exploring a set of simple interaction rules by means of acting. Students became software objects (or "classes" in software terminology) themselves and could so transform acted behaviour into state diagrams. Such a practical realisation of a complex concept as object-oriented programming helped students to understand contexts, evaluate design ideas, explore new ideas and communicate designs to an audience. This learning activity is an example of the first modelling skill *analysing a complex problem* and appeared to be an easy way to make state diagrams explicit. The chosen "acting-out" methods can be seen as a strategy to educate modelling without having a technological frame of reference.

In van der Vlist et al. (2008) a method to teach the abstract concept of "machine learning" to students was explained. In this case, we did not remove the technological substrate completely, but we replaced it by a platform with which most students are comfortable: Lego Mindstorms NXT. The related modelling skill is *identify behaviour and dynamics*. With the ambition to teach the students to see patterns, the mathematical background was not omitted. For both reinforcement learning using the complex Q-learning method, and voice command learning using neural networks, the underlying principles were explained to the students using equations.

In the first example, the technology was removed. In the second example, technology was replaced by a simplified vehicle as most students are confident with Lego. This was done to bring the model and the real world as close as possible. In other words, the mathematical modelling is decoupled from the hardware/software substrate. However, when the teaching activity is about hardware or software, this is not always an effective option. When teaching programming, students must write code in a commonly accepted language like C or Java. This is done by focusing on the creative part of programming (Alers and Hu 2009) and using a robotic platform for a practical approach. Although this is not a modelling nor mathematics assignment, it proves that there is room to bring students into a state where they may discover that modelling skills, *analysing a complex problem* by means of state diagrams and *identify behaviour and dynamics*, have become within reach.

The *predictive power* skill of mathematical modelling is implemented amongst others in a course about geometrical principles (Feijs and Bartneck 2009). In that course tesselations are used to create plexiglass forms. A tesselation is a collection of plane geometries with no overlaps and no gaps. Industrial Design students were asked to create tesselations by using mathematical software like Mathematica instead of the usual visual drawing tools. The didactic of this approach is that students are empowered in their success of creating when they start to express patterns in equations. Again, a setting is created where technology is not the limiting factor when students explore their thoughts.

Finally, the modelling skill of *evidence based design* is implemented in an assignment on the basics of electronics. Electronics, especially analogue electronics, is seen as the toolset to give concepts "eyes and ears" by means of sensors and actuators. The learning goals are mainly limited to (1) switching actuators with transistors and (2) placing resistors to limit currents, and (3) low-pass and high pass filtering of sensor signals. For all three learning goals, one has to calculate currents and voltages to pick the right electronic components immediately: there is no efficiency in electronic design by iterative trial and error. We empower the students by three approaches. First, we have created a low-threshold electronics studio. The electronics assignment includes a guided workshop in the studio after which students are found there on a regular basis. Assistants are always available in the atelier for answering questions. Secondly, the mathematical skills are reduced to specifically solve the three learning goals as mentioned above. Finally, specific building blocks are identified based on components available in the electronics studio (i.e., shift registers, a limited set of sensors, etc.) and are well documented on the intranet. This approach appeared to be successful.

5 Balance Between Design Engineering and Modelling

Traditional engineering education and strategies work from theory towards practice. This means that, roughly speaking, students are learning mathematics in the first year, scientific theory and practice in the second and third, before they can do the practical applied engineering work in the end or during the masters phase. In Fig. 47.1 this is represented as a three-stage approach from left to right. The mathematical foundation is seen as the base for understanding science, which in its turn is needed to create new things. In the representation, the block "science" is used in a broad sense: it includes computer code, fabrication techniques, drawing skills and electronics. The right block represents all practical work to integrate scientific knowledge into a prototype or product and to explore the impact in our society.

As already explained in the introduction, the vision of the Industrial Design Department in Eindhoven to work on intelligent systems, products and related services, resulted in the implementation of a competency centred education model. This is done by starting with practical realisations (on the right of Fig. 47.2);

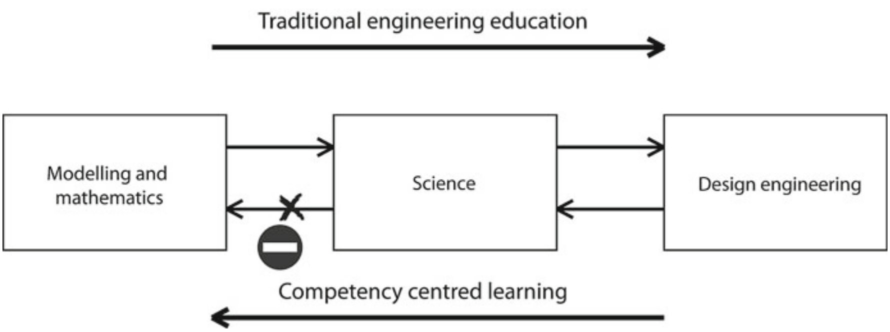

Fig. 47.1 One-way learning direction in traditional education and the opposite direction in competency centred learning

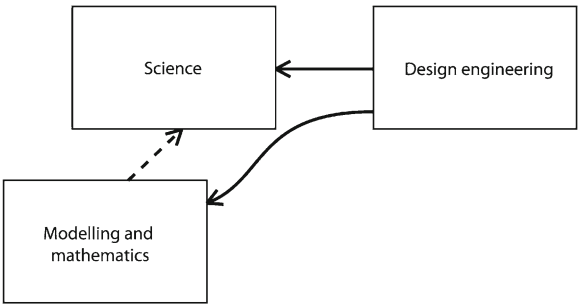

Fig. 47.2 Learning strategy for stimulating the mathematical modelling skill optimised for competency-based learning

scientific and engineering backgrounds offered on-demand. From that perspective, the mathematics and modelling question comes third, instead of being the foundation of our thinking. The inverted execution method is placing challenges on how to educate and how to do research. Design students prefer to explore using tangible artefacts, not with mathematical formulas.

For Industrial Design students in the flow of their work there is a virtual barrier when going from science to the natural need for mathematics. We experience this in their way of working and we can only guess the reason. It appears that investing in a deeper layer of abstraction is not seen as worth the effort. In the approaches discussed in the previous section, we assumed the barrier of scientific knowledge is too high to see the value in the mathematics and modelling skills; therefore, these approaches were based on lowering or removing the scientific substrate. This means in terms of Fig. 47.1 that the modelling and mathematics competency has been placed next to science as shown in Fig. 47.2.

What we are aiming at with this approach is that the opportunities of the competency *descriptive and mathematical modelling* can be experienced without being limited by a lack of technological skills. Once this is experienced, the dotted line may be put in practice in future projects, which is now in the natural direction.

6 Education Example: *Introducing Microcontrollers Assignment*

An assignment in the core of designing intelligent systems is about deploying microcontrollers for realising prototypes. At the Department for Industrial Design, assignments are learning activities offered to Bachelor students. Assignments consist of 6 weekly lessons of two contact hours, plus 36 h of self-study and/or practical work. In the *Introducing Microcontrollers Assignment* the self-study mainly consists of a design case. This assignment was originally organised by the Department of Electrical Engineering of our university. As of the academic year 2010–2011, the assignment was transferred to the responsibility of Industrial Design because "the message of the assignment did not reach the students". In the first term (Q1) of the academic year 2010–2011 the assignment was given in the old style, to discover how it could be improved. Afterwards, some adjustments were made which changed it from "traditional engineering education" (technology push) to "competency based learning" (technology pull from design perspective). Next, it was repeated in the fourth term (Q4). This is an excellent opportunity to verify the difference between the original model of Fig. 47.1 and the alternative learning strategy model of Fig. 47.2. The execution in Q1 and Q4 was the same in the sense that the offered theory was similar (microcontroller architecture, C-programming, on-chip hardware, hardware interfaces, a system design), and the design cases were the same. The cases consisted of designing an electronic game. The exact descriptions of the case for Q1 and Q4 are:

Introducing Microcontrollers Assignment (for Q1)

Measure the time between pressing button A and button B and communicate time in milliseconds over the serial bus.

- Hint 1: Use timer-interrupts to define timing
- Hint 2: A third button may be needed to reset the game
- Challenge 1: Implement as a small game. Player 1 can press a button any time, player 2 has to respond within 500 ms

Introducing Microcontrollers Assignment (for Q4)

Make a game. One of these:

- Make a “high striker”: a game where you have to hit a pressure sensor. Implement such that people want to play more
- Reaction game: when person 1 presses a button, person 2 has to react within a reasonable time by pressing a second button. Test to find what a reasonable time is. Find a feedback method: a buzzer or LED for success or fail. Make it such that either player 1 or 2 can do the first press
- Colour memory game: Player 1 mixes two or three LED colours (with pot meters) into one RGB colour, presses a button and puts the pot meters in a random position. Player 2 has to memorise the colour and has to reproduce it. The microcontroller determines whether you are close enough.
  - Explain design choices, if possible by calculations
  - Show state diagram (or other used abstraction of system)
  - Evaluate functionality (user test?)

So, in Q1 the technical perspective was chosen by starting with a technical problem of measuring an interval. In Q4, the question started with describing the application, and asked for technical insights later. To substantiate the change from technology push to technology pull was further substantiated by three other modifications. First, we asked for an appropriate housing for the game to be made to focus on the user experience, rather than seeing the code plus circuit as the end result. This was accompanied by a lecture with examples of attractive functional casings. Next, one lecture was added on how to communicate about a microcontroller system. This was needed for debugging, to find effective details from experts, and to structure the problem before solving it. In that explanation there was an introduction to state diagrams to transfer concepts into programmable solutions. This is a method to create the direct link from design needs towards modelling, while bypassing the hardware state (curved arrow in Fig. 47.2). Finally, in Q1 the

Table 47.1 Scoring criteria for microcontroller assignment

Criterion	When scored?
Analysing a complex problem	State diagram, flow chart, notion of design choices
Identify behaviour and dynamics	Insight, explore, characterize a sensor, A/D input window consideration, sample rate consideration
Predictive power	Equation, FEM
Evidence based design	Eliminate options by calculations, calculate transistor operational point, power consumption calculation
Box	Electronics is packaged, integration, form and senses

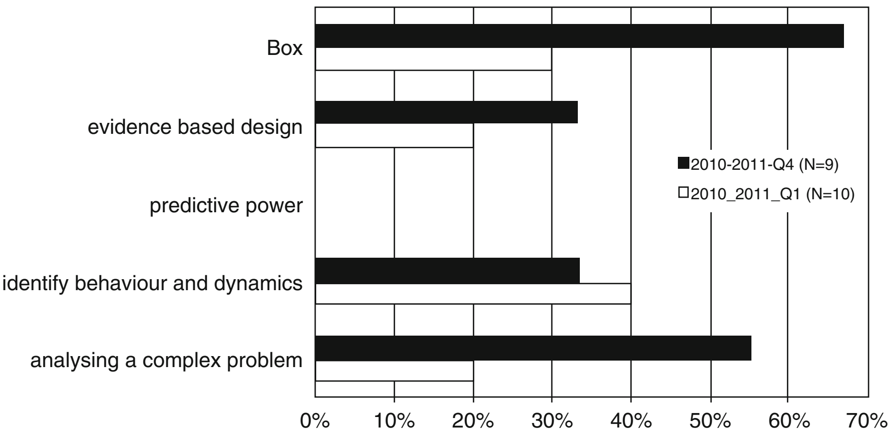

Fig. 47.3 Scores on four mathematical modelling skills before and after changing the content of *Introducing Microcontrollers Assignment*

introduction questions, before the design cases, were about specific technical functions, like "timers" and "sampling for A/D conversion". In Q4 the introduction questions were more about becoming familiar with the microcontroller: connecting it, writing subroutines, playing with communication between computer and the microcontroller board.

To compare Q1 to Q4, we assessed the end reports for the four skills of modelling for design. In addition, we evaluated whether the students worked from the perspective of the end-product; so, whether the end result has a functional shape or packaging. There was however one risk in the evaluation: the lecturer, being the first author of this chapter, did the evaluation of the outcome himself. To avoid a biasing effect, a clear scoring list was made first in order to have an objective set of criteria (Table 47.1).

In Q1 there were ten groups, in Q4 nine. The scores are shown in Fig. 47.3. It can be seen that the number of groups creating a packaged functional game increased from 30 to 66 %. This is interpreted as a perspective change from pure technology to the user or end result. In the new setting the skill of analysing a problem has been stimulated much better. This was mainly seen in the communication of students in terms of state diagrams, which immediately resulted in more structured code.

More students gave explanations of their design considerations in numbers (*evidence based design*). This was mainly done for selecting the right electronic components. The skill of *identifying behaviour and dynamics* scored less. This can be attributed to a question about timers which gave a very favourable outcome in Q1, but which was removed in Q4. Note that it is not the only assignment contributing to the competence *descriptive and mathematical modelling*; so there is no problem that not all students score on all skills. The skill of *using a predictive model* is not scored at all, because it falls outside the scope of this assignment.

7 Conclusion

In a project driven environment with a competency based learning approach, students perceive mathematical modelling differently. A model was proposed where conventional education is seen as a process from mathematics to design, while competency driven approaches tend to have an inverted sequence. We assumed there is a virtual barrier for on-demand learning when touching the mathematical modelling layer under the layer of technical skills. Several successful attempts were done in the past to remove the technology skill from the chain in order to make the opportunities of modelling visible.

We evaluated a learning activity that was changed in favour of this model. A simple reformulation of the design exercise towards an end product made students think from the user perspective. This was done by teaching them to talk about their design (e.g., with state diagrams) in order not to be confined by programming or electronics skills. Overall, there was an improvement in the ability to analyse the problem.

References

Alers, S. H. M., & Hu, J. (2009). AdMoVeo: A robotic platform for teaching creative programming to designers: Learning by playing. In M. Chang, R. Kuo, Kinshuk, et al. (Eds.), *Game-based education system design and development* (Lecture Notes in Computer Science Series, 5670/2009, pp. 410–421). Heidelberg: Springer.

Birenbaum, M. (2003). New insights into learning and teaching and their implications for assessment. In M. Segers, F. Dochy, & E. Cascallar (Eds.), *Optimising new modes of assessment: In search of qualities and standards*. Dordrecht: Kluwer.

Feijs, L. M. G., & Bartneck, C. (2009). Teaching geometrical principles to design students. *Digital Culture & Education, 1*(2), 104–115.

Hu, J., Ross, P. R., Feijs, L. M. G., & Qian, Y. (2007, June 11–13). UML in action: Integrating formal methods in industrial design education. In *Proceedings of the second international conference on technologies for E-learning and digital entertainment (Edutainment 2007)*, Hong Kong, China (Lecture Notes in Computer Science, 4469, pp. 489–498). Berlin: Springer.

Hummels, C. C. M., & Frens, J. W. (2009, April 4–9). The reflective transformative design process. In *CHI 2009 – Digital life, new world: Conference proceedings and extended abstracts; the 27th annual CHI conference on human factors in computing systems*, Boston (pp. 2655–2658). New York: ACM Press.

Pereira, A., Oliveira, I., Tinoca, L., Amante, L., Relvas, M. J., Pinto, M. C., & Moreira, D. (2009). Evaluating continuous assessment quality in competence-based education online: The case of the e-folio. *European Journal of Open, Distance and E-learning. EDEN*. http://www.eurodl.org/index.php?article=373. Accessed 25 July 2012.

Segers, M., Dochy, F., & Cascallar, E. (2003). The era of assessment engineering: Changing perspectives on teaching and learning and the role of new modes of assessment. In M. Segers, F. Dochy, & E. Cascallar (Eds.), *Optimising new modes of assessment: In search of qualities and standards*. Dordrecht: Kluwer.

van der Vlist, B. J. J., van de Westelaken, R., Bartneck, C., Hu, J., Ahn, R. M. C., Barakova, E. I., Delbressine, F. L. M., & Feijs, L. M. G. (2008, June 25–27). Teaching machine learning to design students. In Z. Pan, X. Zhang, & A. El Rhalibi (Eds.), *Technologies for E-learning and digital entertainment: Proceedings of third international conference, edutainment 2008*, Nanjing (pp. 206–217). Berlin: Springer.

Vygotsky, L. (1978). *Mind in society*. London: Harvard University Press.