The Role of Mathematics and Modeling in a Competency Centered Learning system

Geert Langereis, **Jun Hu and Loe Feijs**

Eindhoven University of Technology, Eindhoven, The Netherlands

g.r.langereis@tue.nl

Abstract With competency based learning in a project driven environment, we are facing a different perspective of how students perceive mathematical modelling. In this paper, a model is proposed where conventional education is seen as a process from math 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. 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.

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 (Hummels, Frens, 2009). The associated reflective transformative design process is based on learning and developing “from doing” and “by doing”, and is as such highly dependent on creation processes. This is where it distinguishes fundamentally from classical design engineering approaches. In classical engineering education, a foundation of mathematical tools and thinking is educated first. Secondly, the scientific frame is set, 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 in fact 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 notified by our students as a next investment in their competency development.

This paper will explain the competency based learning system for design education first. Next we will discuss what aspects of modelling are required for academic design students. In a subsequent section, some examples of assignments aiming at transferring modelling skills are explained. In a section about balancing the design engineering and modelling, a vision on 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 effectiveness and outcome. The theory and proposal is evaluated based on our experience with a specific assignment on microcontrollers. The application field of design engineering, and the implementation of modelling skills in favour of design engineering education, is not static. We are continuously debating on the processes, needs and skills required to realise our interactive products. The evaluation in this paper is part of that open discussion.

Competency Based Learning as Optimised for Design Education

The Eindhoven University of Technology (TU/e) is one of the three Technical Universities in the Netherlands and has now nine departments. In 2009 the TUE had around 3100 employees, 126 professors, 7100 students, 190 PDEng students, and 640 PhD students.

Industrial Design (ID) is one of the nine departments. 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 Environment (also known as Ambient Intelligence), as well Entertainment Computing research. Special attention is given to methodological and theoretical issues. 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 ID department brings together expertise in the fields of speech interfaces and multi-modal interfaces, robotics, signal processing for perceptive user interfaces, aware environments, applications of agent technology in user interfaces, and user modelling. The department of ID was established in 2001, and currently includes 205 staff, 36 of whom are permanent full time researchers, the remaining are part-time lecturers, administrative personnel, post-docs and Ph.D. students.

The focal area of the department of Industrial Design, being interactive products, has resulted into an educational system that differs significantly from the other departments. For designing intelligent/interactive 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 “learning by doing”. This means that education starts with practical work before students discover which technology is involved. Competences are offered on-demand as an answer to the self reflection gained at executing projects. This learning model applies particularly well for industrial designers. It is not knowledge driven because the industrial designer has to develop contexts of use, actively explore concepts, evaluate alternative solutions, and bring new products to the world.

To assess the quality of the students, and to monitor their progress, ten competency areas are defined. The areas are:

- ideas and concepts
- integrating technology
- user focus and perspective
- socio-cultural awareness
- designing business processes
- form and senses
- teamwork and communication
- design and research processes
- self-directed and continuous learning
- 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 ten important skills of an industrial designer. 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 (three days per week), and on-demand taught in classes (two days a week). None of the classes is mandatory, and basic math is not part of the education programme.

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 is normally seen in one out of four embodiments. These embodiments of modelling, or skills of the designer, are a combination of mathematical skills and system insights.

The first skill is where models are used for *analysing a complex problem* by breaking it into pieces. In this case, the model can be state diagram or a flow chart, and does not necessarily have to be finalised into a mathematical model. In fact, the structure of the model is the first attempt to give the problem a shape. 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. An example is the use of state diagrams before software is written. Based on the state diagram, a designer can evaluate options and process flow using an abstract graphical representation.

A second modelling skill for design is to *identify behaviour and dynamics* of systems. To put this in practice, the required skills are closer to mathematics. The notion of feedback systems, second order dynamic systems and for example phenomena like friction, are typical engineering skills, almost close to craftsmanship, resulting into a predictive design. A designer can prevent oscillations in a system by identifying mass-spring systems, for example.

The *predictive power* of models becomes the strongest when a numerical mathematical model is implemented. The model can consist of closed form equations, or of a simulation environment. Closed form equations normally can be solved to find design criteria or to exclude options. Simulation can be done in several ways. We distinguish finite element models, as known for mechanics, empirical models, and analytical models. The analytical models create a direct link between observed phenomena and system hardware and are based on understanding. Empirical models require explorative knowledge about the system and can only be used for interpolation: phenomena outside the explored range may fall outside the valid range of the model. The finite element models are based on understanding on a micro level, but are used to estimate behaviour on a macroscopic level. It depends strongly on the skills of the engineer/designer to derive explicit characteristics on a macroscopic level. However, they are useful to explore systems on a computer (in a simulated domain) before building them in the real domain. In the simulated domain one can speed up time and evoke events, which is normally not possible in the real world.

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 the Industrial Design department of the Eindhoven University of Technology. It is worth noting, that in practice the competency "descriptive and mathematical modelling" is closely related to the competency "integrating technology". The reason is that the four skills mentioned above are the most directly applicable for realizing concepts in hardware. Although developing technology is not the core business of the department, hardware is needed as a substrate to explore in-tangible concepts.

Some Implementations of the Modelling Competency

In a previous paper (Hu et. al, 2007), a teaching method was presented to learn 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) 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 to communicate designs to an audience. The learning activity is an example of the first modelling skill “analysing complexity” 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 fact, the technology (or “science” or “software”) was completely removed from the exploration flow.

In another paper (Vlist et. al, 2008), a method to teach the abstract concept of “machine learning” to students was explained. Machine learning is about algorithms where technology improves behaviour by trial and error. This is extremely important for those who will have to design intelligent products. 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 the second one “identify behaviour and dynamics”. For 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 equations 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: 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, Hu and Alers 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 in a state where they may discover that modelling skills “analysing complexity” by means of state diagrams and “identify behaviour and dynamics” have become within reach.

The “predictive power” skill of mathematical modelling is amongst others implemented 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 for example an assignment on the basics of electronics. Industrial Design students ask for an assignment in basic electronics because this is the most accessible substrate to make products interactive. Electronics, especially analogue electronics, is seen as the toolset to give concepts “eyes and ears” by means of sensors and actuators. In another assignment, to be discussed later, microcontrollers are introduced to create the versatile brain of the interactive concept. Given this view on analogue electronics, 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. Here we are facing the problem that electronics is experienced as new and difficult, and in that confused state we have to convince the students they will need calculations to prove their choices are correct. Removing or substituting the hardware in our educational approach by an equivalent system cannot solve this: we need the electronics. Our approach consists of three elements empowering the students with hands-on electronics and give them the cognitive space to see the importance of the calculations. First, we have created a low-threshold electronics atelier. The electronics assignment includes a guided workshop in the atelier after which students are found there on a regular base. Assistants are always available in the atelier for solving 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 atelier (shift registers, a limited set of sensors, etc.) and are well documented on the intranet. This approach appeared to be successful.

The Balance Between Designing Engineering and Modelling

Traditional engineering education and strategies work from theory towards practice. This means that, roughly speaking, students are learning math in the first year, science in the 2nd and 3rd, before they can do the practical applied work in the end or during the master phase. In Fig. 1 this is represented as a three-stage approach from left to right. The thought behind this is that a mathematical foundation is the base for understanding science, which in its turn is needed to create new things. In the representation of Fig. 1, the block “science” is used in a broad sense: it does include raw engineering of 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.

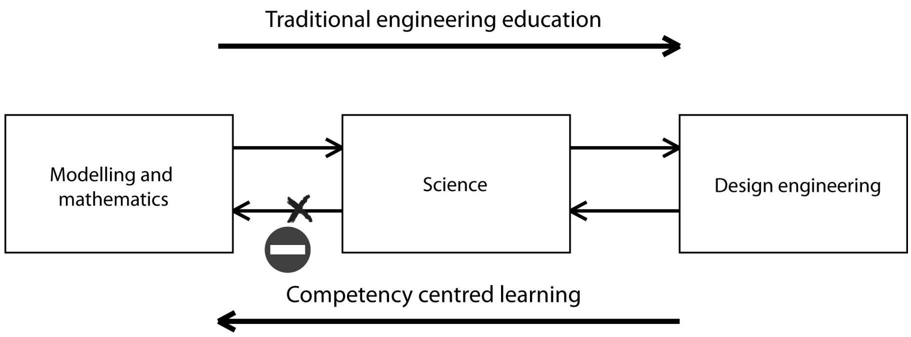

Fig. 1 One-way learning direction in traditional education and the opposite direction in competency centred learning

As already explained in the introduction, the choice of the Industrial Design department in Eindhoven to work on interactive systems, products and services, resulted into the implementation of a competency centred curriculum. This is done by starting on the right of Fig. 2 with practical realisations; scientific and engineering backgrounds are 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 putting 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 stop when going from science to the natural need for math. We experience this in their way of working and we can only guess about 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. 1 that the modelling and mathematics competency has been placed next to science as shown in Fig. 2.

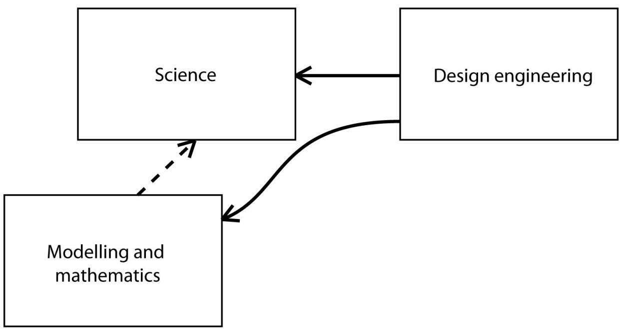

Fig. 2 Learning strategy for stimulating the mathematical modelling skill optimised for competency-based learning

What we are hoping for 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.

Education Example: the Microcontroller Assignment

An assignment in the core of designing interactive systems is about deploying microcontrollers for realizing prototypes. At the department for Industrial Design, "assignments" are a pre-described learning activity offered to bachelor students. Assignments consist of six weekly lessons of two hours, plus 36 hours of self-study, preferably including a significant amount of practical work. In total 48 hours spread over one quartile. In the assignment "Introducing Microcontrollers" the self-study mainly consists of a design case.

This assignment, as offered to Industrial Design students in the bachelor phase, 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 quartile (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). The assignment was repeated in the fourth quartile (Q4). In fact, this is an opportunity to verify the difference between the original model of Fig. 1 and the alternative learning strategy model of Fig. 2.

In fact, both the assignment in Q1 and Q4 were 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.

Microcontroller design case 1&2

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 push.

Color memory game: Player 1 mixes two or three LED colours (with potmeters) into one RGB colour, presses a button and puts the potmeters in a random position. Player 2 has to memorise the colour and has to reproduce it. The microcontroller determines whether you are close enough.

What was changed was:

- We motivated them to make a package to focus on the user experience, rather than to see the code plus circuit as the end result. This was accompanied by a lecture with examples of good-looking functional casings. It was assumed this approach inverted the technology push into a design driven technology pull.
- One part of the lectures was replaced by a new part about how to communicate about a microcontroller system. This was said to be needed for debugging, to find effective help from the atelier assistants, and to structure the problem before solving it. In that explanation there was an introduction of state diagrams to transfer concepts into programmable solutions. We assumed this to be a method to create the direct link from design needs towards modelling, while bypassing the hardware state (curved arrow in Fig. 2)
- In Q1 the 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 getting familiar with the microcontroller: connecting it, writing subroutines, playing with communication between computer and the microcontroller board.

To compare Q1 to Q4, we tested the end reports on 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. The criteria are summarized in Table 1.

Table 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

In Q1 there were 10 groups, in Q4 9. All are bachelor students. The scores are collected in Fig 3. It can be seen that we increased the number of groups creating a packaged functional game 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 into more structured code.

More students gave explanations of their design considerations in numbers (evidence based design). This was mainly done for picking the right electronic components. The skill of "identifying behaviour and dynamics" was scored less. This can be attributed to a question about timers which gave a very profitable 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.

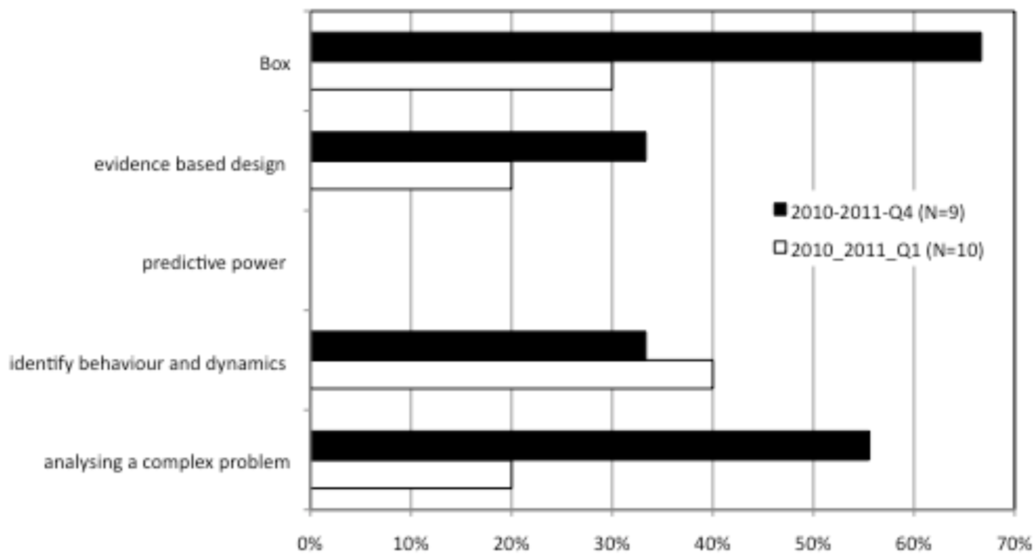

Fig. 3 Scores on four mathematical modelling skills before and after changing the content of the assignment "Introducing Microcontrollers"

Conclusion

We have a different perspective of how students perceive mathematical modelling in a project driven environment with a competency based learning approach. It all depends on how the material is offered up to what level students absorb the theory. A model was proposed where conventional education is seen as a process from math 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 and helped them to find the patterns behind the problem first, before translating them into technology. This was done by learning them to talk about their design (for example with state diagrams) in order not to be confined by programming or electronics skills. In the end we could see an improvement in the ability to analyse the problem.

References

Hummels, C.C.M. & Frens, J.W. (2009). CHI 2009 - digital life, new world: conference proceedings and extended abstracts ; the 27th Annual CHI Conference on Human Factors in Computing Systems, April 4 - 9, 2009 in Boston, USA. The reflective transformative design process, (pp. 2655-2658). New York: ACM Press.

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

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

Alers, S.H.M. & Hu, J. (2009). Learning by playing. Game-based education system design and development. AdMoVeo: a robotic platform for teaching creative programming to designers, (pp. 410-421). Springer Berlin/Heidelberg.

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

Hu, J. & Alers, S.H.M. (2009). Proceedings of Design and Semantics of Form and Movement (DeSForM) 2009. AdMoVeo: an educational robotic platform for learning behavior programming, (pp. 218-219). Eindhoven: Philips Design.