A REVIEW OF COMPUTER SKILLS IN INDUSTRIAL DESIGN EDUCATION: ISSUES, OPPORTUNITIES, AND RECOMMENDATIONS

GIOVANNI JESUE CONTRERAS GARCIA

PhD

The Hong Kong Polytechnic University

2018

The Hong Kong Polytechnic University

School of Design

A REVIEW OF COMPUTER SKILLS IN INDUSTRIAL DESIGN EDUCATION: Issues, opportunities and recommendations

Giovanni Jesue CONTRERAS GARCIA

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

September 2017

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

_______________________________________(Signed)

Giovanni Jesue CONTRERAS GARCIA
_______________________________________(Name of student)

Abstract

Industrial Design is a discipline concerned with determining the form and function of mass-produced goods. This process depends extensively on the use of models. Until the advent of personal computers, modelling was done mostly by drawing on paper in the case of two-dimensional models, and by making physical mockups using clay, wood, foam, etc in the case of three-dimensional models. With the advent of personal computers however, this process became increasingly computer-based. This application of comptuers for modelling and visualization, constitutes the foundation of what is generally known as 'Computer Aided Design' (CAD) in the Design and Engineering disciplines, and which has become a most essential skill for any practicing professional.

The development of these computer skills has been studied in the past, most of this research however, has been done in disciplines with a longer research tradition such as Architecture or Engineering. Moreover, most of this research has been approached from the perspective of Computer Aided Design, not from the perspective of computer skills in general. Therefore it is unknown to what extent Industrial Design students acquire these skills as they transition through college.

Besides the fact that computer modelling has become essential in Industrial Design, studying the development of computer skills is important for other reasons. There are computer skills not related to Computer Aided Design, which have been identified as basic literacies in the 21st century. These other skills are intricatelly related to the development of advanced computer modelling techniques which pose opportunities and challenges for Industrial Design education.

These studies investigate the computer skills being fostered in Industrial Design schools, and how their development is translated into the curriculum. The studies are based on a survey of 38 Industrial/Product Design schools from different parts of the world, through which their corresponding curriculums were studied/analized. This information was complemented with a number of interviews with practitioners, academics and researchers, and other sources such as artefacts analysis, action research and auto-ethnography.

Among other, the studies found that the development of comptuer skills in Industrial Design schools focuses almost exclusively on developing 3D modelling skills, often even excluding 2D skills. The development of other skills, such supporting project management with the use of computers, or computer programming is almost null. The studies also found that most schools do not teach students to work with Polygonal Models, and that a good number only teach students to work with either Solid or Surface (NURBs) models, but not both. One of the arguments raised in the studies, is that due a number of trends, students should learn to work with all three different types of models, including polygonal ones.

The studies also found that while strictly speaking the term CAD refers to the use of computers to support the design process, the general understanding in the field, is of CAD being essentially about modelling, and even just about 3D modelling. This tints the research and discussion around the education of computer skills in the field.

The studies finish by providing a series of recommendations to enhance the development of computer skills in Industrial Design Education, by using a framework developed in these studies.

Publications

Contreras, G. J., & Siu, K. W. M. (2015). Computer programming for all: A case-study in product design education. *Procedia-Social and Behavioral Sciences, 182*, 388-394.

Rivas, M., & Contreras, G. J. (2016). The Neurolecturer as Model for Design Education: Fostering Creativity and. *Design Education for Fostering Creativity and Innovation in China* (pp. 212-226). IGI Global.

Siu, K. W. M., & Contreras, G. J. (2015). Personal Interaction and Informal Learning: The Case of China. In *Measuring and Analyzing Informal Learning in the Digital Age* (pp. 140-152). IGI Global.

Siu, K. W. M., & Contreras, G. J. (Eds.). (2016). Design Education for Fostering Creativity and Innovation in China. IGI Global.

Siu, K. W. M., & Contreras, G. J. (2016). 20 Years of Doctoral Research Studies in Design in Hong Kong. *Design Education for Fostering Creativity and Innovation in China* (pp. 227-244). IGI Global.

Siu, K. W. M., & Contreras, G. J. (2017). Disruptive Technologies and Education: Is There Any Disruption After All?. In *Educational Leadership and Administration: Concepts, Methodologies, Tools, and Applications* (pp. 757-778). IGI Global.

To him, who has given me everything, and to whom everything belongs. May these studies and everything that from they may derive, serve his purposes. Amen.

To my family, proof that God exists.

Acknowledgments

I would like to start by giving thanks to my supervisor, Prof. Kin Wai Michael Siu, without whose support and vote of trust these studies would have not been possible. During the length of these studies Michael constantly pushed me to attain higher goals and his role has been instrumental in my academic career.

I would also like to thank all members of the Board of Examiners: Professor Tim Jachna of our School of Design at PolyU, for serving as the chair, and Professors Michael Tovey (Coventry University) and Kong Bien Chua (Hong Kong CitiU), first of all for accepting the invitation to serve as external examiners, and for all their feedback and suggestions, which proved most useful in preparing for the Viva.

My gratitude goes as well to all the interviewees and respondents of the survey at the different academic institutions, for accepting the invitation to participate, and taking the time to provide their insight, essential for the completion of these studies.

To all my former PhD fellows and friends at The School of Design: Fawaz Bakhotma, Vivian Li, Cynthia Xiao, Albee Chen, Marcus Wernli, Satyakam Sharma, Oluwole Soyinka, Elaine Wong, Paul Lo, VJ Wang, Betty Durango, Hara Chon, PeiYao Cheng, Kimberly Sheen, Karen Yang, Julia Naskova, YiHua Wong and the rest. Thanks for making the times at school a bit more lively and extending a helping hand when needed. The same goes to the fellows and friends in other departments: Heriberto Saldivar, Rafal Radecki, Ridvan Aydin, and Aleksandra Ziaja.

I also cannot omit the secretaries at The School of Design and The Research Offices; Florence Lam, Emily Leung, Winnie Li and Irene Ho, for their help with the numerous, administrative procedures and form-filling that had to be completed along these PhD studies, from the times I was applying, to the very end.

I also want to thank my current colleagues at the University of Sussex; Prof. Ann Light, Prof. Ben Du Boulay, Kate Howland, Luis Ponce Cuspinera and Pollie Barden, for taking the time to meet me and offering their support and feedback. Special thanks go to my conational Grecia Garcia Garcia for taking the time to read chapters one and seven entirely and offering extensive feedback.

Finally I would like to thank Ray and Shirleene Kiew for kindly offering their home during my trip to Hong Kong for the Viva. As a former College teacher once said: any 'personal' achievement, is hardly truly individual. In reality all achievements are directly or indirectly underpinned by multiple individuals in our present and past. To all those individuals not mentioned earlier who in one way or another have supported me to get to this point, I want to express my gratitude.

These studies would have not been possible without the support of the Research Grants Council of Hong Kong, and without the schollarship provided by The Hong Kong Polytechnic University. These have given me an opportunity that few have, thanks for making the dream come true.

Table of Contents

ABSTRACT III

PUBLICATIONS V

ACKNOWLEDGMENTS VII

TABLE OF CONTENTS IX

1 INTRODUCTION 1

1.1 INDUSTRIAL DESIGN 1

1.1.1 Industrial Design Process 3

1.1.2 Models in ndustrial Design 3

1.1.3 The Computerization of Models 9

1.1.4 Computers in Industrial Design 10

1.2 COMPUTER SKILLS IN NDUSTRIAL DESIGN EDUCATION 11

1.3 PROBLEM STATEMENT 13

1.4 AIM 15

1.4.1 Research question(s) 15

1.5.2 Objectives 16

1.5 JUSTIFICATIONM 16

1.5.1 A Shifting Field? 20

1.6 METHODS 23

1.7 CONTRIBUTION 24

1.8 OUTLINE OF THE STUDY 25

1.9 CHAPTER SUMMARY 27

2 LITERATURE REVIEW 31

2.1 INDUSTRIAL DESIGN PROCESS 31

2.1.1 Basic Design Process Model 32

2.1.2 Other Variations 37

2.1.3 Concurrent Engineering 40

2.2 COMPUTER MODELLING 43

2.2.1 2D models 44

2.2.2 3D models 46

2.3 INDUSTRIAL DESIGN EDUCATION 55

2.4 COMPUTER MODELLING IN INDUSTRIAL DESIGN EDUCATION 55

2.4.1 Delivery 57

2.4.2 Integration 58

2.4.3 Advantages 59

2.4.4 Insufficiency 59

2.4.5 Lack of Foundational Knowledge 62

2.4.6 *Tool Domestication* ... *62*
2.4.7 *Disregard/Scepticism of computer skills/technology* ... *63*
2.5 Chapter summary ... 69

3 METHODS ... 75

3.1 Aims ... 75
3.2 Paradigm ... 76
3.3 Methods ... 79
3.3.1 *Approach* ... *80*
3.3.2 *Data Collection* ... *86*
3.3.3 *Analysis* ... *93*
3.4 Aims ... 95

4 CONTEXT ... 99

4.1 The Rise of Computer Programming ... 99
4.1.1 *Computational Thinking* ... *100*
4.1.2 *Computer programming* ... *101*
4.2 The Rise of Computer-mediated Instruction ... 103
4.3 The Rise of Generative Design ... 107
4.3.1 *Generative design* ... *109*
4.4 the Rise of Additive Manufacturing ... 119
4.5 Chapter Summary ... 120

5 TRADITIONAL COMPUTER SKILLS ... 127

5.1 Objectives ... 127
5.2 Approach ... 136
5.2.1 *Integration* ... *136*
5.2.2 *Starting point* ... *138*
5.2.3 *Order/sequence/progression* ... *142*
5.2.4 *Intensity/Depth/Scope* ... *145*
5.2.5 *Content* ... *151*
5.3 Teaching Methods/Strategies ... 159
5.4 Use of computer-mediated education ... 160
5.5 Chapter Summary ... 166

6 EXTENDED/NEW COMPUTER SKILLS ... 173

6.1 Objectives ... 173
6.1.1 *Computational Thinking* ... *178*
6.2 Approach ... 180

6.3 Generative Design ... 185
6.3.1 *Tools* ... *186*
6.3.2 *Teaching Methods/Strategies* ... *191*
6.3.3 *Symbiotic relationship* ... *197*
6.4 Chapter Summary ... 199

7 CONCLUSION ... 207

7.1 Regarding the Range, Extent and Overall Curricular Approach ... 208
7.2 Regarding Addressing Trends ... 210
7.3 Recommendations ... 211
7.3.2 *Traditional computer skills* ... *213*
7.3.3 *Integration Approach* ... *216*
7.3.5 *Computer Programming* ... *217*
7.4 The PRODISST Framework ... 219
7.5 Limitations ... 222
7.6 Contribution and Future Work ... 223
7.6.1 *What next* ... *224*
7.6.2 *Shifting Field or Not* ... *225*
7.7 Chapter Summary/Final Remarks ... 226

APPENDIX I (LIST INSTITUTIONS SURVEYED) ... 229

APPENDIX II (PROFILE OF PARTICIPANTS AND THEIR INSTITUTIONS) ... 231

APPENDIX III (LIST OF INDIVIDUALS/EXPERTS CONSULTED) ... 239

APPENDIX IV (PILOT STUDY) ... 247

GLOSSARY ... 251

REFERENCES: ... 275

Chapter One

1 INTRODUCTION

1.1 Industrial design

Industrial design is an activity concerned with determining the formal characteristics of mass-produced goods. Tomas Maldonado, of the influential Ulm School of Design, defined industrial design as "…*a creative activity whose aims are to determine the formal qualities of objects produced by industry."* These qualities are not just the external characteristics, but also the structural and functional relationships that transform a system into a coherent unity from the viewpoint of both, producers and users (Maldonado, 1977). The Industrial Designers Society of America on the other hand, define it as *"the professional practice of designing products used by millions of people around the world every day."* And adds: "*Industrial designers not only focus on the appearance of a product, but also on how it functions, is manufactured and ultimately the value and experience it provides for users."* (IDSA, 2018). Ulrich and Pearson on their part, define Industrial Design as an activity that transforms a set of product requirements into a configuration of materials, elements and components that comprise an artefact (Ulrich & Pearson, 1998), and Earle (1994), defines Product Design as: *"the creation, testing, and manufacture of an item that usually will be mass produced, such as an appliance, a tool, or a toy"* (Earle, 1994 p.16).

Industrial designers are concerned with developing the form and way in which users interact with products. Industrial designers design numerous things used in everyday life; from cars to power tools, from medical equipment to mobile phones. They consider how to make products *"easier to use, more efficient, cheaper to produce or better looking.* (Porter & James, 1999; Unver, 2006, p. 323). Cross (2007), says that the most essential job of a designer, is to *"provide for those who will make a new artefact, a description of what that artefact should be like."* Adding that; "*The focus of all design activity is that end-point"* (Cross, 2007, p. 33).

The birth of Industrial Design as a discipline, is directly related to the advent of industrialization and the rise of the consumer society (Valtonen, 2007, p. 14). This particular approach to designing consumer products, is rooted in the philosophy and practice of the Arts and Crafts, the school of the Bauhaus, and the streamlining movements of the 20th century. It is called 'industrial' because it is concerned with the design of products that are manufactured by industrial processes (Cross, 2007; Unver, 2006, p. 323).

In the modern enterprise, industrial design is often part of other research and development (R&D) operations, and it is characterized by constant engagement with other departments. This is why it is said that industrial designers are interface and catalysts between departments such as marketing and engineering (Rusten & Bryson, 2009; Valtonen, 2007, p. 16). The incorporation of industrial designers in multi-disciplinary teams in charge of New Product Development however, often meant that they had no control of the final stages of the design process. These later stages were carried on by personnel in charge of Design for Manufacture (Bryden, 2014, p. 11; Desbarats, 2010).

Note on terminology: In most literature the terms product design and industrial design are used as synonyms (DiTullo, 2014). Some scholars argue that each term is used in different parts of the world, some other point to slight differences in focus, and yet some other argue that the term product design is simply a later development with no substantial differences between one and the other (Unver, 2006, p. 323). While for the sake of consistency, an effort has been made to stick to the term industrial design throughout this thesis, both terms are treated as synonyms.

1.1.1 Industrial Design Process

The Industrial Design process is an iterative process that generally starts by identifying a problem or 'area of opportunity' to design a new product. Next, research is done to determine the requirements of the different stakeholders–manufacturer, retailer, user, etc., and then translating the findings of that research into a list of design criteria. Afterwards, design ideas are generated and evaluated against the criteria previously generated. The best options are then selected, refined, and evaluated again. This process continues until an optimal design is found, or the time for the project is over.

The process implies carefully understanding requirements, coming up with ideas for solutions, and implementing them in the redesign of a product, or the creation of an entirely new one. Bryden (2014) describes this process as: *"…an iterative process involving analysis, thinking, conceptualizing, visualizing, model making, prototyping, testing and refining"* (Bryden, 2014, p. 7). This process requires understanding and often re-interpreting a brief, doing research, making sketches, detailed illustrations, CAD models, physical models and prototypes to test the design.

1.1.2 Models in Industrial Design

Being a discipline concerned with envisioning physical products yet to be made, the Industrial Design process relies extensively in the use of models for visualization purposes (Cuffaro, 2006; Efer, 2017). Porter & James (1999), recall that two essential functions of the industrial designer are: to visualise the product concept, and to represent alternative design solutions. Walter Rodriguez on the other hand, acknowledges that visualization *"is the single most important ability required to become an excellent designer"* (Rodriguez, 1992, p. xi). Similarly, Kalpakjian et al. (2014), say that Product Design requires of *"analytical and physical models of the product, for the purposes of visualization and engineering analysis"* (Kalpakjian et al. 2014 p.9). Unver (2006) on his part, recalls that models are the *"essential evidence, which has underpinned both the design process and the educational dialogue"* (Unver, 2006, p. 324).

Often the literature cites Donald Schon's work on 'The Reflective Practitioner' to illustrate how models underpin designer's though process–what Schon refers to as: 'Reflection in Action'–and through which designers evolve their ideas. For example, Oxman (2006) recalls, that Design Education is based on a model predominantly characterized by Schön's notion of 'visual reasoning.' In this model the designer 'establishes a dialogue' with the materials of the problem. It is also very much in line with Gardner's 'theory of multiple intelligences,' which suggests that creative professionals like designers think visually (Oxman, 2006). More recently, Luscombe (2018), cites the 'Theory of Extended Mind,' which suggests that actions are performed not just to achieve a goal, but also to *"help work things out"* (Luscombe, 2018 p.10).

Far from being just a means for communication–notwithstanding that perhaps models are the best tool for communicating attributes of a product design–models are thinking tools for designers. Ashby & Johnson (2013) for example, say that sketching is *"a kind of image-based discussion with oneself or with others" (p.*43). Cross (2007) as well, acknowledges that the thinking process of the designer, seems to hinge around the relationship between internal mental processes and their external representation. This 'conversation' between what designers visualize/imagine internally, and the external representations they make, shows how Industrial Design is a reflective practice.

If it is through models that designers explore ideas, solve problems and communicate with others, designers need a medium upon which 'half-formed' ideas can be expressed–modelled–and then reflected upon. The most basic of these mediums is the paper drawing, reason why it is undoubtedly the most common, but not the only one. Depending on when in the process they are used, these representations can be 2D drawings, physical models, or computer models (Cross, 2007; Porter & James, 1999).

As defined by Grix (2010), a model is: *"an abstraction from reality that serves the purpose of ordering and simplifying our view of reality while still representing its essential characteristics"* (Grix, 2010, p. 21). Similarly, Root-Bernstein (2001) write that all models are *"distillations of the elements considered to be the most critical determinants of structure and function."* And have the objective of making accessible something that is difficult to experience easily (Root-Bernstein, 2001, p.229). And Ye et al. (2004) say that, *"Engineers design their models in the computer as a mock-up of the real-world physical models"* (Ye, et al., 2004).

Root-Bernstein & Root-Bernstein (2001), typify models in four groups: 1. Representational or physical models, displaying the physical characteristics of an object; 2. Function*al* models, capturing the essential function or operation of an object or mechanism; 3. Theoretical models, capturing the fundamental concepts governing a process; and 4. Imaginary models, which display the features of something cannot be observes directly. Milton and Rodgers (2011), on the other hand, classify models used in Industrial Design in four main groups: sketch models, mock-ups, appearance models and test rigs (Milton & Rodgers, 2011 p.98).

McMahon and Browne (1998), recall that in the Design and Engineering disciplines, of all the properties of a design that can be modelled, form and structure are of particular importance, and the most appropriate method to represent–model–these has traditionally been graphical (McMahon & Browne, 1998 p.8), reason why the majority of models used in Industrial Design, are what Root-Bernstein & Root-Bernstein (2001), call 'representational.' While the literature often refers to different form of visual aids using different terms–particularly making a distinction between two-dimensional and three-dimensional representations, 'drawings' and 'models'–all are in fact some form of model.

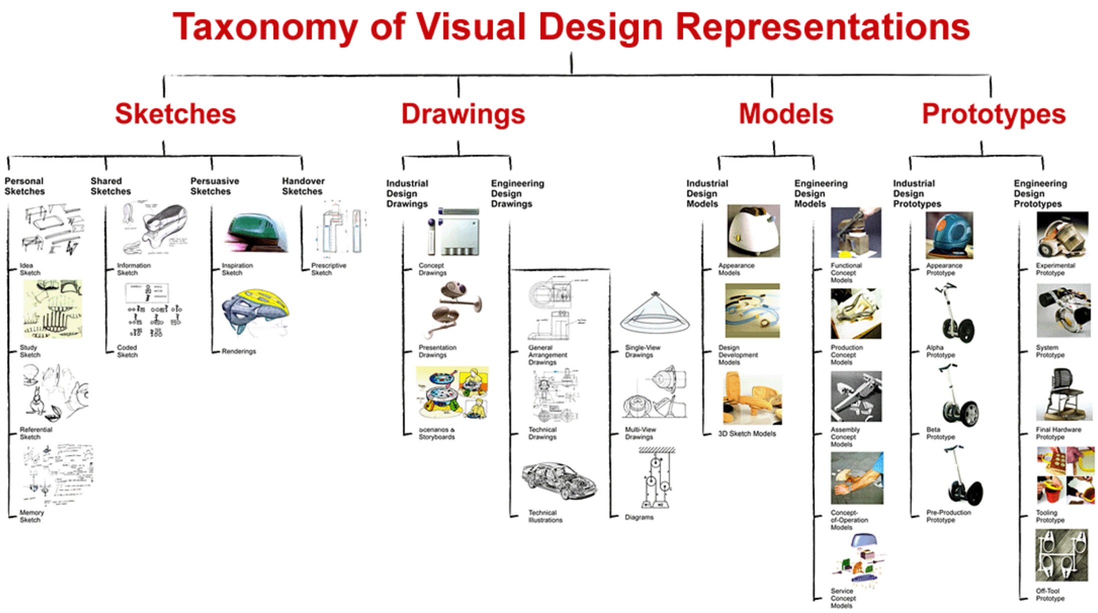

Figure 1.1 Different types of visual representations used in Industrial Design (Pei et al. 2011)

Pei et al. (2011) classify a number of different visual design representations used in Industrial Design practice, all of which can be classified as models (Fig. 1.1). The Industrial Designers Society of America as well, identifies 30 different types of sketches and models commonly used in industrial design practice (IDSA, 2018).

Different types of models are used at different stages of the design process for different purposes, for example: early models made of polyurethane foam or cardboard are used to review/test basic overall aspects of the product. As the process moves on, models are more detailed and representative of what the final design will look-like (Ashby & Johnson, 2013; Milton & Rodgers, 2011). Milton & Rodgers (2011), recall that models can be used to check the functionality, usability, ergonomics, proportion and form of concepts, test the structural integrity of a design, test the consumer reaction.

Figure 1.2 Initial foam and final appearance models of a glue gun by an industrial design student at the University of Cincinnati. (Photo by author(s) June 5th, 2006)

McMahon & Browne (1998) on the other hand, list some of the functions of design models as: record and manipulate ideas, provide a basis for the evaluation of the design, facilitate the communication of the design between participants (p.7). According to them, some of the models necessary during the New Product Development process include: models of the functional and other customer requirements, models of constraints on the design–imposed due available materials and processes, models of loads to which the design will be subject, models to evaluate the performance of the design–in the case of mechanical properties this can include stressor thermal analysis, or aerodynamic assessment (p.13).

As a communication tool, models can trigger evaluative and/or generative actions on the receiving end. In either case, these actions involve extracting information from the model, and combining it with new information in the realization of a new model (McMahon & Browne, 1998 p.10).

1.1.3 The Computerization of Models

Until the advent of personal computers, most modelling in Industrial Design was done by sketching–in the case of two-dimensional models–or, by using mediums like: polyurethane foam, Styrofoam, wood, clay or cardboard in the case of three-dimensional models. With widespread availability of affordable personal computers and software, it became possible to have the equivalent of both: two-dimensional and three-dimensional models in the computer. Given the numerous advantages that these computer models offered, the modelling process started to be increasingly dependent on computers. Nowadays, technical illustrations, 3D models, and more recently, hand sketches are extensively–and in some cases almost exclusively–done in the computer. This transformation has had a profound impact on the practice of the discipline over the past 3 decades, the most notable perhaps, is allowing industrial designers to achieve higher levels of efficiency and precision (Bryden, 2014; Dönmez, 2013; Desbarats, 2010; Kalpakjian et al. 2014 p.9; Luscombe, 2018; McMahon & Browne 1998; Unver, 2006; Schoonmaker, 2002; Stiny & Onur, 2012; Tovey, 2002; Tovey and Owen, 2000; Valtonen, 2007; Varinlioglu et al., 2015; Ye et al., 2004).

The ‘computerization’ of modelling brought with it the promises that generally accompany technology; being able to do more and better. In the case of CAD, this meant efficiency and precision. Unver (2006) for example, notices that with computers a single individual can do what would have required of a team of specialists in the past. Among other things for example, computer modelling allows the experimentation with colour and texture. On the other hand, advances in Rapid Prototyping, have made the process of obtaining high-quality physical models from CAD models quite easy (Ashby & Johnson, 2013 p.43; Bryden, 2014).

1.1.4 Computers in Industrial Design

It is this application of computers for the purposes of modelling and visualization that constitutes the foundation of what is commonly known as 'Computer Aided Design,' in the Design and Engineering disciplines (McMahon & Browne, 1998 p.1; Musta'amal et al., 2012), and which has become a most essential skill for industrial designers (Bryden, 2014; Dönmez, 2013; Desbarats, 2010; Kalpakjian et al. 2014 p.9; Luscombe, 2018; McMahon & Browne 1998; Unver, 2006; Qi et al., 2010; Schoonmaker, 2002; Stiny & Onur, 2012; Tovey, 2002; Tovey and Owen, 2000; Valtonen, 2007; Varinlioglu et al., 2015; Ye et al., 2004).

In its simplest and broadest sense, CAD can be defined as "the use of a computer in the design process" (Bilalis, 2000). Strictly speaking then, CAD is applied in many different fields, such as chemistry, bio-engineering, or even nutrition. The term CAD however, has been more prominently used in disciplines which rely on the use of visual aids–models–such as Engineering, Design–including all different design disciplines: Graphic, Industrial, Interior, Fashion, etc.–and Architecture. McMahon and Browne (1998) for example, claim that the aim of CAD is to *"apply computers to both the modelling and communication of designs*" (p13).

Some of the advantages of computer modelling include: preventing errors and misinterpretations, being able to evaluate and test design better, improved efficiency and achieve shorter development timescales, producing more accurate models, creating realistic images/visual representations, facilitating the communication/collaboration–Concurrent Engineering–with other departments, reducing development costs, facilitating Design for Manufacture, facilitating design documentation–Paperless Design, automating the creation of technical drawings, and improving the ability to maintain design intent along the design process (Bryden, 2014; Cuffaro, 2006; Desbarats, 2010; Kalpakjian et al. 2014 p.9; McMahon & Browne 1998; Unver, 2006; Schoonmaker, 2002; Tovey, 2002; Tovey and Owen, 2000).

Moreover, 3D CAD and Rapid Prototyping, have also given back industrial designers control over the configuration of the final product (Bryden 2014; Desbarats, 2010). Cuffaro (2006) for example, acknowledges that the 'conversion' from drawing to a three-dimensional product, for many years occurred at the tooling stage, which meant, that the person in charge of making the production molds had 'the last word' about how the final product would actually end up looking-like (Cuffaro, 2006 p.200).

1.2 Computer Skills in Industrial Design Education

Industrial design education has been traditionally defined by craft and problem solving, with an increasing concern for business more recently. It is an education based on study courses, hands-on product development, and a master-mentor relationship derived from the school of the Bauhaus (Cross, 2007; Findelli, 2001; Kolko, 2005; Valtonen, 2007). Industrial designers are typically educated in university programmes where they learn to make a variety of models ranging from hand drawings to quick physical mock-ups using paper, cardboard, foam and clay, they also learn to make detailed models using a variety of other materials.

Instruction of computer skills in industrial design schools has traditionally focused on training students to use software to create models, and to use these models to communicate their designs—such as with 3D renderings— (Dönmez, 2013; Hanna, 2004; Valtonen p.15). The United States Department of Labour for example; says on its occupational handbook website, that most Industrial Design programs include training in drawing, computer-aided design and drafting (CADD), and three-dimensional modelling.

This instruction, which will be referred as 'traditional computer skills,' has delivered in exposing students to the advantages of CAD in terms of efficiency and precision, and even empowered students with weak spatial thinking and communication skills—i.e., marker renderings (Unver, 2006). However, a number of issues with it have been identified in the past. For example; the fact that it tends to focus on developing what Chester (*2008)* calls *'C*ommand *K*nowledge,' which is simply teaching students what each of the different tools in a CAD software can do, but not really how to apply those tools in a design project. Ye et al., (2004) for example; say that most study programs focus on teaching students skill, *"what buttons to push"* but they fall short on theory (p.1457).

It has also been argued that, sometimes this instruction is simply insufficient, failing to bring students all the way up to the point in which they can really model the things they want to model (Dankwort et al., 2004). Furthermore, in some cases it is also seen as having a negative effect. James Self (2012) for example, says that on the one hand, CAD attracts students with immediate feedback and 'glossy images,' while on the other it conditions them to work in a particular way.

1.3 Problem Statement

The adoption of computers in Industrial Design is not one without issues. The fact that computers have been adopted in Industrial Design practice for example, does not mean designers make effective use of them. Moreover, these issues have often percolated over education. The fact that the foundation for the use of computers in Industrial Design is for the purposes of modelling and visualization, means that other aspects of computer application and computer education have remained unexplored.

When discussing the way designers tend to use computers, for example, George Stin describes it as: "more of an archival kind of enterprise" in which the main aim is to provide representations to evaluate designs in order to track them throughout their lifetimes (Stiny & Onur, 2012, p. 8). Yet, it is unknown whether current instruction of computer skills in industrial design education prepares students to apply computers in different aspects of the design process.

In addition; Friedman (2014), recognizes that social systems like universities, are prone to accumulate detrimental cultural and behavioural patterns, While Oppenheimer (2003), warns that education systems are complex social institutions carrying a strong moment of inertia. Thus, it is plausible to think that education in general tends to remain stagnant.

There are reasons to believe that the approach to the instruction of computer skills when computers were introduced in design schools may not be appropriate now. While the introduction of computers in design schools has delivered in the promise of efficiency, precision and productivity, it could be argued that this 'computerization' has essentially been a 'domestication' of computers. In other words; the types of models, and the modelling processes have remained essentially unchanged, the only difference, is the means used to create them. For example: designers start by envisioning a 3D form–usually through a process of exploration through sketches–then, the designer builds a 3D model of that form by 'digitally hand-crafting it,' that is, by building it as if working with a physical model, but in the computer. In fact, several of the modelling tools in the computer are called the same as the tools used when working with actual physical models.

While the Instruction of computer skills has been researched in the past, and despite computers started to be widely introduced in industrial design schools more than 20 years ago, a lot is still unknown about the development of computer skills within Industrial Design Education (Bryden, 2014; Unver, 2006; Togay et al., 2016).

One of the reasons, is that much of the research done in the past, stems from other disciplines such as Engineering or Architecture. While there are commonalities across different disciplines, not all findings from past research are applicable to Industrial Design Education. For example, while in other disciplines with a longer research tradition like architecture, questions such as whether students should learn computer programming or not have been debated for long, in Industrial Design this discussion is only emerging.

Moreover, much of the existing research regarding the study of Computer Skills Development in Higher Education, has been done from the perspective of CAD, not from the perspective computer skills in general. In addition; a good portion of this existing research–in the specific context of Industrial Design Education–was done at the turn of the century, when computers were being introduced in design schools. Thus, often this research is outdated.

There are no studies for example, that show whether the issues found in the past have been addressed and/or how. Equally, there are no general studies that show what specific computer skills are being taught, how the development of these skills is approached through the curriculum, what are the learning objectives that schools have, etc. In consequence, it is unclear how the instruction of computer skills has evolved over time, and how well it does in preparing students to meet the challenges the workplace.

1.4 Aim

These studies investigate the instruction of computer skills in industrial design education with the aim of identifying knowledge gaps and generating insights that can help us understand how well it prepares students to meet the challenges of the workplace today and in the future, as well as to find ways of improving it.

By 'wow well,' it is meant: the diversity and extent to which comp. skills are developed, how this development is approached through the curriculum and how it helps students to address a number of current trends that are discussed in chapter four.

1.4.1 Research question(s)

This general aim can be synthetized in the following research question:

How well does higher education of industrial design prepare students to meet the challenges of the current and future workplace, in terms of the computer skills that students acquire as they transition through college?

1.4.2 Objectives

1. Determine the range of skills taught, the extent to which they are taught, and how this is their instruction approached through the curriculum.

2. Determine to what extent the computer skills fostered in Industrial Design schools help students to take advantage of developments in Basic Computing Education, Generative Design, Additive Manufacturing and Computer-mediated Education, and which pose challenges and opportunities in terms of the development of the computer skills of Industrial Design students

3. Make a series of recommendations on how this instruction can be improved.

4. Develop a framework that can help with the development of study plans.

The studies look at the subject of study from a global perspective, rather than focusing on a specific type of school or region of the world. The rationale, is that while conditions may change locally, computer skills are universal. Keirl (2006) for example, points out that "*Any quality education today is an education for global citizenry.*" We think globally when it comes to economy, the environment, communications, health, etc. Not thinking globally about education is absurd (Keirl, 2006 p21).

1.5 Justification

The fact that much of the modelling process has been computerized, has made the range of models and modelling techniques available to industrial designers increasingly higher and complex. It has also made computer skills a matter of great importance. If, as discussed earlier, industrial designers have a dialog with the mediums they use when modelling–whether paper, clay, 3D software. Their modelling skills with each of these mediums, have an impact on the fluency of this dialog and though process. On the other hand, if computer modelling has supplanted traditional modelling to an important extent, it follows that the computer modelling skills of designers are essential in allowing them to have a 'fluent dialog' with the–digital–mediums accessible through the computer.

Thus, the importance of looking into the instruction of computer skills cannot be overstated. Since there are models and modelling approaches that only exist in the computer realm, looking into the instruction of computer skills is to look into whether this instruction prepares students, to take advantage of these approaches or not. The importance of computer modelling skills thus, goes far beyond efficiency; as a medium, the computer medium enables modelling processes not available elsewhere (figure 1.3). This means that computers are often the only way to access these models and modelling approaches. Computers however, are different from other tools in that, they can be programmed to react to certain conditions (such as user input). They represent the closest embodiment we have of an intelligent tool.

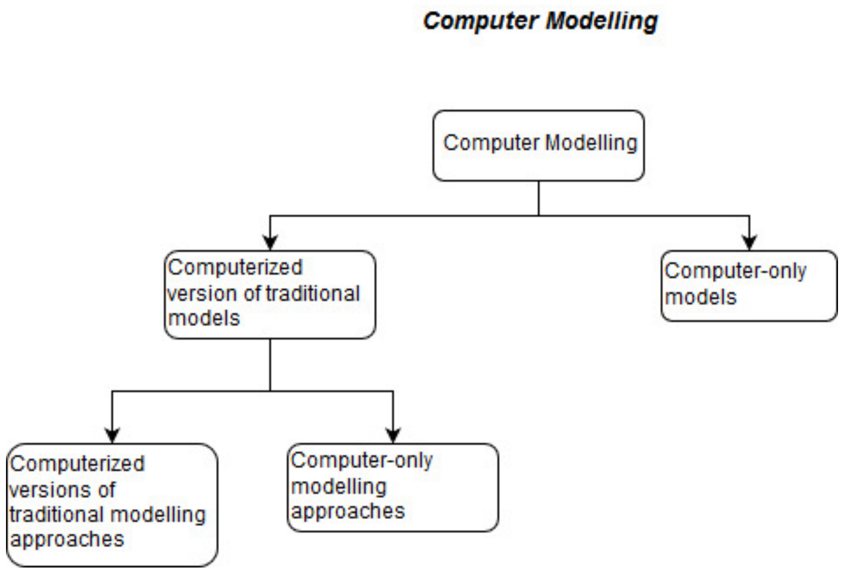

Figure 1.3 Computer models and modelling approaches

Computer models for example, do not have to be 'passive,' they can react and change with or without the designer's input. Similarly, in techniques like generative design, forms are generated using 'active' transformation approaches only possible in the digital realm. This has important implications because it changes the dialectic relation between designer and sketch, typical when working on paper. In fact, Oxman (2006) for example, argues that the digital design model is 'essentially contradictory' to other models such as those described in Schön's 'reflective practitioner,' in which visual representations of a design are manipulated while reasoning through a succession of stages, generally using sketches as medium. For Oxman, *"sketching as design thinking through iterative stages of visual discovery is the antithesis of the digital model"* (Oxman, 2006, p. 42).

Besides that Industrial Design depends extensively on models and modelling nowadays depends extensively on computer modelling, there are computer skills that have been identified in recent years as essential computer literacies for any person in the 21st century, most notably computer programming. Jeannette Wing (2008, 2014) for example, argues that every person will need what she calls 'computational thinking skills' by the middle of the century. While Miles Berry (2015) provides an account of how the entire national basic computing education curriculum in the United Kingdom, has shifted away from teaching students ICT to teaching them computer science, precisely with the aim of helping them develop those computational thinking skills. These developments represent challenges and opportunities for the instruction of computer skills (Varinlioglu et al., 2015). Looking into the instruction of computer skills in industrial design education, also makes it possible to know whether the skills that students are acquiring, address the challenges and opportunities that changes in context represent.

The other reason why these skills is important in Industrial Design education, is because of the rise of Generative Design. Generative Design is a computer modelling approach very useful in producing models that are very complex and often resemble structures from nature. These models are therefore often very aesthetically appealing, but at the same time very difficult to manufacture using traditional manufacturing methods. This in turn, ties in with the next reason why the study of computer skills is important. The rise of Additive Manufacturing. With the rise of Additive Manufacturing Methods, most notably 3D printing, the limitations of traditional manufacturing methods can be overcome. These trends are discussed in further detail in chapter four.

While it is in general well acknowledged, that higher education institutions have the obligation to address the development of the overall technological literacy of students. There is an impression that institutions fail to prepare college graduates when it comes to digital literacy (Ritz, 2011; Duggan, 2013; Katz, 2005; Dankwort et al., 2004, p. 1445; Duncan-Howell, 2012; JISC 2012; Murray & Perez, 2014; Ritz, 2011; Sioli, 2003, p. 3).

In the specific case of Design Education, Gun (2012) acknowledges that the role of computers in design has remained an open question for decades, and that academics and researchers have a responsibility to constantly look into computation and its relation to design, so that we can continuously advance our understanding on the subject (Gun, 2012, p. 2). Self (2012) too, says that design education needs to foster confidence in novice designers by providing students with *"greater awareness of the character of design tools, their strengths and limitations"* (Self, 2012).

Oxman (2006) recognizes that in light of the importance and depth of issues around digital design, it is necessary to develop a theoretical framework for the creation of design education theory (Oxman, 2006, p. 38). Investigating the instruction of computer skills, allows to see if, and how these issues are being addressed. When computers were introduced in design schools, it was not possible to see the potential negative effects of this transformation in the long term. Moreover, Tiene (2001) points out that something that often prevents teachers from doing educational research, is lack of knowledge about how to do it (Tiene, 2001 p40).

1.5.1 A Shifting Field?

The last two decades have seen the slow emergence of a discourse within Design Education. This discourse has often been the subject of debate and division within the field. Some of the arguments put forth in this discourse are for example; that the demands on design practice in the 21st century are significantly different, and that the nature of product designer's work 'has changed,' that Design has evolved from a narrow focus on aesthetics, to include areas such as services and branding. This discourse usually argues as well, that the boundaries between the design disciplines 'have disappeared,' that while in the past, product designers were only concerned with the physical components, now they need skills in a broader range of disciplines, and need to 'broaden their sphere of responsibility.'

In some cases, it is even claimed that now products are replaced by services, and that consequently, the product 'is vanishing' (Findelli, 2001; Gornick & Grout, 2008; Kiernan & Ledwith, 2014; Norman, 2010). In this context, it is claimed, the traditional 'crafting skills' that Industrial Design schools tend to develop are 'obsolete,' therefore, it is argued that Design Education must change.

In front of the other more 'glamorous' issues within the wider Design/Design Education discourse, the study of computer skills may seem unimportant. A proper critique of this discourse demands a much deeper discussion. Suffice to say here, that in these studies, it is considered that as long as humans live in a material world and in a market economy, not only the product won't 'vanish,' but the crafting skills that have historically defined Industrial Design will always be of central importance. It is also considered, that often the proponents of this discourse—perhaps because they lack actually a background in Design or a related discipline—make the terrible mistake, of confusing Industrial Design with other design fields such as Interaction Design, and/or User Experience Design, when in fact, each of these is a separate field, each with—ultimately—different design concerns.

That these 'new design domains'–shall all exist–may require other skills than those traditionally taught in Industrial Design schools is not questioned. That however, does not mean that the skills that define Industrial Design are irrelevant, or will anytime soon. In fact, even Norman (2010) himSelf, & Baek, 2017, seems to acknowledge this when saying: *"the need for styling, for forms, for the intelligent use of materials will never go away."* Moreover, this discourse seems to ignore the wealth of–serious–literature, linking craft and creativity. As discussed previously, crafting skills are essential in facilitating the thought process of designers–Design Thinking.

It is also the belief of this author, that at the core of this discourse, there is both; an old battle between specialists and generalists, and a human desire of emancipation and to reject the past, but once again such issues demand of a much deeper philosophical discussion well beyond the scope of these studies.

It has to be pointed out as well, that the view held in these studies is shared by a growing number of scholars and practitioners, who see the dangerous implications of this discourse. In fact one only has to see the reaction of commentators to Norman's (2010) article, or the bitter complaint of famed designer Gadi Amit (2010) to realize this. More recently for example, Efer (2017), has written that aesthetics, styling and brand, are what make up 'the very soul' of what the product is about, and calls for a fresh, renewed design education that brings conceptual design ideas to "physical fruition" (p192). Moreover, he emphasizes that creative abilities should result in "tangible products" to be manufactured.

In terms of the subject matter of these studies; Kiernan & Ledwith (2014) in fact, end up admitting that computer skills like 3D modelling, are recognized as important by practicing designers. Dönmez (2013), on the other hand, showed that 'using computers well' is the 2nd most important criteria after 'having a design degree' when hiring designers. In addition; a 2017, survey of the British Industrial Design Association (BIDA), strong CAD skills are recognized as a defining characteristic of good industrial design graduates and of the best industrial design programs (BIDA, 2017). And Luscombe (2018), in a simple yet powerful essay, has recently argued that, since all tools are thinking tools, they should be bestowed with "epistemic credit" (Luscombe, 2018 p.11).

1.6 Methods

These studies were carried out by doing a revision of the instruction of computer skills in a number of industrial design undergraduate programmes from different countries. Empirical data was acquired following a mixed-methods approach relying predominantly on the use of surveys and interviews as instruments for data collection. The study also used other approaches and instruments, such as documentary reviews, artefact analysis, and auto-ethnography, drawing from the author's own experience as lecturer.

Other sources, such as seminar and conference presentations were also used for the purpose of providing context for further discussion. This combination of methods are known to work well together, and can provide an up-to-date overview of the subject while adding a layer of reliability.

1.7 Contribution

These studies expand our current understanding of the instruction of computer skills in Industrial Design Education, by providing an up-to-date overview of what skills are being taught, to what extent they are taught, and how relevant they are for the current and future workplace. It reveals whether the current instruction of computer skills fosters skills beyond the usual areas of computer modelling and visualization—such as animation or project management—and also whether it fosters other forms of advanced computer modelling methods—such as Generative Design—besides the traditional 'digital hand-crafting' methods. It also expands our current understanding by showing how developments in areas relevant to the instruction of computer skills, represent challenges and opportunities, and provides examples of how these can be addressed.

This work contributes as well, with the development of a framework that can be used as a guide in the implementation of instruction of computer skills in industrial design education. This framework, can help schools in developing a computer-skills curriculum, providing guidelines and recommendations for industrial design schools to improve student's learning and application of computer skills. By looking at the computer skills they teach, and the approach they follow to teach them, it is possible to identify the position that Industrial Design Education has in general when it comes to technological education. In this sense, this study may contribute as well, towards the development of a theory of instruction of computer skills in Industrial Design Education.

While these studies focus on the instruction of computer skills from the perspective of Industrial Design, as a discipline that shares roots with other design disciplines like Interior and Graphic Design, the insights generated through this research can be relevant to understand instruction of computer skills in these other disciplines. At a more general level, this work serves as a reference for anyone interested in conducting research in this area in the future.

1.8 Outline of the Study

This first chapter, sets the general aims, objectives and motivation of the study. It starts by defining Industrial Design, and discussing the incorporation and role of computers in industrial design practice. It provides the background of the study by discussing industrial design education and the instruction of computer skills in industrial design schools.

Chapter two goes over the literature review. It starts by making a detailed review of scholarship about the Industrial Design process, ad reviewing how the usage of computers has evolved in Industrial Design. It also looks at the relation between industrial designers and engineers in the workplace, and continues by reviewing the computerization of modelling processes in Industrial Design, and the implications of this computerization. The chapter then moves on to reviewing the literature regarding the history of instruction of computer skills in industrial design education, and the different issues that have been identified with it over time. The chapter also reviews some of the different discourses around this instruction and around technology in general. The chapter closes with a summary of the literature review and provides a theoretical framework for the discussion in following chapters.

Chapter three goes over the research approach, it describes in greater detail the methodology followed in the studies; from the theoretical basis for the selection of methods to the design of the instruments for data collection, sampling, and methods of analysis. It discusses the advantages and disadvantages of each approach and the tools and techniques used.

Chapter four provides a review of the context on which this research is framed. Through a mix of secondary and primary sources; it looks at four trends that impact the instruction of computer skills; the rise of computer programming as a basic literacy in the 21st century, the rise of computer-mediated education, the rise of Additive Manufacturing, and finally, the rise of Computational Creativity and Generative Design, and its application to Industrial Design and thus Design Education.

Chapters five and six present the findings of the study using a series of statistics and charts. Each chapter then moves on to discuss the results through a comparative analysis of the information gathered. This discussion is framed in the context of the literature review and the information presented in chapter four. The chapters identify areas of opportunity to improve the instruction of computer skills in Industrial Design schools. Based on this discussion, the prevailing understanding of CAD in Industrial Design is challenged. The chapters close by discussing the different possibilities to improve the instruction of computer skills. The discussions are complemented with findings from other works such as a study looking at the level of digital literacy of higher education students.

Finally, chapter seven presents the conclusions drawn from this study and revisits the implications over the instruction of computer skills in industrial design education. These studies argue that a number of factors such as the definition of CAD that prevails in industry, and academia–that computer aided design equates computer modelling–have had a negative effect over the scope of computer skills fostered in industrial design schools. The studies argue as well, that a revision of computer instruction in industrial design schools is necessary to ensure that the computer skills that students acquire as they transition through college are appropriate. The chapter then, presents a series of recommendations on how to do this, and introduces a framework to facilitate this revision. Finally, the chapter ends by summarizing the achievements and limitations of the study as well as discussing possible paths for future research.

1.9 Chapter Summary

This first chapter, has introduced the general aims, objectives and motivation of the study. It has defined Industrial Design, and discussed how the Industrial Design process depends extensively on the use of models. Models are at the centre of industrial design practice and education of the discipline. They are an essential tool for communication and discussion of design ideas. It has shown how with the advent of personal computers modelling has been increasingly based on computers, what is generally known as 'Computer Aided Design' (CAD) in the design and engineering disciplines. The introduction of CAD in Industrial Design brought with it a new way of modelling, a type of modelling mediated by technology. This transformation implied a new way of doing things. Since CAD has partially replaced physical modelling mediums such as clay or foam, it has become the medium through which designers explore, learn about, and create forms. Moreover, like any other medium it has particularities of its own that must be understood by anyone who wants to master the medium, it demands from the user to do things in a certain way. CAD provides ways to automate aspects of modelling, and to make models in ways that only exist in the computer realm.

Issues with past research looking into the development of this type of computer skills have been discussed. For example; most of this research has been approached from the perspective of Computer Aided Design, not from the perspective of computer skills in general. The importance of the study has been discussed and framed against the broader discourse of Design Education. Besides that computer modelling has become essential in Industrial Design practice, studying the development of computer skills is important because computer skills not related to computer modelling have been identified as basic literacies for any person in the 21st century. The aim and methodology of the studies has been presented. The studies are based on a combination of surveys and interviews that were complemented with information from other sources. Finally an outline of the thesis' chapters has been presented.

Chapter Two

2 LITERATURE REVIEW

Having established the background of the study, this chapter goes into the subject matter in more detail. The chapter starts by presenting an in-depth review of the design process, to then define what is computer modelling, and how it is different from traditional modelling methods often used in Industrial Design. The chapter also identifies certain discourses around CAD education that serve as reference for the discussion in subsequent chapters.

2.1 Industrial Design Process

Bryden (2014) says that the design process is an *"an iterative process of research, analysis, thinking, conceptualizing, visualizing, model making, prototyping, testing and refining"* (Bryden, 2014 p.7). Earle (1994) on the other hand, defines the design process as: *"a way of devising innovative solutions to problems that will result in new products or systems"* (Earle, 1994 p.15). Similarly, McMahon and Browne (1998), agree that *"design progresses in a step-by-step manner from some statement of need through identification of the problem (the specification of requirements), a search for solutions and development of the chosen solution to manufacture, test and use"* (McMahon & Browne, 1998 p.5). In addition, Kalpakjian et al. (2014), on the other hand, argue that Product Design *"involves the creative and systematic prescription of the shape and characteristics of an artefact to achieve specified objectives, while simultaneously satisfying several constraints"* (Kalpakjian et al. 2014 p.7).

The Design Process is an iterative process that generally starts with the identification of a market need or an area of opportunity to develop a new product, which in turn may represent an opportunity to expand an existing product line. The ultimate goal is the complete specification of a product that satisfies the market need, or fills the area of opportunity. This process involves defining and/or re-defining the problem precisely, determining requirements–design criteria–based on a number of factors such as potential consumers–users, context of use, etc. (Ashby & Johnson, 2013 p.33; Cuffaro, 2006 p.24; Earle, 1994 p.18; Kalpakjian et al., 2014 p7). Along the design process, it is common that industrial designers interact with marketing, engineering and other R&D specialists (Cuffaro, 2006 p.25).

In a market economy, New Product Development usually occurs in response to a market need, which is then translated into a design brief. Industrial designers are then responsible for finding ways to meet the design brief, by developing design ideas all the way from a vague concept, to detailed Product Design Specifications. This process is aided by a wide-range of R&D and marketing personnel (McMahon & Browne, 1998 p.4).

This stage is heavily assisted by market analysts and/or sales specialists, who provide insight into market trends and consumer desires/needs (Ashby & Johnson, 2013 p.33; Cuffaro, 2006 p.24; Earle, 1994 p.18; Kalpakjian et al., 2014 p7).

2.1.1 Basic Design Process Model

There is a general awareness that due the wide variety of design scenarios, there is no detailed process model that serves all purposes (McMahon &

Browne, 1998 p.5). In fact, Burdek (2005), says that often, models of the design process are most valuable as didactic tools, to explain the design process and to train designers in logical thinking (Burdek, 2005 p.226). In general however, and at its most basic level, there are three broad stages that can be identified in the Industrial Design process, irrespectively of the specific scenario. These stages are: research, conceptualization, and refinement (Cuffaro, 2006 p.22). There are numerous variations of this process, and often these three stages are called differently, for example; Ashby & Johnson, (2013) refer to these three stages as: conceptual design, development and detailed design (p.34). Similarly, McMahon & Browne (1998), propose a model of the design process including four steps: clarification of the task, conceptual design, embodiment design, involving the development of the design in more detail, and, detail design, involving the detailed specification of dimensions, tolerances, materials and form (p.6).

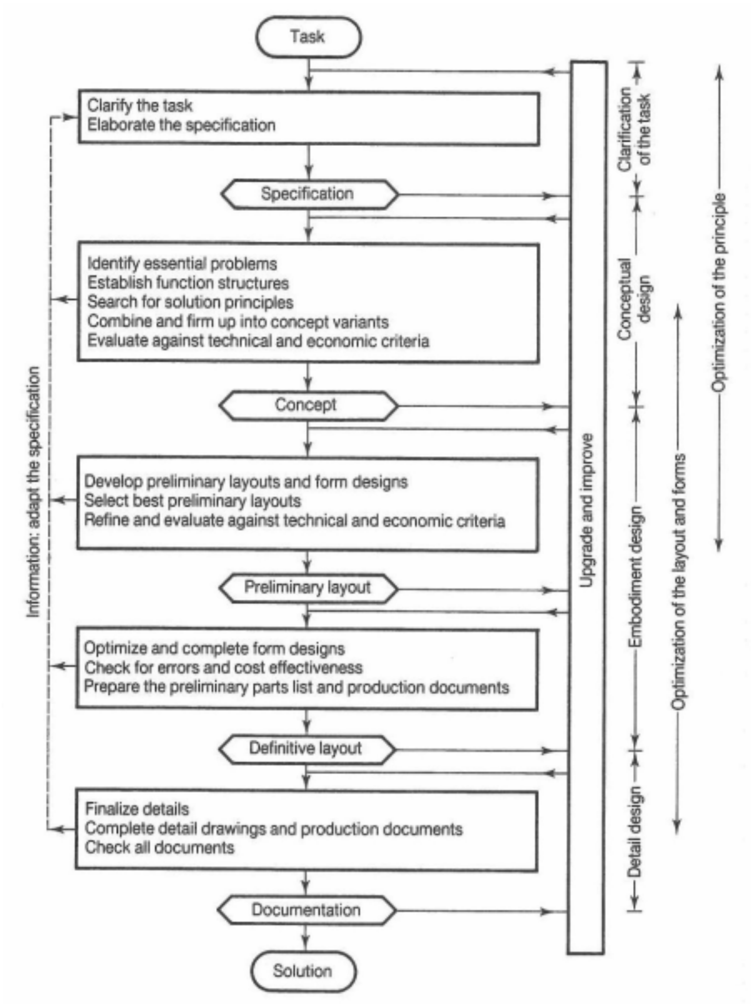

Figure 2.1 Model of the Design Process proposed by McMahon and Browne (1998 p.5).

According to Burdek (2005), the development of this basic design process, was heavily influenced by developments in Space Research. In this area, there was an effort to *"dissect the design process into discrete steps,"* being those: 1. Understanding and defining the mission, 2. Collecting pertinent information, 3. Analysing the information gathered and drawing conclusions, 4. Developing solution alternatives, 5. Assessing the different options generated, and further developing the best (Burdek, 2005)p252

It should be noticed, that this basic model of the design process, has many parallels with the overall creative process of Wallas (1976). And very much like this general creative process, diverges during the initial stages, and then converges towards the end, while trying to find an optimal design solution.

Cuffaro (2006), is careful to notice that the Product Design process is itself part of the much larger process–the macro process–of New Product Development (NPD), which is beyond the activities of any specific discipline (Cuffaro, 2006)p24. Often these two macro and 'micro' processes, since are both about the development of the product, are confused in the literature. This in part, creates confusion about the scope and areas of competence of Industrial Design as a discipline.

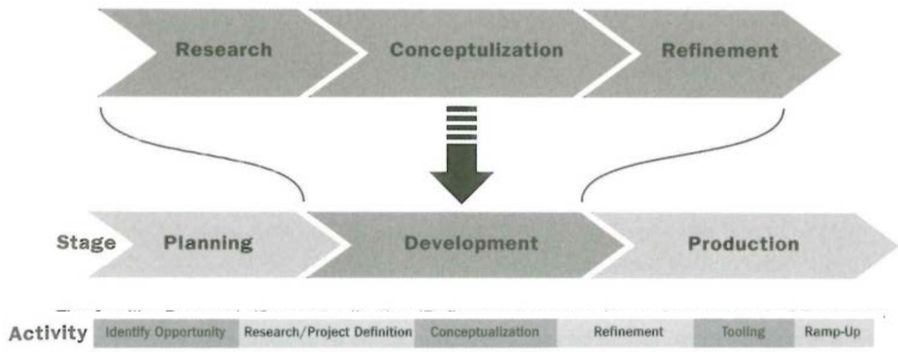

Figure 2.2 The 'macro' and 'micro' processes of New Product Development and Product Design Process side by side, along with the participation level of the Marketing, Design and Engineering personnel at each stage (Cuffaro, 2006).

Industrial Design or Product Design process–the micro level–mostly the responsibility of the Design Department, which at least in large corporations is usually part of a larger Research and Development (R&D) Department. Design

Departments, can be integrated by a range of different designers–mostly Industrial–and different types of engineers.

In the early stages of the design process, a tentative solution is proposed. This solution, is evaluated from different viewpoints in order to determine its strength to meet the design requirements. At the conceptual stage, a wide range of ideas–technical, aesthetic, functional, etc–are considered. Once evaluated against the design criteria, during the development stage, promising concepts are further developed, analysed and tested (Ashby & Johnson, 2013 p.34; McMahon & Browne, 1998 p.7). Finally, at the last stage, the process proceeds in the same way to complete the Design for Manufacture proposal

The information recovered through market research–and other sources including internal organization's information–is used to put together a list of criteria or requirements, which are then used to guide all subsequent development efforts, and to evaluate design ideas (Cuffaro, 2006 p.25)

Cuffaro (2006), also recognize that design research is just an extension of the wider market research (p26), but the aim is to get a deeper insight that allows designers to determine the specific characteristics of the product to be designed (Cuffaro, 2006)p26

At this stage too, the attributes of the form are determined, and the potential suitable materials are explored (Ashby & Johnson, 2013).p34

During the refinement stage, industrial designers and engineers start to work more closely around aspects related not just to the final product configuration, but also the design for manufacture. At this stage, suppliers are identified, assembly processes are devised, and costs are revised. During this stage too, a number of design models are usually produced, these include: detailed renderings, drawing packages, appearance models, and a CAD database (Cuffaro, 2006 p.27).

This stage concludes with the configuration of a feasible solution that is then ready for specification and final testing. During the specification stage, tolerances, materials, finishes and manufacturing processes are specified (Ashby & Johnson, 2013).p34

Once the design department has generated a Product Design Specification (PDS), industrial engineers and/or process-planning specialists, will start developing detailed manufacturing plans and strategies (McMahon & Browne, 1998 p.4).

Similarly, Findelli (2001), admits that the most practiced and logical structure of the design process is 1. The identification of a problem or need, 2. The envisioning or conceptualization of a final goal or solution, and 3. The development of that solution envisioned in further detail.

2.1.2 Other Variations

More and Less detailed versions of the design process can be found in the literature. Earle (1994) for example, proposes a six-stage design process including: 1. Problem identification, 2. Preliminary ideas (ideation), 3. Refinement, 4. Analysis, 5. Decision, and 6. Implementation. (Earle, 1994 p.18).

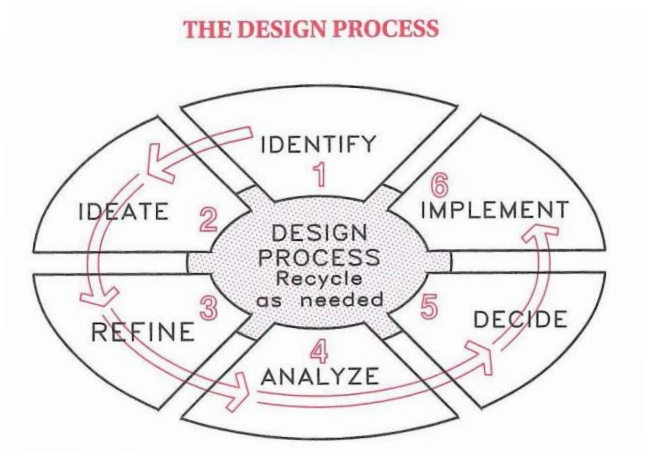

Figure 2.3 Model of the design process proposed by Earle (1994).

Similarly, Bill Dresselhaus (2000), outlines a six-stage process (Fig ____). Once again, it should be noticed, that often the literature makes no distinction between the overall New Product Development Process (NPD)—the macro level—under the control of top management and involving several disciplines/departments, and the Product Design process, usually under the control of the Design Department alone.

Figure 2.4 Model of the design process (Dresselhaus, 2000)

There are also, other, less-structured models of the design process. In contrast to these mode detailed versions of the design process for example, the 'bubbles' model of Ken Wallace. In this model, there is no linear path between the design brief and the product specification, instead, there is an 'n' amount of bubbles that could appear between them (fig. 2.5). This approach may be more representative of how the most experienced designers work, but by far, a typical design process is defined by a series of iterative steps or stages (Ashby & Johnson, 2013). P36

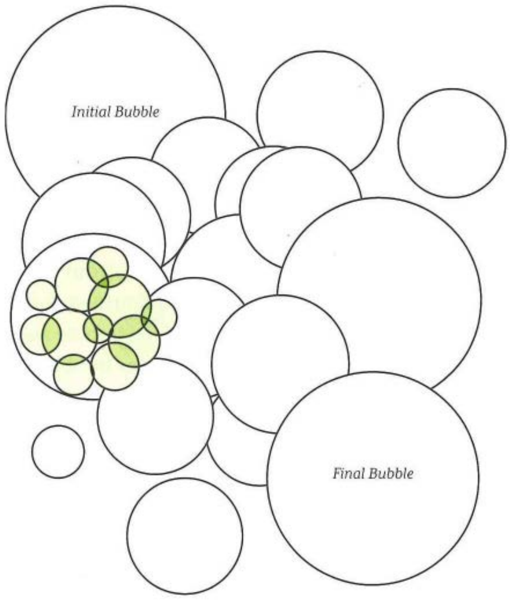

Figure 2.5 Model of the design process by Ken Wallace (Ashby & Karan, 2013).

Increasingly shorter development times of today's complex products, are achieved by assembling together large teams of people, who collaborate in the development of new products (McMahon & Browne, 1998 p.4).

2.1.3 Concurrent Engineering

Several scholars point a the transition/evolution of the design process from being a purely sequential/linear process, towards being a much more iterative process in which activities take place at the same time. This 'parallel' approach is known as 'Concurrent Engineering' (Kalpakjian et al. 2014 p.7), and its main idea has likely been adopted in other fields—essentially the same approach is known as 'agile development' in the IT industry.

In a traditional design process, Product Design Specifications are produced by the Design Department and passed on to the manufacturing specialists for the preparation of mass production plans—also known as an *"Over-the-wall"* approach (McMahon & Browne, 1998 p.12). Often this means that Design for Manufacture considerations are not considered until the Product Design Specification is passed on to the manufacturing specialists for the first time. Then, because the Design Department tends not to take into account these Design for Manufacture considerations into account until this point, when the Product Design Specification is passed on to the manufacturing specialists, more often than not, the manufacturing specialists will ask for modifications, and will send the Product Design Specification back to the Design Department for amendments. Depending on the particular scenario, this can happen several times, making the process highly inefficient and costly (Desbarats, 2010; Kalpakjian et al., 2014; McMahon & Browne, 1998 p.12)

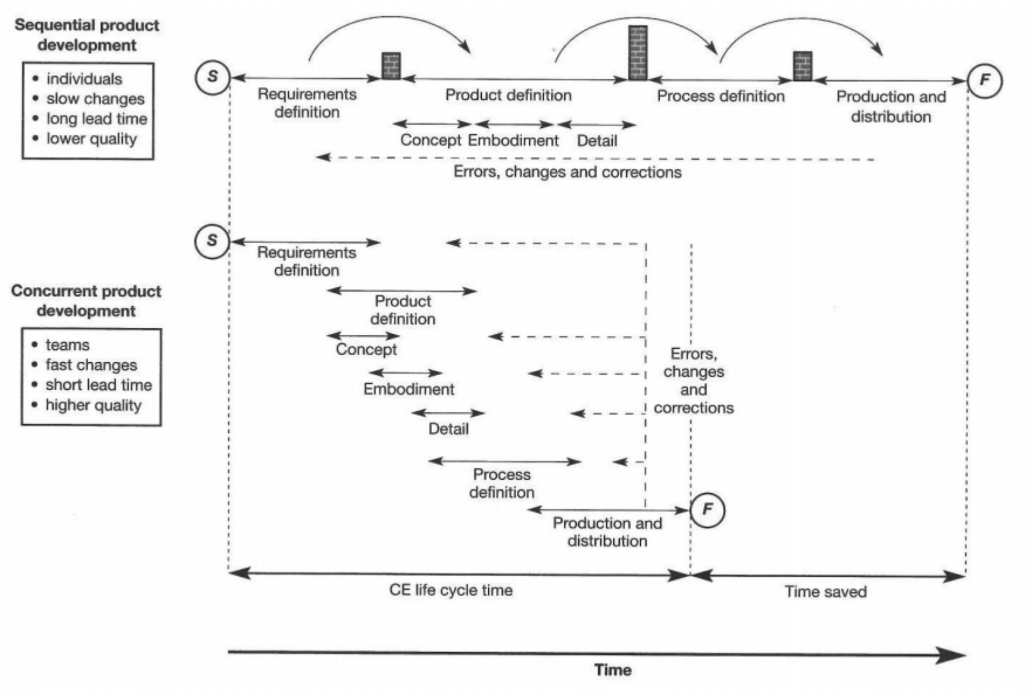

Figure 2.6 Models of the Traditional or Sequential, and Concurrent product development processes side-by-side (McMahon & Browne., 1998 p.13).

While a sequential approach–also known as 'cascading'– may be the most suitable in certain cases, it has a number of important disadvantages, perhaps the main of which is a longer development time, in addition, an issue that has plagued New Product Development for years, is the loss of design intent. As the design of the product 'passes on' from department to department, it suffers constant alterations, so that the resulting product may differ substantially from what the market required (Kalpakjian et al. 2014 p.7). Porter (1995) as well, says that the earlier that concurrent design happens in the new product development process, the greater the gains in lead-time and cost reduction. A basic requirement for concurrent design and engineering is the 3D model (Porter, 1999).

Concurrent Engineering aims to overcome these issues by assembling together a group of specialists in different areas, to work together at all stages of the development of the new product; from conception through manufacture and use in service to maintenance and disposal. (McMahon & Browne, 1998 p.12). Pushed by the need to bring products to market in increasingly shorter periods, the aim of Concurrent Engineering, is to involve all stakeholders of the New Product Development Process from the early stages, avoiding this way, sudden and costly changes at later stages (Kalpakjian et al. 2014 p.7).

As can be seen in fig 2.7, Computer Aided Design has been instrumental in facilitating Concurrent Engineering, because CAD models provide a 'central description' that informs all design and manufacturing activities (McMahon & Browne, 1998 p.15).

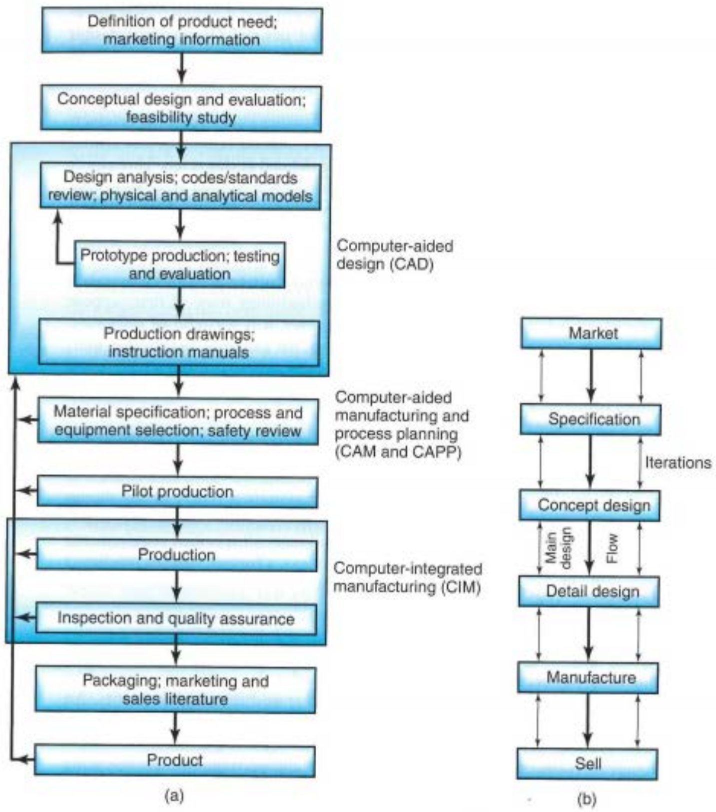

Figure 2.7 The traditional/sequential design process, and the Concurrent Engineering approach side by side (Kalpakjian et al., 2014).

2.2 Computer Modelling

In the previous chapter, the crucial importance of models in Industrial Design practice was discussed. Computer models can be seen as computer representations of geometric entities that could exist in the real world. A CAD model is the set of information, usually mathematical expressions of geometric bodies, expressed in computer code, and which represent a geometric entity/body, whether two or three-dimensional. This information, which is s contained in a computer file, can be made manifest in different ways, such as when displaying an image of the model in a computer screen, or when being physically built such as when using a 3D printer. Schoonmaker (2002) proposes the following definition for a CAD model **"***the*

3-D computer graphics based object that users create and interact with" (Schoonmaker, 2002, p. 172).

According to the type of model created, computer modelling can be divided in two and three-dimensional. Since CAD models are based on mathematical abstractions of geometry, they are always 'accurate,' because at their core there is a mathematical formula. Two physical models even if carefully built, cannot be assumed to be perfectly similar.

2.2.1 2D models

It has been shown in chapter one, that while drawings may not be immediately seen as models, any representation of the world is essentially a model. Similarly, while CAD may not be immediately associated with drawing, the first type of models 'computerized' were drawings ("What is CAD," 1992).

Lostritto (2012) says that, if drawing is defined as any representation that exists in two dimensions, then a drawing not only is a 2D model, but it can exist in the traditional paper medium and in the 'computer medium' such as the screen (Lostritto, 2012, p. 61). The fact that CAD entered creative fields such as Industrial Design by replacing drawing can explain why CAD has been taken as synonym of drawing in the past; in 'What is CAD' for example, it is argued that CAD is used in industries that are 'drawing intensive.'

Similarly, Ye et al. (2004), say that in the context of new product development, CAD is taught to "assist their design drawing and drafting" (Ye et al., 2004), and Dankwort et al. (2004) recall that early CAD systems were introduced to reduce time and money by replacing the conventional drawing board (Dankwort, Weidlich, Guenther, & Blaurock, 2004b).

2D CAD models present a series of advantages of hand-drawings, such as: dynamic measurement, all kinds of transformations, automatic dimensioning, groupings, user-defined options, hiding and showing drawing elements, undo and redo, plus the creation of related geometry (offset, copies, patterns, fillets, etc.), and the ability to perform mathematical analyses using those drawings (Schoonmaker, 2002).

2.2.2 3D models

3D CAD models can be classified in different ways; such as by the type of computer geometry used to build them, the number of parts they contain, or the modelling approach used to build them. According to the type of 3D models created, 3D modelling can be divided in Wireframe, Solid, Surface and Polygonal. According to the procedure or environment created computer modelling can be divided in parametric, and Non-Parametric (Bryden, 2014; Bilalis, 2010; McMahon and Browne, 1998; Schoonmaker, 2002; Ye et al., 2004).

Bilalis outlines the overall 3D modelling process as follows: starting with a rough sketch, applying colour and texture to it, creating the 3D model from the 2D sketches and then rendering and animating the model to further evaluate, present and sell the concepts (Bilalis, 2000). Methods differ between different CAD programs but there are some similarities, and fundamental modelling concepts are largely the same (Bryden, 2014).

Some advantages of 3D models in general include: visualization, automated generation of drawings, analysis of geometric properties, interference checking, intelligent models, associativity, the facilitation of integrated product development (concurrent engineering) (Schoonmaker, 2002; Bilalis, 2000).

2.2.2.1 Solid Models

Solid models are defined by mathematical expressions that represent volumetric bodies such as a sphere, a cylinder, a cube, etc. Solid models can never be 'open,' thus, there is no way to unwrap a solid model like a polygon can be unwrapped. Unlike NURBs-based models, the surface of Solid models has no control points, therefore it cannot be 'sculpted' like NURBs models can. On the other hand, this also means that the surface definition is very stable and can be scaled down or up without showing issues. For this reason, Solid models are commonly used in CAD packages geared towards mechanical engineering. Most parametric modelling environments are based on Solid models due their characteristics.

This type of models are assumed to be 'massive'–filled volumes. According to Schoonmaker (2002), it is now assumed that all 3D CAD systems will create solid models, however, this might be the case of particular industries or disciplines only. In the case of Industrial Design, there is no definite preference between solids and surface models as it will be discussed later (Schoonmaker, 2002).

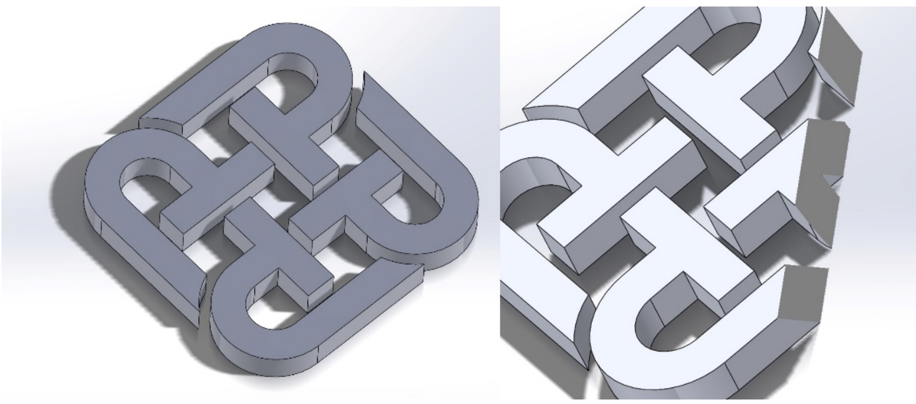

Figure 2.8 Example of a solid model, the model is 'massive' (by author)

Solid models have several advantages, such as that a number of 'physical properties' can be calculated and known, such as volume, mass/weight, surface area, and moment of inertia. In addition, Solid models can be used with 'Boolean operations.'

Roughly speaking, the Solid Modelling process consists of sketching 2D geometry (lines, arcs, dots, etc.) on a plane, and then using that 2D geometry to create 3D geometry by process of extruding, revolving, and sweeping among other operations.

2.2.2.2 Surface Models

A Suface Model, is defined by mathematical expressions that define surfaces, and which can be represented graphically using computer graphics. As opposed to solid models, surface models are models of geometric bodies that are not 'massive,' they are only 'a shell.' These models can form 'closed' or open bodies, and can be completely flat as well. Surfaces in surface models have no thickness, which is physically impossible, and these surfaces can be curved in one direction, or both (double curvature). Most surface models are based on a mathematical model known as NURBS, reason why some CAD systems are known as 'NURBS' modelling systems. NURBS stands for on-uniform rational basis spline–*mathematically defined curves that have their shape controlled by control points sitting off the curves* (Bryden, 2014).

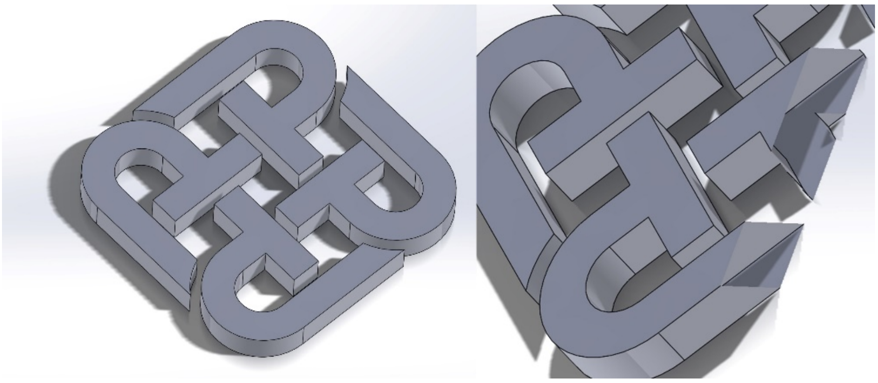

Figure 2.9 Example of a surface model, the model is hollow (by author)

Surface modelling is commonly employed in the modelling of parts with 'free-form' double-curvature surfaces, such as car-panels. In some cases these surfaces are known as 'Class-A' surfaces because these are the surfaces that the consumer sees. These are surfaces are most often the surfaces that industrial designers work-with (Bilalis, 2000; Bryden, 2014; Schoonmaker, 2002). Common CAD software based on NURBS include: Autodesk's Alias (formerly Alias Wavefront) and McNeel's Rhinoceros 3D.

Some of the advantages of surface modelling are the ability to create surfaces based on a 'cloud of points' usually generated through a 3D scanner or digitizer, or through a series of cross-sections, which is useful in reverse-engineering processes.

2.2.2.3 Parametric models

Parametric models are models in which mathematical formulas are programmed into the model so that features are related to each other. In parametric models, the construction history is stored in the computer file, and offer the advantage that the model can be 'rolled back' to a given point in its construction history. After modifications have been made at any of these 'history' points, all subsequent modelling features/actions are automatically updated. In consequence, the order of construction is very important in this type of models. This construction history is commonly known as 'history tree.' (Schoonmaker, 2002, p. 185). Bilalis (2000) argues that since the 1990's all solid modelling systems are parametric in nature.

One advantage of parametric modelling systems is the ability to work with assemblies–or assembly models. This facilitates concurrent engineering in large-scale R&D teams (Bilalis, 2000). Bryden also notices that an advantage of solid–parametric–models is that they can be used in computer engineering software to perform engineering calculations that can only be performed on models with a 'mass.' In addition, the fact that these models often require less work when using them for rapid prototyping or computer-aided manufacturing (Bryden, 2014).

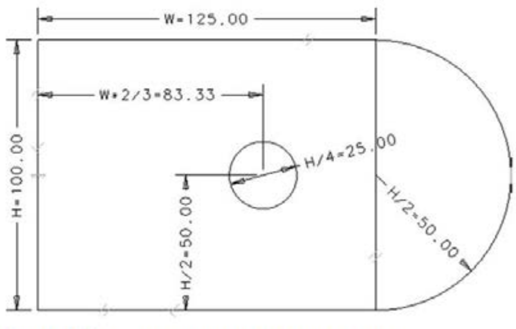

FIGURE 8.18 A simple example of parametric dimensions.

Figure 210 Example of a sketch in a parametric modelling environment (Schoonmaker, 2002 p212)

2.2.2.4 Polygonal models

Polygonal Models, are three-dimensional models defined by a collection of two-dimensional polygons (generally triangles or quads), each of which forming a surface, and arranged next to each other to form a three-dimensional mesh. Polygonal models are the most basic type of models, and because of their versatility, polygonal models have been extensively used the entertainment industry.

Polygonal modelling is like working with a wireframe mesh. Polygonal models have the benefit that forms can be sculpted more intuitively and that the size of the computer file they generate is typically smaller than in the case of surface models. These models are suitable, and widely used in animation and the entertainment and video games industries. Polygonal meshes are usually made of triangular or four-sided polygons. These polygons can then be subdivided where necessary to achieve more detail in the mesh (Bryden, 2014).

Until recently, polygonal models were not widely used in industrial design due that while the type of geometry they generate is good for visualization purposes (rendering), it cannot be used with traditional or extractive rapid prototyping/CAM techniques such as CNC milling. In fact, it is clear that some scholars do not consider polygonal models—and consequently polygonal modelling systems—as CAD at all. Bryden (2014) for example considers that CAD modelling is divided into two basic techniques: surface modelling and solid modelling (Bryden, 2014). Common 3D-CAD modelling packages based on polygonal modelling include: Autodesk's Maya and 3DS Max and Maxon's Cinema 4D.

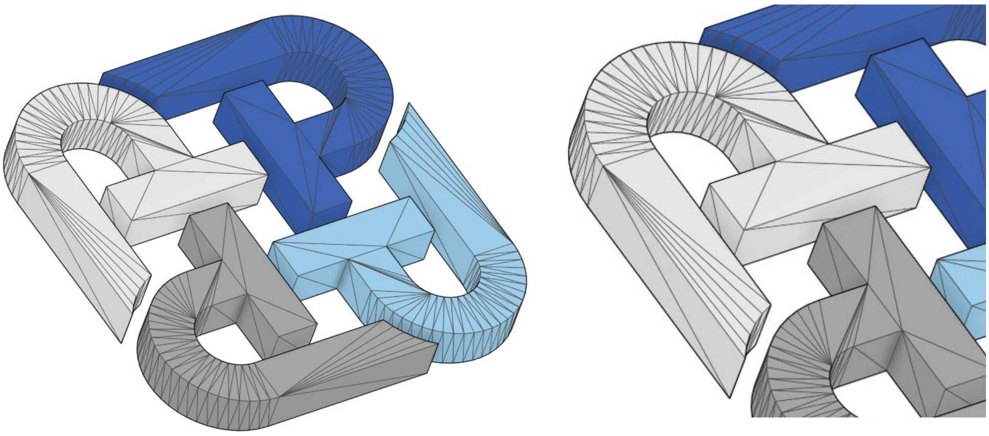

Figure 2.11 Example of a polygonal model, the model is made of multiple polygons [triangles] (by author)

2.2.2.5 Assembly Model

Assembly models are made of several other models, each being a different part of a larger product. Usually these models use 'instances' of each of the different parts that they group together, meaning that they just refer to each of the parts and position them in relation to each other, but the assembly models itself does not contain the actual model of each of the parts. In assemblies, parts are modelled separately and then put together. This requires a specific way of working; you need to know how parts fit together (Bryden, 2014).

2.2.2.6 Best approach

Each of the 3D model types previously described, involves its own modelling approach, each requiring of a different 'know-how.' Solid modelling is as if sculpting in a machining workshop, surfaces modelling on the other hand is like working with pieces of stretchable cloth (surfaces are trimmed, lofted and stitched). In solid/parametric modelling, the references to create models are usually dimensions or technical drawings. In surface modelling on the other hand, there are no references or the references are sketches or hand drawings where dimensions are not important (Bryden, 2014). Parametric/solid modelling also requires an understanding of concepts like dimensioning constraints and Boolean operations so that the benefits of such approach are exploited. Another concept intrinsic to parametric modelling is 'associativity.' Whereby a link is established between a model and its drawings or between a model and an assembly, so that whenever on is modified, the other is automatically updated (Bryden, 2014).

Also, Qi et al., (2010) recognize that parametric solid modelling has a longer/steeper learning curve, in comparison to NURBS modelling. Assemblies for instance require the user to be familiar with the concept of 'instancing.' Chester (2008) also says that parametric modelling involves much more predicting. In surface modelling on the other hand, concepts such as the surface continuity–how smooth is the transition between surfaces–is very important (Bryden, 2014).

Although hybrid parametric systems–commercial titles would include PTC's Pro-Engineer and Dassault's Solidworks–allow the modelling of surfaces, the way surfaces are built and manipulated is not the same as in a 'pure' surface (NURBS) modelling program. This is because the way surfaces are mathematically defined is not the same; in NURBS modelling programs, surfaces are made from curves that do not have to be sketched on a flat surface/plane. In parametric modelling programs however, surfaces are typically built from curves that exist on a flat plane, even if the resulting surfaces have double-curvature (Bryden, 2014).

There seems to be therefore, no agreement about whether there is a 'best' computer modelling approach for Industrial Design. Desbarats (2010) argues that "product development is done in solids" and that industrial designers and engineers should both work in solids. He notices that the role of industrial designers has evolved to a point where industrial designers are expected to produce final and ready for manufacture documentation. In this regard, advances in solid modelling CAD systems have shown that industrial designers can not only successfully master the modelling of complex surfaces, but also to manage certain manufacturing issues such as draft angles and packaging of internal components. This has solved a long problem of collaboration with engineers not willing to take responsibility over models created by industrial designers due mistrust of their ability to deliver manufacturable forms (Desbarats, 2010).

The previous argument is fair; being able to collaborate interdisciplinary with other teams has allowed industrial designers to unlock the benefits of product lifecycle management (PLM), creating designs that make more effective use of data assets shared across these teams. This has been acknowledged by other scholars as well (Bryden, 2014; Porter, 1999; Unver, 2006).

There are however, equally strong arguments that favor the use of Surface modelling–ture NURBS Surface modelling that is–over Solid modelling. Bryden (2014) for example, argues that surface modelling is clearly more suitable for the way designers work, particularly at the beginning of the design process. Similarly, Gill Chapman noticed that a suitable CAD system for product designers should offer the ability to "*work in a 'sculptural' way with surfaces*," because, as he acknowledges; it is crucial to "*the way product designers want to work*" (Chapman, 1999). Ashby & Johnson (2013) too, assume that

Industrial Designers work with surface models (p34). And Bilalis (2010) acknowledges that CAD systems for industrial design are mainly surface modelling tools with 'very good' rendering capabilities (Bilalis, 2000).

It could also be said, that overall, surface modelling resembles more like modelling foam and is more suitable for form exploration. Solid/parametric modelling is more suitable for specification and design of assembly features. Because in everyday practice in most modern enterprises industrial designers need to collaborate with engineers, an ideal computer modelling education would include both: surface and parametric modelling.

In addition, in some cases, the advantages of solid/parametric modelling are difficult to capitalize because it is hard to find plane surfaces to reference a sketch to and standard Cartesian planes are not useful in this case. Thus, parametric models are less suitable for free-form visual exploration (Bilalis, 2000; Bryden, 2014; Schoonmaker, 2002).

Similarly, while some scholars suggest that surface modelling is more difficult, some others say it is actually easier. Schoonmaker for example, acknowledges that "surface modelling can be quite complicated," (Schoonmaker, 2002, p. 225). Similarly, Unver (2006) recognized that the learning curve of hybrid systems is harder (Unver, 2006). An explanation of that apparent contradiction, is that often times, the literature omits whether the discussion presented is about Surface modelling in a parametric environment, or in a NURBS environment.

While there may be no agreement whether Solid or Surface modelling is best for Industrial Design, it is in general agreed that both are necessary (Dönmez, 2013). Bryden (2014) for example, acknowledges that a common workflow in Industrial Design, is to use surface modelling first and solid modelling later, because it is early in the design process when product specifications are still unclear and when designers need the most freedom to explore forms. Only when form exploration is not important or when the product's surfaces are relatively simple, parametric modelling would be suitable from the outset. The experience of this author as industrial designer confirm this as well.

2.3 Industrial Design Education

Craft and problem solving have defined Industrial Design Education, it is an education based on a model derived of the school of the Bauhaus, which revolves around a design studio course, which in turn is used to simulate design practice (Oxman, 2006; Cross, 2007; Findelli, 2001; Kolko, 2005; Valtonen, 2007). Industrial designers are educated in university programmes where they learn to make a variety of models using different techniques. Industrial designers were mostly trained in Art schools until degree programs started to appear (Chapman, 1999; Oxman, 2006; Starling, 1999; Unver, 2006). As it will be shown later, the findings of this study reveal that this may be changing. Courses in engineering schools on the other hand have a more technical orientation and tend to place more emphasis on visual, aesthetic and ergonomic aspects (Unver, 2006).

2.4 Computer Skills in Industrial Design Education

The most prominent form of computer instruction in industrial design education is precisely that which has to do with Computer Aided Design. It is an instruction that essentially focuses on developing computer modelling and visualization skills, initially by teaching computer drawing—Ye et al. (2004) for example, say that, in the context of new product development, CAD is taught to “assist their design drawing and drafting” (Ye et al., 2004).

The introduction of mid-to high-end CAD systems in industrial design schools only started to take place in the late 1990’s early 2000s.

This instruction has relatively worked well in exposing students to the advantages of CAD in terms of speed and automation when working with models. Lately this modelling and visualization has tended to focus more on teaching computer 3D modelling.

The development of computer skills in Design and Engineering disciplines has been studied in the past from a variety of angles; Dankwort et al. (2004) divide the knowledge of computer aided technologies in 1. General, 2. Specialized, knowledge and 3. Specialist knowledge (Dankwort, et al., 2004, p.1443). Being the general knowledge the responsibility of higher education, the rest is the responsibility of companies/workplace.

Chapman (1999) on the other hand, gives an account of how CAD instruction has been handled in the past at some institutions. In fact, a very similar approach was implemented at the University of Cincinnati where this author taught CAD to Industrial Design students, although ‘specialization paths’ were not available.

Unver (2006) in turn, gives another example, describing how CAD courses being three hours per week, and describes how during the first six weeks of the term students spent the first hour and a half of these sessions watching demonstrations, while the remaining time they received ‘computer-based training.’ During the remaining six weeks, students worked on a project: “*These* *projects might be a Lego car modelling for mid-range software, a car exterior and interior modelling for surface modeller, and 3D Animation and interaction with 3D characters for the animation software"* (Unver, 2006 p328).

2.4.1 Delivery

Tamasin Cole, from Middlesex University, reports of an attempt to establish asynchronous CAD education as early as 1994. Although this study was within the context of visual communication, not Industrial Design, it provides an argument to support face-to-face instruction in concluding that despite all the benefits of asynchronous instruction "such a system must never be seen as a substitute for contact teaching where skills are to be applied in a creative context' (Cole, 1994).

Similarly, in 'what is CAD', it is said that knowledge of the CAD systems can be obtained through manuals, personal training, and by trial and error. However, it is also clearly pointed out that an instructor is necessary, and that "It is usually a combination of the three, that produces a successful CAD user" ("What is CAD," 1992). Schoonmaker (2002) too, says that "*Only an appropriate CAD training program is going to be able to provide this proficiency*" (Schoonmaker, 2002, p. 1).

Unver categorizes a third method to learn CAD but not commonly used in industrial design schools: computer-based training–or CBT. In this method, the approach can be discrete or somewhat integrated to a CAD design project, but not the main design studio. Unver (2006) says that this is a good approach when the user already has some knowledge of CAD but not if he/she is starting from scratch. Students do not get enough feedback and struggle to transfer their skills to other projects.

2.4.2 Integration

Chapman (1995), found that most 'IT teaching' in Industrial Design schools is not integrated into other project work (Chapman, 1995). The same is acknowledged by Chester (2008), and Unver (2006). Chester links the dominance of this pedagogy to the fact that the emergence of CAD coincides with the behaviourist movement in education (Chester, 2008). Unver (2006) considers this to be related to the fact that in the past one had to do a number of drawings in order to 'master the skill' of drafting, this led to the 'discrete' model of teaching skills around the design studio.

Varinlioglu et al. (2015) divide the pedagogical models used in the instruction of computer skills in architecture schools into: discrete and integrated. Discrete refers to teaching CAD/CAM independently of the design studio, while an integrated model refers to embedding CAD/CAM learning in the design studio (Varinlioglu et al., 2015).

Both of these approaches have advantages and disadvantages; the discrete model for example has been found to make it difficult for students to apply their CAD skills to the design process. Varinlioglu et al. Say that most students in this model tend to use CAD tools just for representation purposes (Varinlioglu et al., 2015).

> Togay et al., (2016), and Chapman (1995), also recall that that the incorporation of CAD in industrial design schools has been often unsystematic, often times resulting in tools being left unused. They

recognizes a gap exists between what students learn at school and what practitioners require as expertise. Facts that often times are masked by the nice computer graphics that University departments often put forward, and which is the work of unique students, which hardly reflects the overall situation.

2.4.3 Advantages

In educational settings, CAD has been found to helps students to connect modelling and prototyping–through CAM for example. Also, students become less frustrated than when working in a 'traditional workshop,' because the 'new' techniques allow them to achieve more ("CAD-CAM In The Classroom," 2003). Just like in industry, they can explore more, do things more precisely, in a way, they can do more or things that otherwise would not be possible.

Chester (2008) also found that while using 3D CAD does not seem to improve spatial thinking skills, these skills are necessary to make effective use of CAD systems. He developed a pedagogy to help students improve these skills. Spatial thinking skills on the other hand are crucial for acquiring strategic knowledge. Unver (2006) too says, "The use of CAD technology can deepen the student's understanding of final form, structure and performance of a product" (Unver, 2006, p.324).

2.4.4 Insufficiency

Dankwort et al (2004) recognize that education of computer-aided technologies tends to be broad but superficial (Dankwort et al., 2004a).

Traditionally schools have taught a very narrow spectrum of computer technologies that can be applied in the new product development process. For example, CAD education should not just be about teaching solid or surface modelling, it should be a far more complex enterprise, in which students learn about the whole product development process under the lens of computer-aided product creation (Dankwort et al., 2004a).

Similarly Ye et al. (2004) suggest that CAD education in University is not fully adequate (Ye et al., 2004). They acknowledge that even though colleges have made the switch to 3D CAD, the syllabus presented to students was not comprehensive and systematic (Ye et al., 2004).

Schoonmaker too says that designers need to gain experience modelling the type of parts they usually work-with, and then mentally plan how to build the 3D model (Schoonmaker, 2002).

Similarly, Chester (2008) says that expertise in knowledge is developed by the application of strategic knowledge (knowing what strategies to follow (Chester, 2008). He says that just like Chess experts can recognise patterns on a board, 3D-CAD experts recognise patterns in items that can be modelled through the use of certain commands–algorithms (Chester, 2008).

According to him, 3D-CAD knowledge includes three types of knowledge: declarative command knowledge, specific procedural command knowledge and strategic (or metacognitive) knowledge. Declarative command knowledge is knowledge about the commands that are possible in CAD software. For Chester, this knowledge should come 'a priori' so that commands to mirror lines for example can be sought-for and identified in a given software (Chester, 2008, p. 8). Specific procedural command knowledge is the knowledge needed to accomplish a task. Chester says that this is the type of knowledge around which CAD instruction orbits; however, it is redundant beyond the scope of specific CAD packages (Chester, 2008, p. 8). Strategic 3DSM-CAD knowledge is the knowledge by which, through processes of Metacognition a modelling strategy is devised (Chester, 2008).

Similarly, Self (2012) acknowledges that expert designers use their experience on the design process to make more effective use of tools. Ye et al. found that most study programs focus on teaching students skill, "what buttons to push" but they fall short on theory (Ye et al., 2004, p. 1457). Consequently students know 'how to draw a line but not how to use CAD to enhance the design process and/or think of problems in a way in which the computer helps.

An issue may be the so called 'saturated curriculum.' Jim Budd (2011) offers a comparison of the typical Industrial Design curriculum before and after the advent of what he calls "Digital Technologies" and before and after the "Design for Experience" era (Budd, 2011). This comparison shows the continuous cluttering of the curriculum up to its current point of saturation.

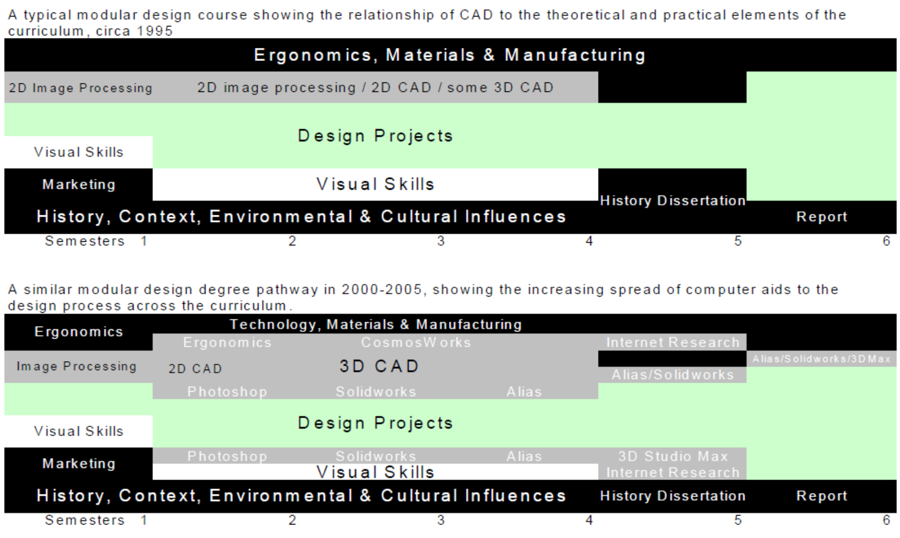

Fig. 3. Product design course structure.

Figure 2.12 Industrial Design curriculum before and after the advent of CAD (Unver, 2006 p326)

2.4.5 Lack of foundational knowledge

Schoonmaker (2002), found that many CAD users do not fully understand the difference between basic IT concepts such as memory and disk for example, or the differences between vector graphics and bitmap images. Amongst other reasons, this knowledge is important, because it can help designers to translate files/models made in software from different vendors, evaluate different CAD systems, and maximize the benefits of CAD systems. The same is argued by Ye et al., (2004), when arguing that in order to fully master a CAD system, it is important to not only have knowledge about a CAD package, but also about computer technology in general (Ye et al., 2004).

In regards to mathematical foundations, Schoonmaker (2002) recalls, that one should not be misled to believe that because 3D is available 2D CAD is not important. On the contrary, mastering 2D geometry and 2D maths may be even more important.

2.4.6 Tool Domestication

There is also a complaint that CAD is simply used as a tool for modelling, but nothing else. Despite their versatility, designers use computers to do what they have always done, and tend to compare computerized applications to traditional ones (Chapman, 1995; McMahon & Browne, 1998 p.252; Togay et al., 2016).

2.4.7 Disregard/Scepticism of computer skills/technology

Scepticism around the use of computers persists in academia; Self (2012) for example, claims that Industrial Design should not aim to educate 'CAD Junkies/Monkeys,' and that computers limit the 'flowing, more reflective processes around sketching (Self, 2012). The same position can be seen from Ye et al. (2004) when saying, "CAD is just a communication tool for engineers" (Ye et al., 2004, p. 1458).

2.4.7.1 The computer is just a tool

The view that computer skills are not important because technology is just a tool persist in academia. This may play a role for example, on the fact that Dönmez (2013), found that most CAD courses in Industrial Design programs in Turkey are delivered by part-time lecturers. As Andrew Feenberg says however, just as economy has been seen as not philosophically important in the past, the deep implications of technology are slowly being recognized (Feenberg, 2012).

2.4.7.2 Scepticism and fear

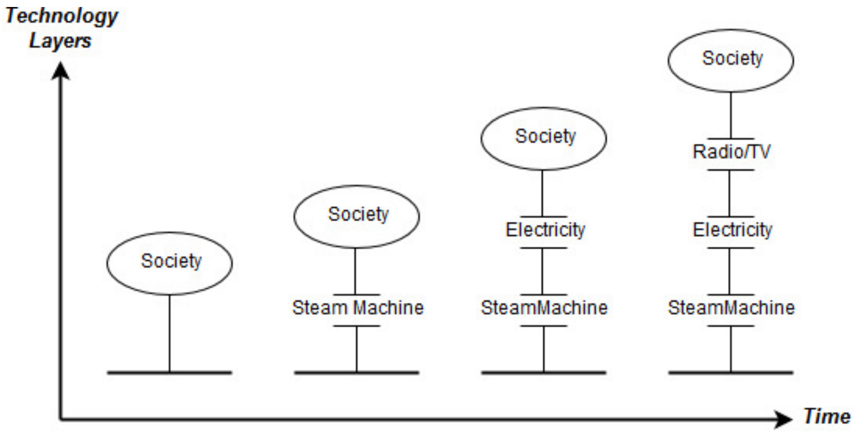

Figure 2.12 Illustration of the idea behind 'The Technologic Trap' (illustration by author).

Technology determines new technology, Technology is developed to control technology, and new technology relies on technology. All these factors together take us into 'the technological trap,' or what Feenberg (2012) calls 'the risk society.'

An example is the QWERTY keyboard, originally developed to solve problems with a technology–the mechanical typewriter–ended up determining the way we type with other more modern technologies that do not have the problems of the initial technology. We could actually be typing even faster now, but because the QWERTY keyboard was introduced, a whole host of newer technologies now are based on it. Surprisingly, as Keirl (2004), points out; we continue to have faith that technology will solve our problems.

2.4.7.2.1 Negative effects

Just as the notion that technology is no more than mere means to an end persists amongst some academics, so does the notion that the use of computers has a negative effect. Moreover, such fears are substantiated by a wealth of literature coming from the philosophers of technology.

CAD in general, has been extensively criticized for limiting the kind of design exploration that designers engage with during concept development. For long it has been debated whether CAD has negative effects on creativity. Consequently, a question that persists is when to allow students to work with CAD (Musta'amal et al., 2012). There is evidence that an early exposure to CAD 'conditions' students to the tool. This, according to Self (2012) happens because novice design students can be reassured by "the command-based affordances of CAD and dazzled by its ability to create slick, glossy images" (Self, 2012).

Chapman found that few students used computers as a creative tool but most used it only for presentation purposes. He acknowledges however, that it is unclear whether this is due the tools' actual constraints or whether this is due the students not reaching a level of expertise that would allow them to express their ideas fully (Chapman, 1995). De Biswas (2012) on the other hand, writes that CAD tools are *"closed toolboxes with prescribed languages / components and the universe of user engagement remains contained within such pre-structured spaces constraining ways of seeing and doing"* (De Biswas, 2012, p. 70). Similarly, Stiny and Onur (2012) say that parametric design tools *"tend to hide computational complexity,"* while still allowing users to generate complex forms (Stiny & Onur, 2012, p. 9). Joseph Weizenbaum on his part, claims that in most cases, computers end up inhibiting children's creativity, because *"the computer programs kids"* instead of other way around (Oppenheimer, 2003, p.38).

2.4.7.2.2 Dependence: The technological trap

Marshall McLuhan is well known for the phrase "First we build the tools, then they build us." Weizenbaum too explains this circular relation that we establish with technology; we first envision and create our tools, but after time those tools shape the way we can imagine our next stage of invention, and before long tools like computers are seen as indispensable part of progress. We ignore other possible alternatives to address the challenges we face. (Oppenheimer, 2003). Along the same lines, Verene (2013) on the other hand says that technology brings its own necessity with it, a fact that is often masked/obscured because discussions around technology tend to be biased/clouded by society's obsession with efficiency. Thus technology is often evaluated just in terms of *"what more can be accomplished by the new approach and how whatever remains can be made more efficient."* This principle of 'technology as tool' overlooks the fact that once a tool is assimilated, we lose the capacity of seeing how to operate without it. *"Technological advance is irreversible*" (Verene, 2013, p. 297).

George Stiny sees this as a particular problem with parametric design tools; *"people use these things and never bother to find out how they work. As a result, they get stuck—as you point out, with the kinds of designs that are merely "available". Their own visual intuitions are no longer important"* (Stiny & Onur, 2012, p. 10). However, existing research does not show whether the problem is CAD, or, early exposure to it, or even the way CAD is taught. For example, the dominance of a 'discrete' teaching approach. The result of this perception about CAD is the discourse that computers/CAD are good mostly for modelling and visualization. Unver (2006) reinforces this notion saying; "computers can only enhance a good concept" (Unver, 2006, p. 326).

Computers are neither neutral nor universal, like any other tool, computers constitute a medium, and the medium always has an impact on the product created through it. Self (2012) argues that; two different design tools can lead to two very different embodiments of design intent (Self, 2012). Luscombe (2018), provides a simple, but very powerful example to illustrate how different design tools imprint their character on the resulting design. Design tools and the products created through them maintain a symbiotic relationship. In discussing this relationship, Erkal (2012) notices that: "*What is designed and built shapes the* reference *frameworks of* what *is expected in the future."* Thus, what is designed in the future, is designed with tools which are *"limited in their own ways with the same frameworks that the society finds relevant"* (Erkal, p. 53; 2012). Computers open design mediums that have an impact on the design intent. These mediums and their corresponding techniques then, become the reference framework for how things are to be designed in the future.

Moreover, as computers have taken over much of the modelling process, it is unknown how schools are dealing with the long-term consequences of technological adoption. While the computerization of modelling brought with it the promises that often accompany technology–the possibility of doing more and better, a number of technology philosophers have warned that whenever a new technology simplifies a task, it does so at a price. The benefits associated with the old way of doing things are eventually forgotten, ultimately lost, and the only question remaining is if what is gained is more than what is lost (Bryden, 2014; Feenberg, 2012; Oppenheimer, 2003, p. 100; Verene, 2013).

Thus, there is a real danger in not assessing technology critically, and in overlooking the fact that technology is more than mere means to an end. Since by default technologies are neither neutral nor universal, there is always the possibility of a hidden 'technological agenda' behind technological change and those pushing for it. Only in recent years, the notion that technology is much more than mere means to an end, and that it affects the output of a design process in several ways has started to be acknowledged (Feenberg, 2012; Luscombe, 2018; McCullough, 1997; Togay et al., 2016). This will be explored in further detail in chapter two.

Computer modelling demands/proposes a different interaction involving brain, computers and hands. With their structure, CAD tools impose ideologies onto the design disciplines. (Schoonmaker, 2002; Yue, 2011). Tsamis (2012) argues that these tools precondition our perception for the simple fact that any form of representation favours certain aspects and excludes other (Tsamis, 2012, p. 48). A 2D model for example, leaves out the possibility of tactile perception. All tools frame our thinking, thus, all tools are thinking tools. (Luscombe, 2018) McCullough, 1997). If we 'think through tools,' then then fluency of students with their tools has a direct impact on their 'Design Thinking.'

2.4.7.3 Counter-arguments

Education is a complex social institution involving many actors and carrying with it a strong moment of inertia. In 1996, Seymour Papert recognized that the reaction of schools to the introduction of computers resembled that of any "living organism;" launching the "immune system" and defending itself from 'the foreign body' (Oppenheimer, 2003). That skeptic view of CAD has been changing over time, and in more recent years some scholars see CAD as another creative tool more than a hindrance (Musta'amal et al., 2012; Togay et al., 2016). Varinlioglu et al. (2015), say that the way students interact with computers during their first years, shapes the interaction for the following years (Varinlioglu et al., 2015). At the other extreme however, some scholars even suggest the use of traditional mediums can be completely skipped (Efer, 2017).

2.5 Chapter summary

The fact that CAD has penetrated Industrial Design by replacing drafting and then modelling has consequently promoted the vision that CAD is drawing and/or that CAD is 3D modelling at different points in time. This has effectively defined CAD in relation to the design tasks that have been computerized at those times. This definition then, tends to determine the future application of comptuers.

There is no agreement about the best computer modelling approach for Industrial Design. Different modelling approaches pose different ways of working/challenges, and it is important that students understand the nuances of each. While some scholars argue that, product development is done in solids, others argue that surface modelling is clearly more suitable for the way designers work, particularly at the beginning of the design process.

Overall, there seems to be an agreement among scholars, that when learning CAD, one of the most important things is to learn the methodologies or process to build CAD models. However, and although Industrial Design Education is based on Studio courses, when it comes to computer skills, these have been traditionally fostered following a ‘discrete’ approach.

Discourses around computers in industrial design education are varied; taking different philosophical positions about technology. One is that computers are nothing more than a tool, with little if any effect over design intent. Another is that exposing students to CAD too early has a negative effect over their ability to 'think outside of the tool' (Self, 2012). Existing research however does not show whether this is the result of the exposure itself, or of how this exposure is handled—the pedagogy followed to teach CAD. A consequence of this view, is that CAD/computers are only good for enhancing existing concepts, but concepts must be developed 'outside' the computer.

Since technology brings dependence with it, the relevance of computer instruction in Industrial Design schools goes beyond mere efficiency; there is no other way to work/do design. This is not only truth because technology tends to displace previous ways of doing things, but also because there are models that only exist in the digital/computer realm.

It is clear that the instruction of computer skills in Industrial Design education has been studied in the past, and that it represents a legitimate field of study. It is also clear however, that big gaps exist; a substantial amount of this research has been conducted from the perspective of engineering disciplines, yet, as the existing literature shows, the use/role of computers in industrial design has its own set of particularities that make it unique from other disciplines. A substantial part too was generated during the late 1990s and early 2000s, time when computers were widely introduced in design schools.

Moreover, most past research has been conducted from the perspective of CAD, not from the perspective of computer skills in general. Consequently there is still a lot about its instruction that remains unknown, starting with what is it exactly that schools are teaching, and how. Nor is it well known if issues identified in previous research have been addressed in modern curriculums. As it will be discussed in subsequent chapters, there are important developments in 3D modelling approaches and rapid prototyping that warrant the importance of asking if the computer skills that Industrial Design schools foster are appropriate.

Chapter Three

3 METHODS

Having reviewed the literature pertaining the subject of study, chapter three goes over the methodology and research approach used; from the theoretical basis for the selection of methods, tools and techniques, to the sources of empirical data, recruitment of participants, design of instruments for data collection, sampling techniques, and methods of analysis.

3.1 Aims

The study aims to acquire an updated overview of computer skills education in industrial design schools, identify knowledge gaps, and identify areas of opportunity/growth for improvement. It also aims to give an overview of important developments in basic computing education, 3D computer modelling, rapid prototyping and computer-mediated education, and to discuss their impact/implications over instruction of computer skills.

The studies also aim to make a series of recommendations on how to improve the computer skills of industrial design graduates. The intention is to look at the overall instruction of computer skills, including CAD. The aim of the study therefore, is not to prove whether a particular approach towards the instruction of computer skills is more effective or not, but to find out what are the approaches followed, and what are their advantages and disadvantages based on contextual changes.

This research therefore, studies a phenomenon with the intention of describing it rather than explaining it in terms of causes and effects. Research following this approach usually presents conclusions drawn by abducting reasoning. In this process, a theory is developed directly from an observation via a logical inference; 'If phenomena A, then B must be a cause.' There could be also other explanations for A, therefore concluding that 'B' explains 'A' could be a mistake. In consequence, in this process 'B' must not just be possible, but also the most feasible explanation for 'A.'

3.2 Paradigm

Grix (2010) acknowledges that different disciplines tend to subscribe to different research traditions (Grix, 2010). It is in general well acknowledged however, that research in the social sciences and the humanities does not typically subscribe to just one of the two main worldviews, instead both are often employed.

As opposed to the positivist approach in which much research is concerned with proving or disproving a hypothesis, research following an interpretivist approach does not aim to prove anything true or untrue, it attempts to describe a phenomenon rather than examining variables. Tiene (2001) points out that this research is also known as 'qualitative,' to differentiate it from the 'quantitative' data that is generally generated from experimental sources (Tiene, 2001 p46). This research approach does not consider the world as being one where truth is objective (Vishnevsky and Beanlands, 2004), instead, this type of studies, aim at understanding experiences and developing theory (Elliott, R., & Timulak, L. 2005).

As stated, this study aims to look into the instruction of computer skills in industrial design education. This positions it at the intersection of two fields/areas of investigation; Technology Education & Design education. As it is proposed in figure 3.1 however, the central concern of the study is the instruction of a particular set of technological skills–computer skills–within the context of a particular discipline; industrial design. Therefore, if there is any research tradition that would serve as a basis/reference for these studies, it would be found in the education research tradition.

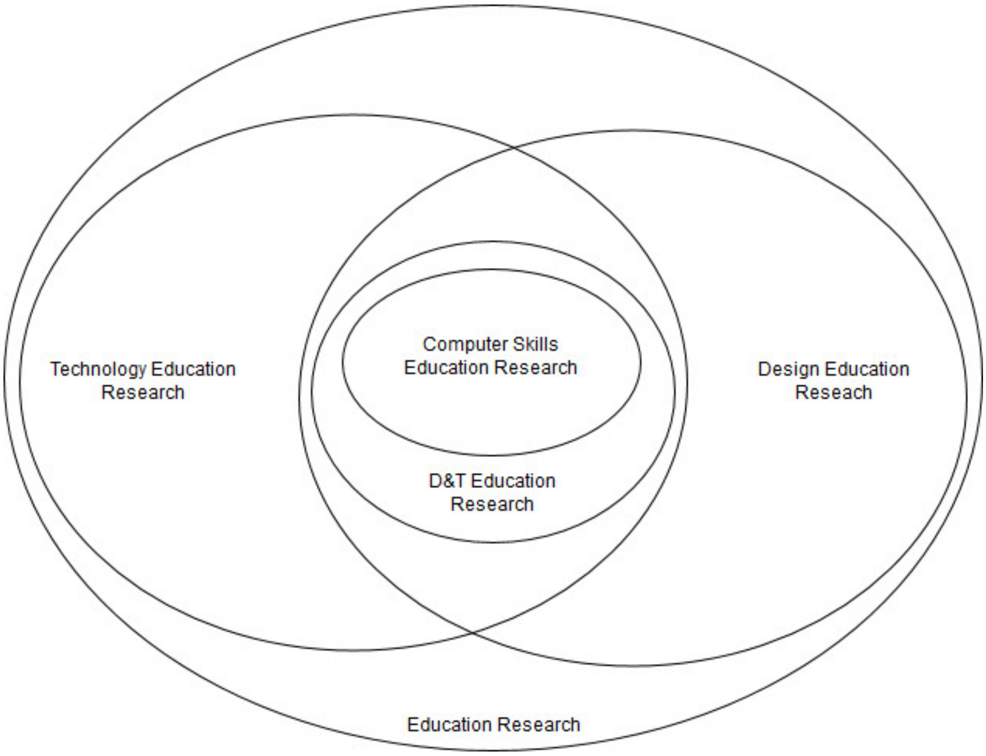

Figure 3. 1 Subject of study in relation to adjacent research fields.

Cohen et al. (2000) on the other hand, acknowledge that educational research has assimilated the two views of the social sciences; the established, traditional view and the more recent interpretive view (Cohen, Manion, & Morrison, 2000, p. 5). They also suggest that these views can then be examined through 3 kinds of methodologies: a) Scientific and positivistic, b) Naturalistic and interpretative, and c) Methodologies from critical theory.

Grix (2010) however, argues that the choice of research methods to follow, are to be determined first and foremost by the research question(s), rather than by the need to subscribe to any particular research tradition (Grix, 2010 p31). Similarly, Tiene (2001) points out that while each researcher tends to have a preference, which approach to follow should depend primarily on the aims of the research (Tiene, 2001 p49)

Moreover, scholar's advice young researchers to avoid spending too much time and effort discerning about research traditions (Grix, 2010 p124). Instead the aim for the new researcher should be to choose the best combination of methods to *"shed the maximum light on the chosen topic"* (Grix, 2010 p121). In addition, the demarcations among research paradigms are not as clear-cut sometimes they are presented, consequently the best research often takes pace in between two research paradigms. Consequently, an open mind is often more important than research training in different disciplines (Grix, 2010 p83).

While not subscribing to any particular research paradigm, these studies follow approaches that are associated with both research traditions; for example, the study employs all available sources of data, this is a characteristic of studies adhering to the interpretivist tradition. In descriptive research, everything can be treated as data, not just data emerging from formal sources like questionnaires and interviews, but also things like conference presentations, seminars, group meetings, music, films, photos etc. In addition, data from these different sources can be either qualitative or quantitative (MacDonald & Headlam, 2008).

These studies employ data from a wide variety of sources, such as notes taken during several seminar presentations that took place during the normal period of study of these PhD studies–August 2013 and August 2016–at the Hong Kong Polytechnic University such as those of Don Norman, Professor Rinus Roelofs, and Jiayi Young. The study also makes use of feedback obtained during conversations held during study trips and job interviews.

Another aspect of these studies that is commonly followed in research adhering to the interpretivist tradition, is that instead of starting with a fully defined research question, the study started with a question that only served to identify the phenomenon of study, to a more focused question (Willing, 2013 p72). MacDonald & Headlam for example, say that descriptive research can start with an open question like: "what about this phenomenon that interests me?" (MacDonald & Headlam, 2008).

3.3 Methods

Considering the previous arguments, the selection of methods was based primarily on the ability to obtain the desired information to answer the research questions. This, was considered to be possible by obtaining curricular information from industrial design study programs in higher education. Curriculums are an important source of information because, as Murray & Perez (2014) point out; *"the clearest articulation of an institution's approach to digital literacy manifests in its curriculum"* (Murray & Perez, 2014). Such information can be obtained from a variety of sources as well; such as educators, program coordinators, students, documentation, and student works. Similarly, and in consequence, this information can be obtained using a variety of methods; such as documentary reviews, surveys, interviews, and artefact analyses.

3.3.1 Approach

Using a variety of methods–often referred to as *'triangulation'* (Denzin, 1978)–has several advantages; Creswell (1994) for example, notices that a mixed-methods approach can include methods within a qualitative or quantitative paradigm–within methods–and also methods from both traditions–between methods (Creswell, 1994). Similarly, Shneiderman et al. (2006) recall that most scholars researching educational technologies agree that no single method is adequate if used in isolation, but that convincing results emerge when multiple methods are used (Shneiderman et al., 2006).

There is similarly, a consensus that using different sources of information and using different methods to collect information from those sources minimizes/reduces the risk of drawing false conclusions. Grix (2010) for example says that one way of avoiding drawing false conclusions from empirical data is by using more than one method of enquiry (Grix, 2010 p125)

Collecting data from more than a single source allows the 'triangulation of data' and adds validity to the study. I can also yield additional information (Tiene, 2001; Booth, Colomb, & Williams, 1995). A mixed-methods approach can also prove beneficial for young researchers due the fact that they may not yet have developed a clear philosophical worldview, and consequently an ontological and epistemological stance of their own.

Collecting information from different sources and methods is particularly important when doing research in the social world. Pierre Bourdieu makes a clear case when illustrating his theory of the 'social field.' From one perspective, this field will look different from another. Therefore, it is crucial to collect information 'from different corners of the field,' only that way it is possible for the researcher to have a better idea what the field actually looks like.

The collection of data for these studies was carried out following such a mixed-methods approach described above. These methods included a survey among industrial design educators from a number of institutions around the world. While curricular information could be obtained from documentary sources, in order to obtain up-to-date and detailed information, it was necessary to obtain this information directly from institutions. In addition; while some schools have part of their curriculums available online, sometimes only list 'core' courses are listed, in addition, they information provided is often superficial, mostly just a list of courses. Therefore, it was determined that this information should be obtained directly from institutions/educators. Similarly, while curricular information could be—at least partially—obtained from students, due time and resources' limitations, and the complications to easily access students from a wide range of institutions, it was decided to focus on educators.

In addition to surveys, the studies also rely on interviews with industrial design educators and experts from industry and academia. According to Grix (2010), interviews can be particularly enlightening when used in combination with other methods of inquiry. In addition, he notices that surveys are most effective when used in conjunction with other methods, particularly interviews (Grix, 2010 p129). The data from interviews provides a different type of insight that enriches and frames the subject being investigated, a 'texture' that is unlikely to emerge just from statistical data (Grix, 2010 p121). Areas of expertise that were identified as areas of specialization of interviewees included: computer aided design, computer programming, generative design, and additive manufacturing.

The studies also use the review of student works and reports, and the experience of this author as industrial/product designer, and lecturer in the US, China and more recently the UK. Consequently, the experience of this author as industrial/product design student, practitioner, and lecturer in different countries is a valuable reference to draw from. This experience can provide anecdotal data, as well as an archive of student works that serve as a reference.

Figure 3.2 shows the main methods of inquiry used in this study. The difference in size represents the relative relevance of each method in terms the input provided. These main research strategies/methods are enriched/complemented with other research such as study looking at the level of digital literacy of higher education students. In addition, the study relies on other approaches that are intrinsic to almost any research project. For example; while this is in essence not a comparative study, Grix (2010) acknowledges that it is very difficult–if at all possible–to do research of any kind without resorting to making some kind of comparison, whether the study is officially as a comparative study or not (Grix, 2010 p53).

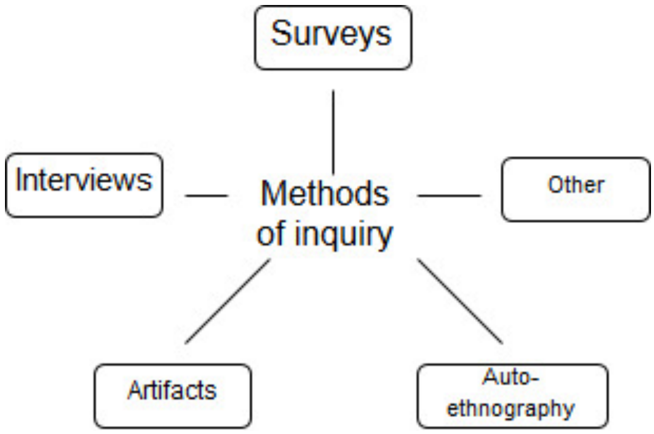

Figure 3. 2 Main methods of inquiry used in this study

3.3.1.1 Preliminary studies

The first part of this research consists of a series of preliminary studies, the first of which was a study to corroborate whether the line of inquiry held promise, identify any wrong assumptions and streamline the research questions. Data for this preliminary study was collected through a very short questionnaire. This initial study only sought to probe the use of computer software amongst: practicing product designers and design educators. The aim was to get a first-hand impression and fresh insight as to how practicing industrial designers and educators understanding the role of computers in their professional practice and what they understand by Computer Aided Design. Practicing Designers were asked to list all the software tools they use for work. In particular, they were asked to think of all software, not just 'typical' design tools. Participants for this initial pilot study were recruited using Linked-in as pool of participants. More information about this initial study and/or its findings can be reviewed in appendix IV

In a second preliminary study, design schools from different countries were asked whether they incorporated any kind of computer programming instruction as part of their study program. This study also sought to know if they did, how they implemented it; what were the programming languages being taught, and how were those integrated in the curriculum. It also sought to know the position of faculty regarding the need to incorporate computer programming as part of the technology education that product design students receive as they transition through college. This pilot study served as a reference for the section about computer programming in the main study, and for questions regarding computer programming in the interviews. For more information about this study please read (Contreras & Siu, 2015).

A third study conducted in this initial stage, consisted of a survey that was distributed amongst over one hundred product design engineering students of the Hong Kong Polytechnic University. This study aimed to looking at the level of digital literacy of higher education students. For more information about this study please read (Siu & Contreras, 2017).

3.3.1.2 Workflow

A schematic representation of the workflow of the research is presented in figure 3.3. The output of the research was constantly used to do more literature review where necessary, and to adjust the focus and scope of the research questions.

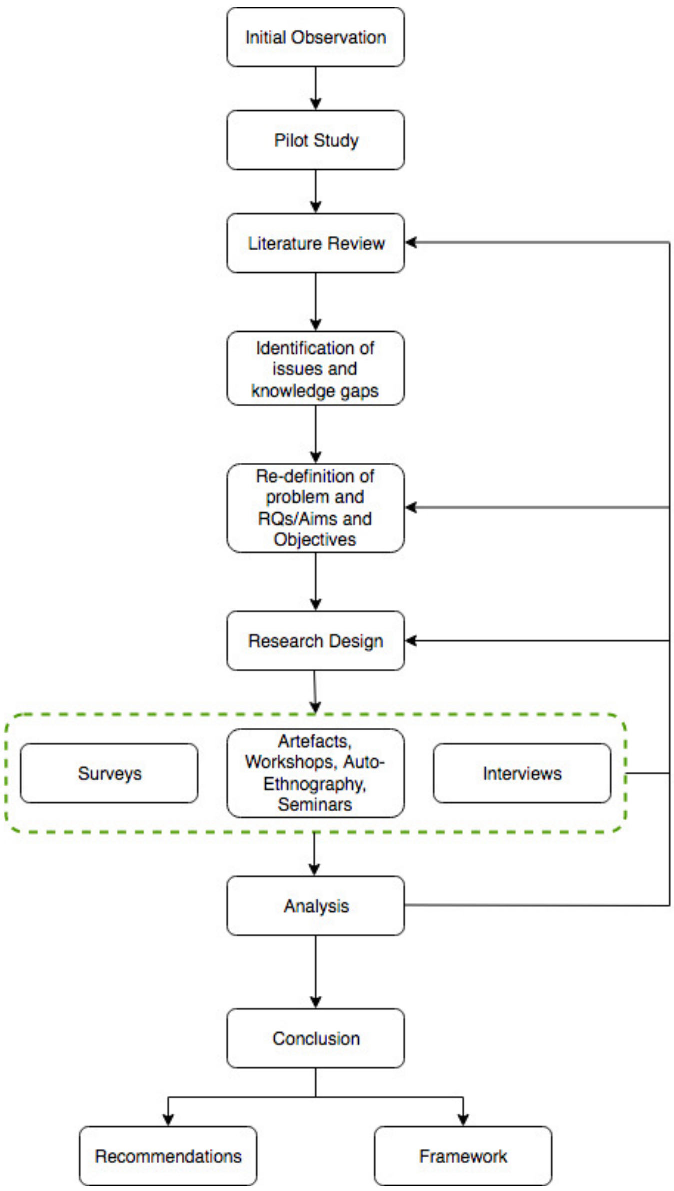

Figure 3.3 Schematic of the Research Workflow

3.3.2 Data Collection

The study relies on surveys and interviews as core methods to collect data/information. Surveys are one of the most common and versatile research methods in all fields of research. Frankel et al. (1996) say that the main purpose of surveys is to "describe the characteristics of a population." In this case such population is comprised of the industrial/product design schools sampled/studied (Fraenkel & Wallen, 1996, p. 368). With surveys it is possible to gather qualitative and quantitative data.

Tiene (2001) also says that surveys can be used to obtain a variety of information using different types of questions, such as asking participants to rank and order a set of items (Tiene, 2001 p47). In addition; surveys can feature open-ended questions, allowing this way to allow participants to express themselves more like in an interview setting, but with the added advantage of not having the interviewer present, which has the danger of biasing the interview. Using open-ended questions makes it possible to obtain more varied and reliable data (Tiene, 2001 p47).

3.3.2.1 Surveys

Since some of the questions raised in these studies can be best answered using quantitative information while some other can be best answered using qualitative information, the survey sought to acquire both. As has been seen previously, using different methods to collect information from the same sources has several advantages. For example; while inquiring about the computer skills that industrial design schools foster was taken as a quantitative endeavour, knowing the opinion of academics about the need to incorporate computer programming in the curricula, was approached with a quantitative strategy in mind. For this reason, the latest were mostly open-ended questions.

The survey aimed at collecting detailed information of courses related to computer skills in industrial design schools. The questionnaire was divided in four different sections; one aimed at gathering general information about the person/educator answering the survey–location of institution, type of appointment, gender etc., Another aimed at collecting information about the particular program and the institution. One more aimed at collecting information about the traditional set of computer skills taught at the institution. Another section seeking to know if the institution teaches computer programming or not, and if so, what is the approach followed. And lastly, one section looking to know whether the institution teaches any form of generative design methods, and if so, what is the approach followed.

The questionnaire was organized along three main lines of inquiry; whether product design students are learning computer programming–or if schools are teaching it. What kind of computer skills are industrial design schools fostering, and how are those skills being fostered–discrete or integrated approach. Whether industrial design schools are learning generative design methods. What is the opinion of faculty in regards to these previous' points.

The questionnaire was integrated by a variety of questions open and closed questions; the latest were mostly single-answer multiple-choice using radio buttons, and multiple-answer multiple-choice using checkboxes. With the exception of open-ended questions, the aim was to gather mostly quantitative data. On the other hand, most open-ended questions in the survey provided qualitative information, along with the interviews, this helped to 'tone' and enrich the statistical information gathered with the rest of the survey. This questionnaire also served as a basis to determine the questions to be used in semi-structured interviews conducted with experts.

The questionnaire was implemented online using Google forms. While there are many platforms on which online questionnaires can be implemented, Google forms was selected because; the service is free and it offers the best combination of features among other free options. In the case in which an institution was located in China, and Google forms were not accessible, questions were sent by email, or sometimes the participant had access to a VPN service that allowed them to access the questionnaire.

Online questionnaires have many advantages, one of them is to be able to work with large samples at low cost, or even no cost at all. With online questionnaires it is possible to reach large populations in distant locations that would require much more resources if conducting the surveys in person. Another substantial advantage of online surveys is the ability to control the flow of the questionnaire using filter or contingency questions. This way a participant can be re-directed to a question ahead or even back using an 'if-then' type of logic. This makes it possible to skip entire sections of the questionnaire that may not apply to a particular participant, but which may apply to another. In addition, online questionnaires can encourage participants to open and share their insights more freely due the degree of anonymity provided by the computer-mediation.

A copy of the online questionnaire can be reviewed at the URL: http://bit.ly/2uzd32M

3.3.2.1.1 Sampling

Participating institutions were identified using a variety of sources; such as online directories of industrial/product design associations like that of the Industrial Designers Society of America–IDSA, school associations such as the World Design Organization, Cumulus and from the author's personal knowledge. Invitations for surveys were sent by email. When the specific person of interest–CAD teachers and/or course leaders–could be identified and the contact information of the person was available, the invitation was sent directly. In the majority of the cases however, this was not possible; either the teaching assignments of each teacher were not specified online, or their contact information was not available or both. In these cases, an email was sent to the person or office that seemed to be the best option, and the request was made to obtain the contact information of the person in question.

Schools were chosen on a convenience-sampling basis, but being careful that all participating institutions fitted the target: being an industrial or product design bachelor's degree study program. And procuring to include a more or less balanced number of institutions from different parts of the world. The objective of the study was that the sample could provide an overall picture of the general situation in industrial design schools.

Crouch & Pearce (2012) say that when data collected is expected to be made generalizable, the sampling is usually made so that data from large populations can be gathered (Crouch & Pearce, 2012). Similarly, Coughlan et al. (2007) say that one of the most important consideration about the sample size in quantitative studies is whether it can represent the population it was drawn from (Coughlan, Cronin, & Ryan, 2007).

Another factor taken into consideration for the selection of schools was the language of the country where they are located. As far as the specific participants within each institution, the main targets were teachers' lecturers of computer-related classes and or course leaders/coordinators. The total number of participating institutions was 45, this represents around

50% of the total of institutions that were approached for participation. The list of participating institutions can be reviewed in Appendix I. The complete details about participating institutions can be reviewed in appendixes I and II.

3.3.2.2 Interviews

Interviews are one of the most common method of data collection (Coughlan, Cronin, & Ryan, 2007) (Burton, Brundrett, & Jones, 2008). Semi-structured interviews allow the exploration of tangential topics during the interview process. In addition; interviews make it possible to collect more in-depth data about a given issue (Tiene, 2001 p47). Grix (2010) notices that semi-structured interviews allow for *"the pursuit of unexpected lines of enquiry."* While the results obtained can still be analysed using a variety of formal qualitative and even quantitative methods (Grix, 2010). If the same set of questions is going to be asked to all participants, then the degree of structure of the interview should be higher (Tiene, 2001 p47).

Feedback from experts was done through semi-structured interviews; however, a small number of unstructured interviews were conducted with some experts. Semi-structured interview provided the right degree of flexibility with most experts. As the interview evolved/unfolded, interviews went into different directions, exploring a variety of topics were explored in greater depth. For example; when an interviewee was a non-academic in generative design methods, questions focused on generative design and computer programming, leaving aside questions about teaching.

Interviews took place in a variety of ways; whenever a face-to-face interview was possible, some of them were interviewed using video-conference tools such as Skype. Others preferred to have a set of questions sent by email, once they replied with their answers, additional questions were asked if their answers needed clarification or if their answers opened the door to inquire more about a certain aspect. This exchange of emails can go back and forth a few times until it was deemed necessary and/or possible.

In a few other cases, experts were found and consulted over specialized internet forums. Internet forums offer the same basic functionality of an email service, with the only differences that conversations can be publicly available, and if this is the case, all members of the forum can participate in any conversation. This was a useful way to get in touch with experts in the field of generative design. Many of whom are leaders of forum groups, and thus are responsible for addressing the questions that are risen in the forums.

Since the data from interviews is not intended to be generalized, Crouch & Pearce acknowledge that samples tend to be usually small and are selected purposefully (Crouch & Pearce, 2012). Participants for interviews were chosen based on how relevant their expertise was for the study. The target group consisted of academic and industry experts in computer aided design or other fields or relevance for these studies (figure 3.3).

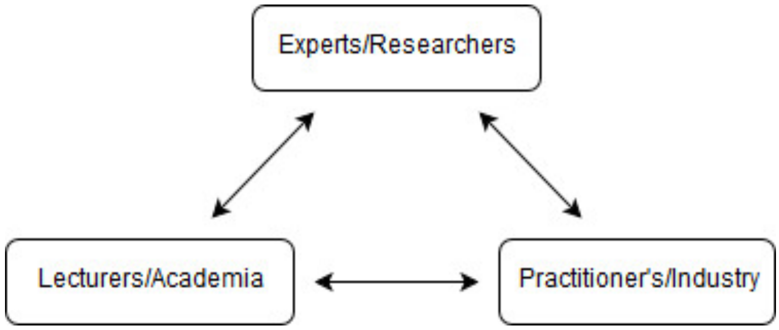

Figure 3.4 Interview Target Groups

Interviewees were identified using a variety of methods; literature review, the press, personal connections, and previous' knowledge from the author. Still, a few more interviewees were identified through interviewees themselves, who suggested to approach other individuals. The list of interviewees is included in the individuals consulted for these studies which can reviewed on Appendix III

Doing research in the same professional field to which the author belongs can be advantageous, as Valtonen (2007) points out, "Belonging to the research group can help the researcher bridge many issues of a technical kind, such as a hostile attitude towards intrusive strangers or issues of professional language" (Valtonen, 2007 p48).

3.3.2.3 Other sources

Tiene (2001) says that artefacts can help researchers understand better the subject of study. Artefacts can represent the outcome of a learning experience which then can be evaluated (Tiene, 2001 p48). In this study, artefacts were integrated by student works collected using a variety of methods; from school displays observed during study trips, works obtained as a result of the author's previous' teaching, learning and research experience.

Other studies included two workshops in which participants had to use scripts with adobe illustrator to generate a pattern/texture that could then be laser-cut, and two seminars with a group of first-year product design students using a simple generative design tool to produce a simple object. In a second workshop, a group of 10 first-year product design students took part in a 2-hours workshop using Autodesk generative design tool 'Shapeshifter' to design something that could be a flower pot. More information about this workshop is presented during the discussion in chapter six.

3.3.3 Analysis

Since the combination of methods used on this study yielded both; qualitative and quantitative data, both had to be analysed separately.

3.3.3.1 Units of analysis:

Grix (2010) says that units of analysis can be individuals, groups, organizations, categories and institutions. He also notices that using different levels of analysis–a multilevel analysis–can provide "different lenses" through which a phenomenon can be seen, and thus a richer insight (Grix, 2010 p48). The units of analysis of this study are institutions. However, the units of analysis for the interviews part of the study are individuals. This means that the level of analysis for the first will be at the macro-level, while for the second it will be at the micro-level, a practice that is common in the humanities.

3.3.3.2 Surveys

The analysis of information gathered with the survey (except open-ended questions) was done using descriptive statistics and inferential analysis. Inferential statistics help to make decisions based on how different variables relate to each other. This process can be done using 'contingency tables' in which Rows and Columns represent different variables, and what appears in the table's cells, is the actual count of 'individuals' or cases that fit that column and row. This information can be used for the purpose of drawing conclusions or generalizing. Gibbs (2013) says that "we can infer some significance based on the values we got" (Gibbs, 2013). These tables are good for recognizing patterns.

Husser (2012) says that for this process it usually makes sense to have just a few values. Therefore, nominal and categorical variables are commonly used. Variables in this process can be nominal or categorical; that is, variables in which results are sorted into categories. Since most of the questions–excluding open-ended–were multiple-choice questions in which answers were in the form of pre-defined texts; the initial computation of data was done in the computer. This computation was done in Microsoft Excel using 'text-string' counting formulas. In the case of questions in which participants had the option to enter non-standard answers–labelled 'other option'–this computation was done using formulas with wildcards, which make it easier to match partially similar, but not identical text strings.

3.4 Chapter Summary

In this chapter, the research approach has been presented. The methodology followed in the studies has described in greater detail; from the theoretical basis for the selection of methods to the design of the instruments for data collection, The advantages and disadvantages of each approach and the tools and techniques used have been discussed. The survey was based on an online questionnaire featuring multiple-choice and open-ended questions, and a response of 38 different institutions was received. It targeted all type of institutions, so long they offered Bachelor's Degree in either Industrial Design, or Product Design. In the case of the interviews, the interviewees had different profiles: some were experts in Generative Design, some were academics, some more were practitioners or entrepreneurs, all related to Computer Aided Design and/or Industrial Design.

For the workshops and the seminars, the mechanic was similar; participants were introduced to Generative Design first, and then they worked on a hands-on project, in the case of the workshops, participants used Adobe Illustrator scripts to create an artwork, which they then laser-cut, and built a small container box. In the case of the seminars, students used Autodesk's Generative Design Tool 'Shapeshifter' to create a simple 3D model which could serve as the basis for the design of a lamp shade or a flower vase.

Chapter Four

4 CONTEXT

Having established the research methods used in the study; this chapter looks at the context in which the research questions are being raised in more detail.

The information presented is based on a mix of secondary and primary sources—pilot study. The chapter looks at four trends that have an impact on the instruction of computer skills; the rise of Computer Programming and Computational Thinking, the rise of Computer-mediated education, the rise of Additive Manufacturing, and the rise of Generative Design.

4.1 The Rise of Computer Programming

Due the exponential growth of technologies based on or relaying on software, one of the key skills identified as a basic literacy for the future is computer programming. As more and more technological innovations are based on or rely on software, having a basic understanding of how computer programming creates the world we interact with every day is becoming a fundamental digital literacy, because Computer Programming allows people to move from being just consumers of technology to become its creators as well (Pearce, 2013; Loukides, 2014; DeLoura & Paris, 2013).

4.1.1 Computational Thinking

Wing and others argue that Computational Thinking will be a skill that every person will need by the middle of the 21st century. In support of this argument, it is said that Computational Thinking is already having a strong impact in all disciplines (Guzdial, 2008; Wing, 2008), the most widely cited example is the sequencing of the human genome. A fundamental aspect of computational thinking is abstraction, the ability to know how to translate life phenomena into information that can be computed. Computational Thinking has gained much attention in the educational arena in recent years. For example, a new basic computing education curriculum was unveiled in the United Kingdom in 2013, precisely with the aim of fostering computational thinking skills and moving away from simply teaching students ICTs. (Berry, 2015).

The term Computational Thinking, was coined by Computer Scientist Jeannette Wing, and refers to the ability to think by drawing from fundamental concepts of computation. It is a type of analytical thinking sharing some characteristics with mathematical thinking, engineering thinking, and scientific thinking. Selby and Woollard, 2010 define it as: a focused approach to problem solving, incorporating though processes that utilize abstraction, decomposition, algorithmic design, evaluation, and generalizations.

Computational thinking also introduces the concept of layers (Wing, 2008). In the case of computational thinking, these layers are in terms of software, however, this process can help in thinking about layers of technology. The interdependence of those layers and consequently about issues related to technological dependence and avoiding technological traps.

In the quest towards computational thinking, coding is the vehicle to get there. The practical experience of programming is almost certainly the best way for primary pupils to learn about computer science (Berry, 2015). Guzdial (2008), recalls that visionary Alan Perlis realized that programming was a necessary step toward "understanding a theory of computation which would lead students to re-casting their understanding of a wide variety of topics in terms of computation" (p25).

4.1.2 Computer programming

Wen computers were first introduced to schools not much software was available for purchase, so schools used to teach how to write software in languages like BASIC. This was positive because it helped kids to see computers more as a creative tool; "whatever you wanted it to do, it would do it—if you learned how to speak its language" (Thornburg, 2013). However, as ready-made computer software became widely available by the late 1980's, word processors and computerized spreadsheets took a central role and made of computer science a less important aspect of computer literacy.

> Learning management programs, such as Blackboard, are being used as a general panacea by higher education disciplines as a means of satisfying technology outcomes. What has been overlooked is the difference between use of ICTs as teaching tools and the development of ICT skills in graduates (Duncan-Howell, 2012).

Michael Starling made the following observation: *"there is almost universal acceptance and use of application software, so there is a strong 'user' community without any strongly developed appreciation of the potential for software development"* (Starling, 1999). Dan Crow says that "somewhere in the mid-1990s we lost our way" asserts Dan Crow, the education system did not pay attention to the expansion of computing and the internet, and instead remained focused on teaching students how to write Word documents (Crow, 2014).

The same is true in higher education; during the 80's and even early 90's not much software was available, and using a computer required of a higher degree of computer science literacy. Industrial Design students have far more tools today than 20 years ago, but at the same time it is uncertain how many of them are capable of using the command prompt.

Computer programming has found a place in the education curriculum of other creative disciplines. However, the need to incorporate computer programming into the Industrial Design curriculum has not been discussed extensively. Norman (2010) and others acknowledge that computer science skills have much to offer to designers to be left out of the curriculum (Norman, 2010).

Candy (1997) for instance notices that;

> Access to the full computational power of a system may be obtained by learning the programming languages that drive them; but, for the

> ordinary non-computing specialist user, these languages are arcane and, thus, the extent of that power remains highly circumscribed (Candy, 1997).

Today it is hard to deny that her vision was correct; the days in which access to sophisticated computer software was a competitive advantage are gone.

A study conducted at Carnegie Mellon university found that many people in fact do programming, without being aware about it. They do so when doing things like creating macros or database queries, however, not having proper foundations about coding however, makes them struggle (Shein, 2014). Research too can benefit substantially and often involves writing code, because no software exist that can do what the researchers need. Thus, computer programming has been regarded as 'the literacy of the future' (code.org), a skill that should become a new component of the 'new cannon' of studies, a kind of knowledge that every person should have at the 'dawn of the information age' (Burleson, 2005; Marucs & Davis, 2014; Wagstaff, 2012; Shein, 2014; Henn, 2014; Shapiro & Hughes, 1996; Trovalds, 2014).

4.2 The Rise of Computer-mediated Instruction

Experts agree that around 70% of what we learn in a lifetime is mostly learnt informally. Although most people think of school as the place where one goes to learn, formal education is neither the most natural way of learning nor where individuals learn the most during a lifetime (Jeffs, 2001, p. 46). Informal learning is as, or even more important than formal learning (Bennett, 2012; Schugurensky, 2000; Siu & Contreras, 2015; Wing, 2008).

The current model of education was developed during the 18th century, and at least partially due to its convenience for industrialists (Gonzales, 2012; Doin, 2012). However, before the introduction of formal education as the 'de facto' form of instruction, people learned informally; boys learnt to do farm work and other tasks by helping their parents and eventually became apprentices of a craft if they had the possibility of working for someone who could teach them (Jeffs, 2001).

According to Schugurensky (2000) and Livingstone (1999), formal education is the education provided through official institutions and which is usually mandatory up to a certain level. Informal education on the other hand comprises all other forms of education outside the official system but still involving people acting as tutors and students, and also some kind of curriculum. Informal learning is all other learning that takes place outside of both; the formal education system and any form of informal education.

Based on this understanding, informal learning can be classified using two criteria: intentionality and consciousness (Schugurensky (2000). This results in four different categories presented in figure 4.1 (Schugurensky, 2000; Bennett, 2012).

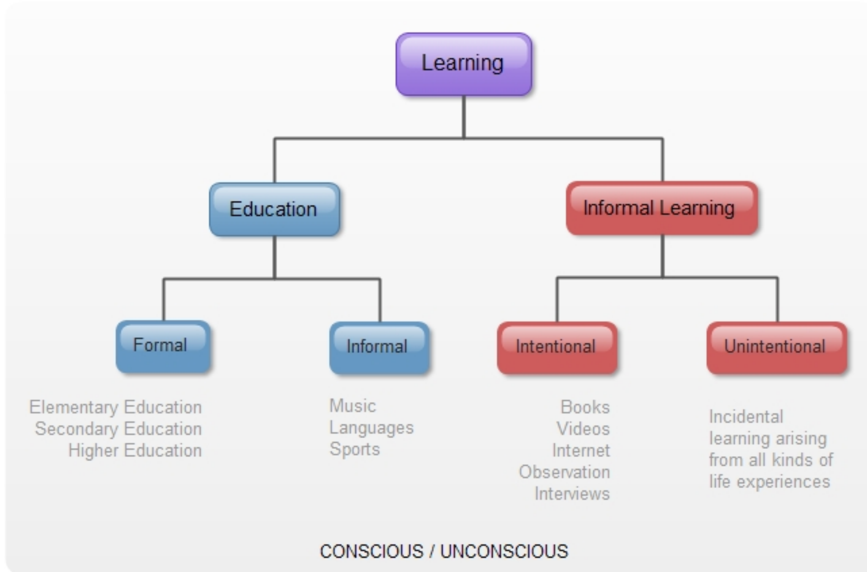

Figure 4.1 Formal and Informal Learning Scheme (Siu & Contreras 2015).

If there is one way in which technology has greatly impacted education is in democratizing access to information. Experts outside academia and even hobbyist often provide better information and in better formats than Universities. With the availability of courses from some of the most-prestigious universities and scholars in online platforms like YouTube, Coursera or edX, the content that Universities can offer continues to grow increasingly limited. The speed at which universities can respond to changes in such a rapidly transforming world puts them in considerable disadvantage; by the time study plans are proposed, university committees put together, analysis are made and curriculum changes are finally approved, online education providers can update their courses several times, leaving Universities are left again far behind (Siu & Contreras, 2015).

Similarly, Stallard and Cocker (2001) predicted that by 2014 'fixed' learning environments would have disappeared in many communities and learning from printed materials would be the exception. Neither has happened, yet, these predictions may have just been premature, not necessarily inaccurate. With the rapid development of information and communication technologies at the beginning of the 21st century trends like 'Cloud Learning,' 'Mobile Learning,' 'e-Learning,' and 'MOOCS' can indeed have a profound impact on education. Both; philosopher Patrick Suppes and computing education pioneer Seymour Papert forecasted that computers would "blow up the school" (Oppenheimer, 2003, p. 20).

With the current state of ICTs, the collection of large amounts of information is possible. With '*Big Data*' it is possible to learn much more about students than ever before–learning styles, habits, obstacles etc. This information can then be used to develop tailored study plans, through which students can learn at a pace and in patterns that the traditional education cannot offer. Morevoer, Big Data can have a strong impact on how students are evaluated, since computers can automatically track their progress as the course progress (Tiene & Ingram, 2001; Koller, 2012; Stallard & Cocker, 2001).

With the advent of ubiquitous computing technologies, ICT-mediated instruction is also versatile, as learners can review their lessons anywhere any time (Noonoo, 2012). Mobile learning is said to hold a great disrupting potential for education because it frees people of the constraints of having to be in a fixed place (Vanden Heuvel, 2013).

ICT mediation can be used to implementing what J. Cross (2007) calls 'learnscaping;' setting up environments to encourage spontaneous interaction between students (Cross, 2007). This shift from ICT to computer science is unlikely to be a temporary fashion. It is the result of a concern that has been building up for decades among/amidst the files of academia, particularly advocates of technology education, some of whom have expressed their frustration with the education system for being slow to implement much needed reforms, (Oppenheimer, 2003; Buckingham, 2007; NooNoo, 2012). Wagstaff (2012) for example writes; "take a look at the curriculum of many classes labelled computer science today and you'll find not much has changed from the days of dial-up modems," and only cover the basics, such as learning how use Word and PowerPoint (Wagstaff, 2012). The same view is shared by Ritz (2011) who acknowledges that "tradition has led many educators to teach technical expertise" (Ritz, 2011).

Similarly, Peng (2006) Computer-mediated education offers many advantages for Industrial Design, for example; easy to update study plans, enhance communication with students, sharing of information, emphasis of certain topics/information, help to realize the potential of different kinds of students

4.3 The Rise of Generative Design

The relatively new concept of *computational creativity* has emerged to encompass the capacity of computers to be creative (Colton, Wiggins, & others, 2012; Burleson, 2005; Boden M. A., 1998). An easy way to understand computational creativity is to simply think of it as the marriage of artificial intelligence with the study of creativity and the creation of creative systems. And has the goal of modelling, simulating, or replicating creativity using a computer (Association for Computational Creativity, 2014). This has led to the field that now we know as Computational Creativity branching out from Artificial Intelligence in order to study these and other areas at the intersection of AI and creativity (Lopez De Mantaras, 2013; Colton, 2009; Colton, Wiggins et al. , 2012).

A more elaborate definition is given by Colton et al. (2009): They say that at its heart, computational creativity is the study of building software that exhibits behaviour that would be deemed creative in humans. Such creative software can be used for autonomous creative tasks, such as inventing mathematical theories, writing poems, painting pictures, and composing music. However, computational creativity studies also enable us to understand human creativity and to produce programs for creative people to use, where the software acts as a creative collaborator rather than a mere tool (Colton, Lopez de Mantaras, Stock, & others, 2009, p. 11).

Until recently, the development and application of computational creativity systems that could potentially produce outputs valuable in industrial design was experimental, therefore only the creators of these systems themselves could experiment with them. This has changed over the past few years. Although the spectrum of tools to do generative design is still limited, there are some widely available options via open-source tools.

4.3.1 Generative design

Up to now, output of generative design methods has been very difficult to use in product design partly due to the manufacturing constraints present in mass-manufacturing. Traditionally, industrial designers have to consider how a product design can be manufactured and design accordingly–a process called design for manufacture–and for products featuring plastic housings/carcases, this means that designers must consider the restrictions imposed by mould-based production processes (Bryden, 2014 p138).

Generative Design represents a different approach, one where human and computer tool 'collaborate' in a different way. According to Krish (2013) the future of CAD will rely heavily on this kind of approach (Krish, 2013). With Generative Design the designer does not just model what she/he had envisioned previously in the computer. The design concept is mostly defined by the designer previously to starting any computer work. With computational/generative design techniques the computer becomes an active 'participant' in the process; the designer may still be using the computer to create a model, but he or she would have only a vague–if any–idea of what the final output/model will look-like. Generative design techniques can be applied at the beginning of the design process to inspire it in a variety of ways.

There are several reasons why this approach is not as popular in industrial design as it has been in other fields; In product design we have different constraints than in other creative fields, and one is the way people interact with a product for example, is different from the way they interact with a building or the way they interact with a graphic or a video game. The other very important difference is that we work with productions of scale, and at the moment these outputs, are either, very challenging to apply to products which can be mass-produced, or it is very expensive to do so because of the complexity they involve.

Therefore, the implementation of generative design tools has often been limited in scope. Design is complex. Consequently, the development of design automation tools has been slow.

> Many generative design tools developed to date are restricted to a limited number of design issues, or to a portion of the design process. In many cases, these can be considered 'toy' applications. Some were developed as teaching aids. Others have served to demonstrate proof of concept, with the potential of being a more powerful design tool left for the future. Simple generative design applications can be useful for teaching basic concepts to beginning design students (Chase, 2004 p689).

Fields like graphic design, architecture or digital-design, do not have these types of constraints, thus it's been easier to use the outputs of computational methods in these fields. Not surprisingly, the use of these methods has been more prominent in these disciplines. Consequently, some of the latest developments in this area have occurred either within these fields or with these fields in mind.

In a way, product design is in a similar position to the one Architecture was 10-15 years back, when industrial designers had more powerful 3D modelling packages that allowed the work with organic shapes for cars or appliances– let's remember that it was the transportation and aerospace industries that pioneered the development of NURBS modelling. At the time the most prominent CAD package in architecture was AutoCAD. So, Architects were 'forced' to 'cross the bridge' and appropriate some of these tools and then apply them in their professional practice. Rhinoceros for example did not become a popular 3D modelling package in Architecture until around 2006 onwards. It is also after this time that the flamboyant architecture of Zaha Hadid for example, starts to become prominent. But architecture as an older discipline, with a longer research tradition had something else, perhaps more valuable; they had started assimilating a different view of computers in design practice. Now industrial/product design is in a similar position when it comes to generative design methods and techniques. The power of current CAD systems to express design intent has boosted interest in the application of computer tools in the design process. This has empowered designers to create highly innovative forms which in turn has renewed the interest in generative/computational design and the production of such forms algorithmically (Chase, 2004 p690).

Terminology/ Types of computational design

Generative design and other forms of computational creativity have been pioneered by people from the arts, artists turned into computer scientists themselves (Boden & Edmonds, 2009). Generative design then can be seen as a modelling approach of a 'higher level.' In generative design the creating system has some 'capacity of agency,' that allows the computer to 'take some responsibility' over the process. According to the classification offered by

Boden and Edmonds,' are of a 'higher level,' they are a form of generative art or *'G-art'* (Boden & Edmonds, 2009).

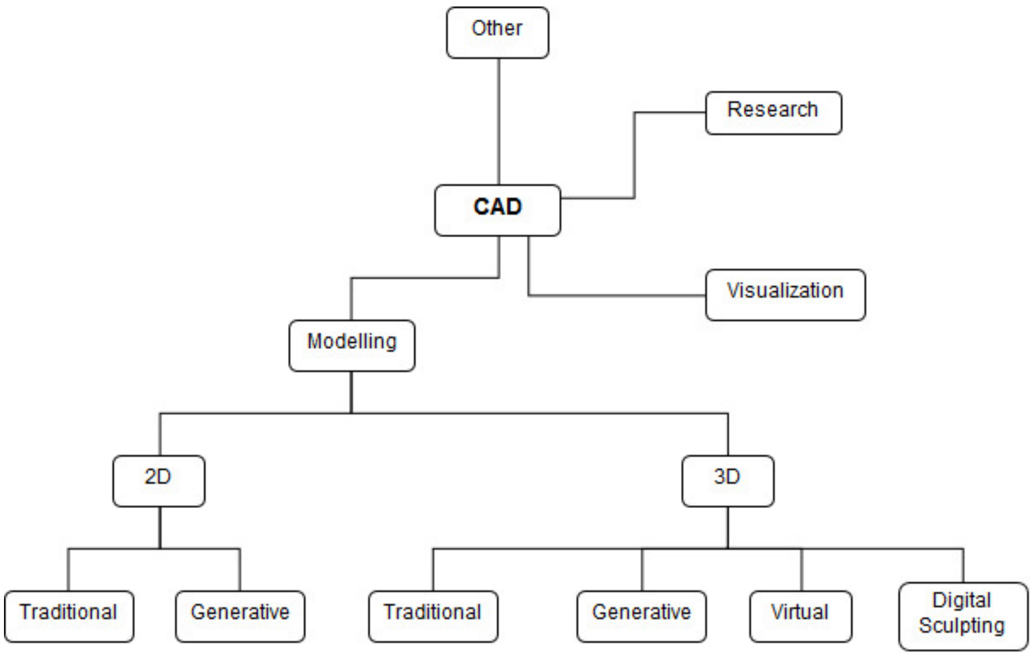

Figure 4.2 CAD branches (by authors).

The root of generative design is generative art; the term generative art has been used to designate different kinds of electronic art forms, but in general is the type of art in which the artwork is created by a process in which the artist is not completely in control (Boden & Edmonds, 2009). While the artwork is created produced by a computer program 'on its own,' Boden and Edmonds (2009) acknowledge that, it is rather impossible to conceive a computer program that requires zero interaction with a human counterpart (Boden & Edmonds, 2009).

The most prominent manifestation of computational creativity in the field of design has been in the form of *'generative design.'* These days, fields like video-games, graphic design, and architecture all make use of generative design techniques. Often times these techniques overlap fields; architects for example use generative design techniques used by video game designers to automatically create cities, landscapes and game level configurations. Jewellery designs have amazed viewers with their intricate and random forms.

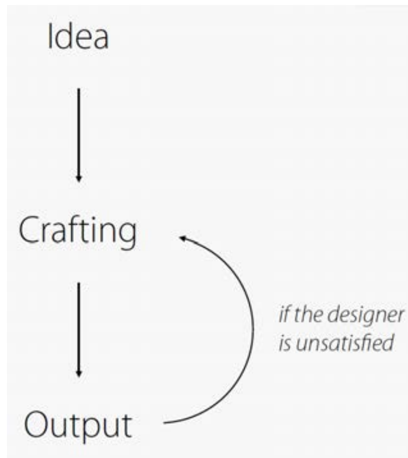

Figure 4.3 Workflow in traditional CAD (Enjalbert, 2014)

Chase (2004) classifies this approach into 'classical' and 'non-classical.' A classical process is when the rules of generation are clear to the designer and they are directly manipulable. A non-classical approach on the other hand is more like a 'black-box' and it is not particularly suitable for novice designers. In terms of understanding the process (Chase, 2004 p691). An example of these would be genetic algorithms.

Tools that feature a graphical user interface or GUI are relatively easy to use, an example is Autodesk's Shapeshifter. But their output is rather limited in application and often abstract. Chase (2004) refers to these tools as 'highly interactive systems.' However, to generate output which are more usable the designer needs to work with more complex tools some of which do not exactly have a GUI like processing, or grasshopper. These tools, however because they produce abstract outputs not directly usable, pose a challenge for novice designers because they need to know how to interpret these abstract outputs in a way they can be applied to a design (Chase, 2004 p697). With this paradigm, the computer takes on a more active role within the design process.

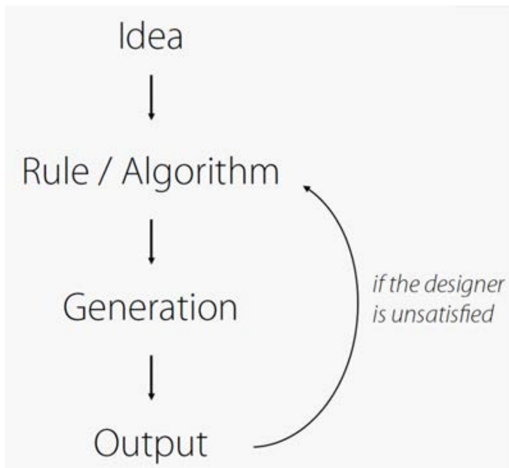

Figure 4.4 Workflow in generative design (Enjalbert, 2014)

A further step in modelling automation/form exploration is evolutionary design. The difference between generative design and evolutionary design is that a 'fitness' and 'crossover' functions are integrated after a round of generation. Before subsequent generations are made, these functions filter 'the best' individuals so that a design is steadily optimized after each round of generation. The 'Bone Furniture' project by Dutch designer Joris Laarman is an example of industrial design using computer software based on evolutionary design techniques.

Figure 4.5 Bone Chair by Joris Laarman (Bryden, 2014)

Generative Design techniques are important for several reasons; first, they automate the creation of forms that can have a variety of uses, second, they enable designers to mimic nature, but in a more sophisticated and fundamental way than just by copying forms from nature.

> There is a long tradition of mimicking nature in design, but the Bone Chair and Bone Chaise push beyond copying natural forms, instead utilizing mathematical code to reflect the code used by nature to create life. …our digital age makes it possible to use nature not just as a stylistic reference, but to borrow the underlying principles to generate shapes like an evolutionary process (Bryden, 2014 p9).

And lastly, they enable a different relation between designer and tool, one in which the tool has a higher degree of agency and in which the designer has less control over the ultimate form produced.

A symbiosis similar to that which has taken place in 'advanced chess' tournaments in which chess players are allowed to use computers. The story of 'Advanced Chess' not only has shown that human-computer teams are superior than computers or grandmasters alone, but also that an average player with strong computer skills makes a better partner of the computer than a grand master with poor computer skills (Gesher, 2010).

The third thing that is relevant in this context is the rise of a new way of doing 3D modelling, one that is fundamentally different from the traditional approach. Due the limitations of current generative design techniques to produce geometries usable in industrial design. These techniques have not been seriously considered, however, the advent of additive manufacturing techniques is fundamentally changing this situation because it removes the limitations that traditionally have prevented forms created through generative design techniques from being manufactured.

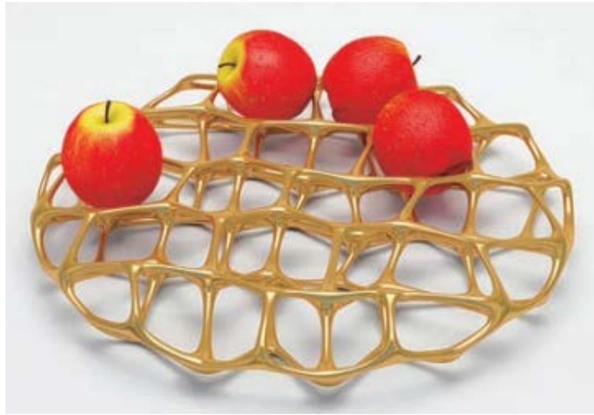

Figure 4.6 Fruit holder by Jane Kitannen (Bryden, 2014)

Figure 4.7 shows another creation of Kitannen. He specializes in using additive manufacturing techniques to create products 'inspired by the mathematics of nature,' which result in creations desirable to consumers. She represents the kind of designer who has understood how to use the computer as a tool that enables a different workflow/design process; using a variety of modelling software to explore his design and often skipping the sketching phase altogether. Something important to notice about Kitannen's process is that he often uses polygonal modelling, which enables him to work freely with forms which he does not need to worry about how to manufacture due the fact that he can simply 3D print them.

The fact that this approach emphasizes the tool, and that most developments in this area are driven by designers themselves, means that it opens the door to a new role of the designer; as tool creator, and this in turn opens the door to new business models and opportunities for them. Chase (2004) recognizes that a more effective use of these tools in education could be achieved if students acquired foundations in geometry and computer programming (Chase, 2004 p697).

4.4 The Rise of Additive Manufacturing

Just as there is a trend to move from ICT towards computer science and from passive 3D modelling towards generative/computational 3D modelling. There is an increasing use of additive manufacturing techniques (Bryden, 2014 p133). Which means that the issues with the models created using computational design approaches will become less important. With the advent of additive rapid prototyping techniques, these limitations are overcome. This is important because it allows the creation of forms/products that were not possible before. These techniques enable a new way of designing, and production of parts that are not possible using 'traditional' manufacturing techniques requiring moulding.

Until recently, polygonal models were not widely used in industrial design because while they type of geometry they generate is good for visualization; it cannot be used with other rapid prototyping techniques such as CNC milling. With the advent of additive manufacturing however, particularly 3D printing, the usability of these 3D models has increased substantially. Polygonal modelling however is suitable for a number of industries in which industrial designers often build careers; such as jewellery, or action figures. However, special systems have been for such applications (systems like Paraform, SensAble Free Form, recently acquired by 3D systems, and others). This these new Rapid

Prototyping techniques are *"freeing designers"* from the constraints of traditional manufacturing methods (Bryden, 2014 p7).

Sportswear manufacturer UnderArmour for example, launched the tennis shoes that appear in figure 4.7 in the year 2016 (Garfield, 2016). Later, competitor sportswear manufacturer Adidas came with their own version in the year 2017 (Teppez, 2017). As can be appreciated in the pictures, the design of the sole is very complicated, and would not be feasible, or even possible to manufacture them using traditional injection moulding methods, however it is possible to do it using 3D printers. While these tennis shoes are in part an experiment to test the feasibility of using 3D printing in production, and while their price is still nowhere near close to compete with tennis shoes produced using traditional methods, the trend is clear, the use of 3D printing in mass production will continue to rise, and with it, a whole set of possibilities for Industrial Design.

Figure 4.7 Examples of 'mass-produced' goods, using Additive Manufacturing methods (Garfield, 2016; Teppez, 2017)

4.5 Summary

Two factors are enabling/promoting the use of Generative Design in Industrial Design; the first is the paradigm shift from ICT to computer science all across the education system, the other is the advent of additive manufacturing techniques, most notably 3D printing. The story of 'Advanced Chess' not only has shown that human-computer teams are superior than computers or chess masters alone, but also that an average Chess player with strong computer skills is a stronger team than a grand master with poor computer skills (Gesher, 2010).

Technology can support creativity and innovation, but only when the person possesses advanced knowledge to exploit it. Computer technologies have become a major enabler for innovation in all creative disciplines. So, having a deeper understanding of how digital technologies work, is becoming a major asset to have. Future designers–that is, design students–have the opportunity to learn about this at school, so they can implement their knowledge once they join the work force, but only if they are trained in computer programming and computational design methods from a product design education perspective.

In addition, the instruction of generative design techniques can also contribute to understand aspects of the creative process, at a time in which design educators still lack proper training–and thus knowledge–about the neuroscience behind the creative phenomenon (Rivas & Contreras, 2016). Implemented in carefully designed study pans, these techniques can enhance the creative output of student's works, while at the same can help students understand how certain forms of creativity work. As the importance of the creative economy is increasingly recognized, any form of technology that contributes towards enhancing the creative output of designers represents a competitive advantage. This is of particular importance in countries currently aiming to transform their economy from being based in low-cost exports, to high-value added products (Siu & Contreras, 2016).

The importance of computer skills goes beyond mere efficiency, because there are models and modelling approaches that only exist in the computer realm. In the context of what is happening in basic computing education, and of constant specialization of industrial design as a discipline, it is beneficial to enable product design graduates so they can work with different types of models, not just visual representations/traditional models.

Computer skills in product design schools have focused on training the students in what is traditionally understood as CAD. This approach has numerous advantages but it mostly represents a computerization of the traditional modelling processes. While done in the computer the user must directly input and transform forms. This may aid students in understanding form, and develop their knowledge and skills in areas such as geometry. The power of generative design tools however, is that the designer can be lead on an exploratory path (Chase, 2004 p689). In this regard, Architects are part of what Jeannette Wing refers to as 'the converted,' those groups of professionals who have understood the role of computational thinking in their discipline. One of the best examples of this vision is biology in the effort to sequence the human genome (Wing, 2014).

In this sense, Architects have realized that there is more to computer modelling than just typical CAD applications. This is a point that has not been reached in industrial design. Another noticeable difference between the development of computer skills in schools of Architecture and schools of industrial design is that in the first, often times computer tools are developed, Chase (2004) for instance says the following when referring to in-house tool development;

> The benefit of in-house tool development is that the tools can serve teaching and research at several levels. Novice users learn basic design paradigms. Intermediate designers use generative design tools in

conjunction with other methods for more complex design. Advanced students become tool developers, gaining further insight into theories of design computation and software design. This approach has proven successful in the past and should be encouraged for the future (Chase, 2004 p697).

In Architecture schools for example, the use of these computational/generative tools/approaches is common at the later stages of the curriculum (Varinlioglu, 2015). The Architecture profession is a good reference for industrial design in terms of disciplinary trends because let's remember that industrial design as a profession branched-out from Architecture. Product design is a later development along the same line. How these four trends are interconnected and facilitate each other is discussed in more detail in chapter six.

Chapter Five

5 TRADITIONAL COMPUTER SKILLS

Having reviewed the context in which the research questions are raised, chapter five presents the findings of the study pertaining traditional computer skills. Starting with the results of the survey carried out with educational institutions, these results are complemented with the opinions of educators and experts, and findings emerging from qualitative methods. These findings are also enriched with findings from other research done in parallel to these studies, such as a study looking at the level of digital literacy of higher education students in Hong Kong.

The implications of these findings are then discussed in relation to the literature review and the context presented in chapter four. This discussion also draws from the experience of the author as industrial designer and industrial/product design lecturer in different countries. The discussion also uses education in architecture as a reference; as Margolin (2012) and Friedman (2014) recognize, the research tradition in the design disciplines is much shorter that in other fields (Contreras & Siu, 2016b). In the case of architecture however, Gun (2012) recalls that studies around computer applications in architecture, dates back to the early 1960's at the MIT, some of which have influenced architectural education worldwide (Gun, 2012).

5.1 Objectives

The results of the survey show that a considerable percentage of participants do not know the objectives of their institution's curriculum in terms of computer skills, as can be seen in figure 5.1.

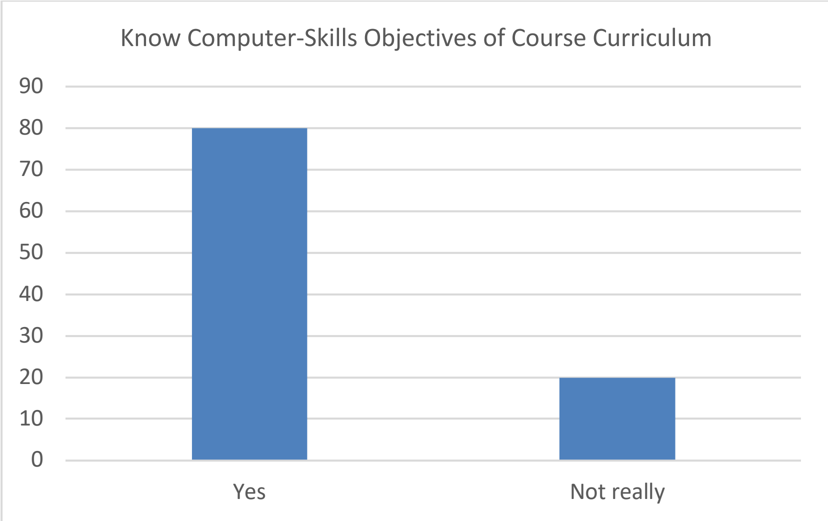

Figure 5.1 Percentage of respondentes who reported knowing the objectives of their institution's curriculum in terms of computer skills

Table 5.1 summarizes the learning objectives reported by those participants who reported knowing the learning objectives of their study program. In this table the objectives have been grouped in different clusters of objectives closely related. These learning objectives are consistent with the skills that institutions reported to teach (figure 5.11) as well as with the software tools that schools teach (figure 5.12) presented later.

Learning objectives in surveyed institutions in terms of computer skills			
3D modelling	Learn about solid-modelling	6	**33**
	Learn about parametric modelling	5	
	Learn about Solidworks	4	
	Learn about surface-modelling	4	
	Learning about Rhinoceros	1	
	Learning about Alias	1	
	Learning about 3D Studio Max	1	
	Develop good modelling habits	1	

	Learn about theories and methods for construction of geometry	1	
	Develop ability to de-construct geometry	1	
	Develop 3D skills	1	
3D rendering, visualization and presentation	Learn about rendering	4	**12**
	Lear about Photoworks	1	
	Develop visualization skills	3	
Learn about rapid prototyping			**5**
Learn about 2D graphics	Learn about Photoshop	1	**4**
	Learn about Illustrator	1	
Technical/Eng. drawing			**3**
Learn about the software processing	Learn about generative design methods	1	**3**
	Learn to create data visualizations	1	
Job readiness	Keeping out with technology	1	**3**
3D/Spatial-thinking			**2**
Learn about animation			**1**
Learn about assemblies			**1**
Computer programming			**1**

Table 5.1 Summary of learning objectives of participant's institution's curriculums in terms of developing computer skills.

As can be seen in this table, there is a clear focus on learning objectives related to 3D modelling. Other aspects such as engineering analysis of 3D models or computer animation are seldom mentioned. This may have to do with the fact that most educators still see CAD as the same as 3D modelling. This suspicion was confirmed at different stages in this study, starting with the pilot study. The following reply to an email asking for help to locate the instructor of computer courses serves as an example;

> Our CAD course is taught by–name–an adjunct instructor. All of our sophomore product design students take a 7-week introductory digital

prototyping course with him using Solidworks as the foundation. In my class I usually follow up with an introduction to exporting fabrication drawings but I do not teach actual 3D modelling.

If what faculty think should be the objectives of the curriculum in terms of CAD (table 5.2), and at the definitions of CAD provided by faculty (table 5.3), it is not so hard to see why this instruction remains almost exclusively focused on teaching students computer modelling and even more, only 3D modelling. Either the objectives or definitions provided, are expressed in terms of modelling/visualization, or they are vague.

Objectives mentioned by participants	Overall theme of the objective
Being able to create with CAD without limits.	CAD/Modelling
Being able to create a design solution in 3D.	
Being able to freely express any form that can be industrially produced.	
Being able to work independently on CAD.	
Being able to model complex forms and then make prototypes from this data.	
Have a well-rounded ability to operate CAD packages.	
To know when and where (in the design process) to use CAD tools.	
Ability to communicate design through digital means at a near-fluent level.	
Make aesthetically accurate models.	
Have mastery of basic (CAD) concepts and tools.	
Solid understanding of 2D illustration and 3D modelling and rendering.	
Provide a generic understanding of how CAD can be used in the design process.	
Be able to configure program tools.	Configure Software
Have a basic understanding of computer programming.	Computer Programming
Make students aware of the methods applicable to digital product design.	Awareness of digital methods
Foster a competitive designer.	Job readiness
Prepare students for work.	

Be aware of computational tools.	Computational Design
Intermediate understanding of motion/video programs.	Animation/Video editing

Table 5.2 List of objectives that participants believed should be the main objectives of the curriculum in terms in term of developing computer/CAD skills.

Definitions offered by participants	Overall theme of the definition
Use 3D computer technologies as part of the design and development process.	
Creating design and physical things in 3D with computer	
Modelling any imaginable form to production readiness with software	
Creating 3D models and directly related 2D drawings	
Digital 3D modelling tools.	
The generation of three-dimensional virtual forms on screen using a computer.	
2D product rendering and 3D solids and surface database	
A way to digitize a design	
Using digital software at any point in the design process, from ideation to production.	
The design, testing and evaluation of product digitally using computer software.	

Table 5.3 List of definitions for 'Computer Aided Design' (CAD) offered by participants.

It should be noticed however, the role that even academic literature may have in spreading this conception; in some cases, scholars themselves have fallen into the trap of thinking of CAD as being just about modelling. Bryden (2014) for example defines CAD as the process of using computers and specialist software to "create virtual three-dimensional models and two-dimensional drawings of products."

Also, to be highlighted, is the relative prominence of objectives related to solid-modelling approaches. These results are consistent with the order in which institutions teach the different modelling approaches (figure5.7) and with the software tools taught (figure 5.12) presented later.

It should be noticed, that a rather surprising finding emerging from the analysis of the profiles of the participating institutions, was that there is a higher than expected number of industrial design programs being hosted in engineering schools. This may be a potential explanation for this result. One of the programs that is hosted in a school or architecture on the contrary, not only reported to incorporate two different surface modelling tools, but also was one of the few institutions that reported incorporating the instruction of a polygonal modelling software tool—3DS Max—in its curriculum. More information about the profiles of the participating institutions can be consulted in appendix II.

Of the participants who claimed to know the objectives of their institutions, only a limited number were actually able to articulate with clarity such objectives. A good number of respondents provided very abstract/broad description of those objectives, often expressed in terms that provide little guidance for a study plan, for example; "to be in balance with analog making", "provide students with digital skills used in the industrial design profession", "provide real-world preparedness", "provide a digital tool set to complement industrial design degree", "modelling any imaginable form to production readiness with software" and, helping students to be "self-dependant & aware."

In some other cases the objectives were not as abstract, but their articulation was still vague. As an example, one of the participants replied: "Use both a parametric and a freeform program." The fact that the terminology used in the description is rather unorthodox means that this could be interpreted in a variety of ways; what exactly is meant by 'freeform'? NURBS? Polygonal modelling? Digital clay? Another wrote: "Computer Rendering." However, computer renderings can be of two kinds;

2D and 3D, consequently this objective too could be interpreted in a variety of ways, conversely each of both could be left out and it still could be claimed that the objective is being satisfied. Several more examples of this kind were reported.

Objectives expressed in such terms leave room to poor implementations in module outlines to happen. However, attention needs to be brought to the fact that sometimes institutions/educators find it convenient to express learning objectives in such terms. The reason is that it is easier to claim that learning objectives are being met while leaving enough freedom for educators to do whatever they want or can. This practice however, cannot guarantee a proper instruction.

Another explanation is that perhaps educators are not aware of the differences between the different modelling approaches. This is supported by the fact that when data provided by participants to the questions regarding what computer modelling approaches students learn, and what are the modelling tools they learn was analysed, important discrepancies were found.

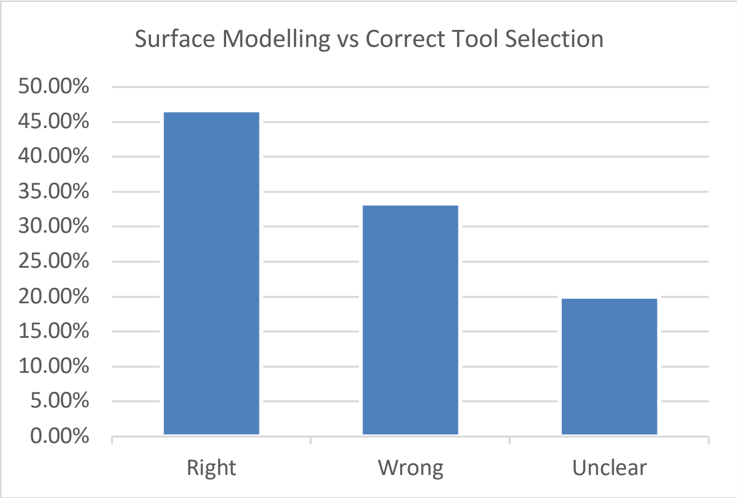

Figure 5.2 Percentage of respondents who chose a matching/not matching modelling tools after saying students learn surface modelling.

As can be seen on figure 5.2, out of the participants that said their students learn NURBs modelling, an important number then said their students learn to work with tools that are either not NURBs. A possible explanation is that even if they understand the difference between surface and solid-parametric modelling approaches, perhaps the difference between surface modelling in a NURBS environment–what the question asked–and surface modelling in a parametric/solids environment is not fully clear.

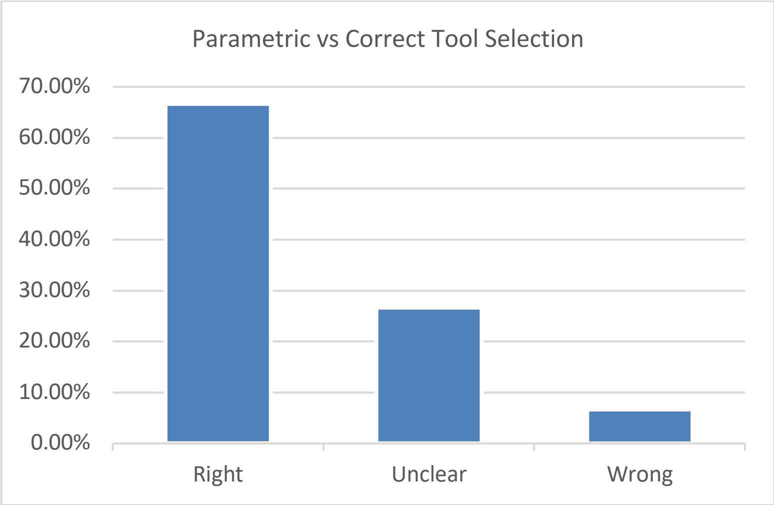

Figure 5.3 Percentage of respondents who chose a matching/not matching modelling tools after saying students learn surface modelling.

The same was observed when analysing the responses of participants who reported their students learn parametric/solid modelling (figure 5.3) above, and those who said their students learn polygonal modelling approaches (figure 5.4) below.

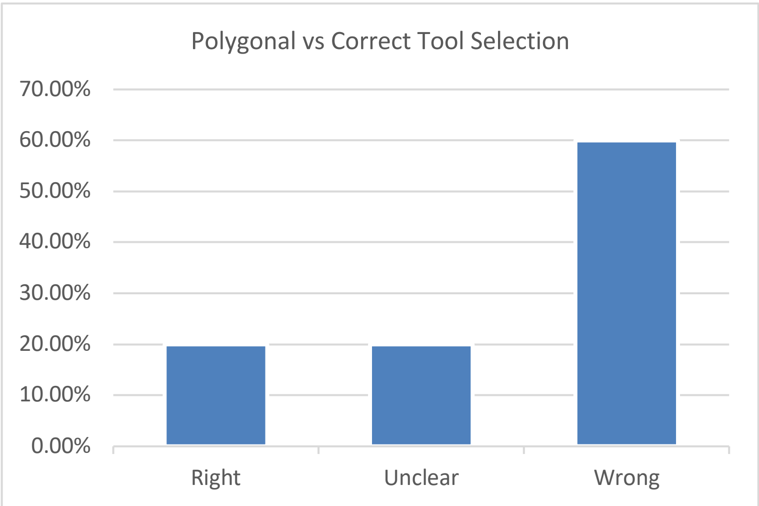

Figure 5.4 Percentage of respondents who chose a matching/not matching modelling tools after saying students learn surface modelling.

Through these answers it is possible to see that perhaps there are issues around teacher's qualifications to train students in this area. As an example; another participant wrote the following as one of the learning objectives: "Utilize 3D printing as a way to verify CAD data, confirm production details and communicate design intent." Why 3D printing? From the answer it is not possible to know what exactly is meant by 'verifying CAD data,' however, why not CNC milling for example? After all, CNC milling can be quickly used to produce 'blue-foam' models which then can be adjusted using manual tools.

5.2 Approach

5.2.1 Integration

The results show that the prevailing way to incorporate instruction of computers in the majority of institutions is to have one or more courses specifically dedicated for this purpose. As discussed in chapter two, this approach, that Varinlioglu et al. call 'discrete' (Varinlioglu, 2015), has the disadvantage of often focusing on developing what Chester (2007) calls 'command knowledge,' and neglecting 'procedural knowledge.' As it will be shown later on, results of this survey show that while small, there is still a number of institutions that follow a 'command by command' demonstration approach in the classroom, effectively falling into the error pointed out by Chester. The difference in percentage between schools that follow this approach and schools that integrate this learning outcome as part of other modules can be seen in figure 5.5.

When institutions follow an integrated approach, this integration can occur/take place in several modules. Typically, these modules will be modules in which students working on a design project that requires of the use of–traditional–computer/CAD skills. This approach is similar to one followed in preparatory education in countries like the UK. Sometimes learning objectives pertaining computer skills were reported to be integrated into course such as *human factors* or *design for manufacture*. In the case of design for manufacture for example, this integration occurs through the creation of engineering drawings.

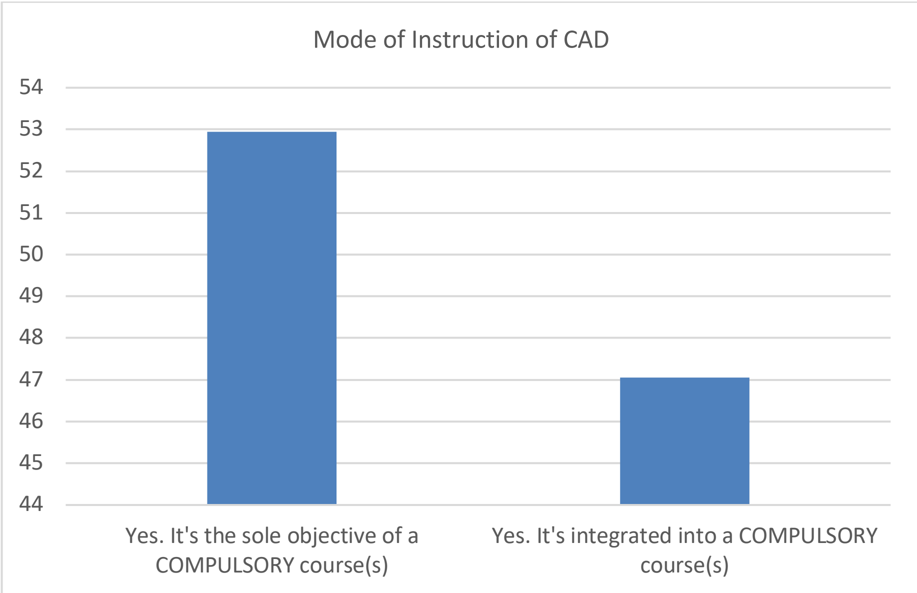

Figure 5.5 Percentage of institutions that follow a 'discrete' or 'integrated' approach in terms of CAD.

Another way this instruction is integrated into other modules is in the form of workshops. In these workshops, students receive support to construct/render their models. Students are therefore required to work in advance so that they come to the workshops prepared with questions. As an example, one institution reported that this integration happens in a module for which the following introduction is provided: "You'll learn to support your design process by using computer, physical modelling and presentation skills, and colour, texture and light using computer and manual techniques."

This approach to integration however is good only when students have received general CAD instruction already. Having all computer/CAD instruction attained through this approach is risky because it cannot guarantee a thorough exposure to basic/general knowledge. The reason is that each project is different, therefore the set of CAD tools and modelling approach would be different. In addition, showing a student how to build something, does not mean the student understands the general principles behind. It is important to pay attention to the motivation behind an integrated approach; this should not be the notion that computer skills are peripheral or secondary. If this is the case this integration approach is likely to result in a poor coverage and poor achievement of learning outcomes.

5.2.2 Starting point

All institutions reported to set their first computer courses in the freshman (first) and sophomore (second) years. However, as shown in figure 5.6, a large number of students will have their first computer-related module/course during their first year. This raises questions as far as to whether students would have had enough exposure to basic design foundations by the end of their first year.

It was shown in chapter two, that a prominent position/discourse among industrial design educators is that exposure to computer aided design should be/is better avoided during the first stages of the student's education.

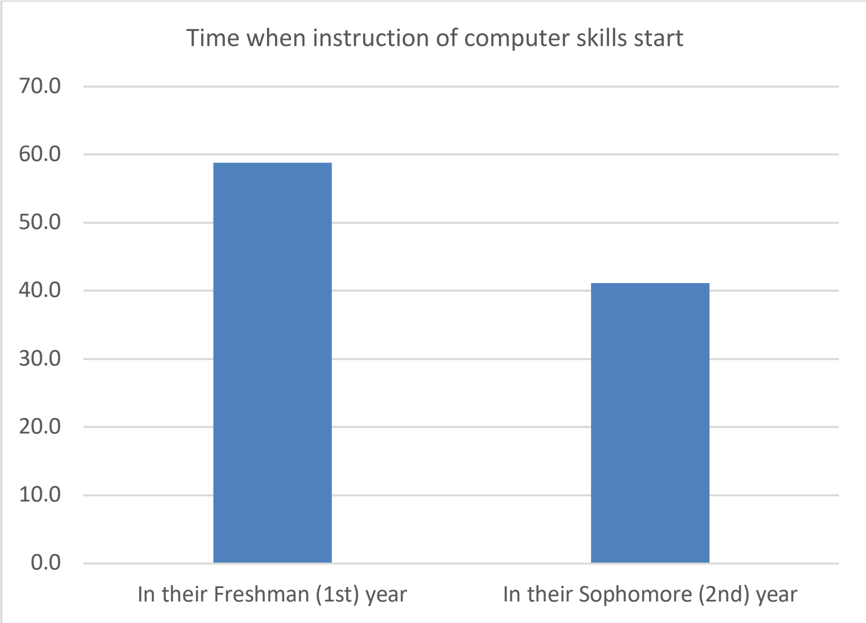

Figure 5.6 Percentage of schools that reported having their first computer courses in the first year or the second.

This is important because, neither the objectives, nor the content of computer-related modules taught in the majority of industrial design programs studied, reported to provide students with the theoretical foundations highlighted in chapter two as necessary to successfully use CAD technologies. Instead they immediately take the student to start using a CAD package. As seen in chapter two, a good understanding of the design process should determine the selection of modelling methods and consequently of tools. However, the results obtained show that this is not a clear objective of the curriculum of most institutions. This exacerbates the risk of students becoming 'masters of the tool' before 'becoming masters of the trade.' A solid understanding of the processes to be implemented using the computer should precede CAD instruction.

As discussed in chapter two, traditional modelling tools were to a substantial extent developed as abstractions/computerizations of traditional tools. Consequently, all computer instruction should be underpinned by a solid exposure to these tools and processes. Once the foundations of the processes implemented in the computer are somewhat understood, then the particularities and benefits of computer implementation can be more easily identified and discussed.

Although good for efficiency, if not supported with proper foundations, jumping straight onto the computer has the drawback of depriving students from learning important concepts i.e., if students start making 3D models using the computer without having had the experience of physical model-making in advance, their sense of scale, and understanding of how surface transitions from one to another is hindered. Is like teaching primary students how to multiply using a calculator and stop teaching them math.

The results of the survey show that most institutions provide no training in technical drawing. Schoonmaker (2002) recognizes that one should not be misled to believe that because access to 3D CAD is commonplace 2D CAD is not important. On the contrary, mastering 2D geometry and 2D maths may be even more important (Schoonmaker, 2002 p181). While 3D modelling makes it possible to configure models in 3D without knowing descriptive geometry. The relation between descriptive geometry and 3D modelling was also observed by this author when teaching computer aided design. Students with a stronger descriptive geometry background were able to start working in 3D relatively faster. Without this background students struggled more to understand how orthogonal views relate to the object/model or what they mean. The same occurred with digital hand-sketching; students with more exposure to traditional techniques were able to understand concepts like blurring and layers more easily.

The implications of a lack of exposure to foundations and traditional forms of modelling go far beyond enabling students in their acquisition of computer skills. Neglecting a thorough exposure to traditional modelling approaches can contribute towards walking into the 'technological trap' discussed in chapter two; the notion that as society we tend to rely on an increasing number of layers of technology that distance us from basic forms of operation.

The introduction of computers in industrial design education may have empowered students with weak visualization skills as recognized by Unver et al. (2006), and exposed students to the benefits of CAD in terms of efficiency and precision (Haughton, 2001). However, Keirl (2004) reminds us that we cannot assume that all is well with our technological world (Keirl, 2004 p23). Oppenheimer notices that whenever a new technology simplifies a task "something is always lost" (Oppenheimer, 100). The question is only whether the value of what is gained is greater than that of what got lost.

For example; with 3D rendering CAD packages, students can render beautiful scenes without understanding basic concepts about lighting. Anecdotal evidence obtained while discussing these studies show that this is a common concern among some educators. For example, a colleague–architect–related how he sees a problem with the use of 3D rendering systems–like Autodesk Revit–because, students have lost a real sense for the material. In the past students would get to know the material in physical form first, and only then they would apply it in their designs. What he sees however, is that more and more students propose materials by simply applying them from the software library, without having any real knowledge about it, any real experience from which they can develop a better understanding about what the material properties are in reality.

The author experienced the same at different points as industrial/product design lecturer; students often applied materials without having any real reference for the physical properties of the material, whether they are environmentally friendly or not, or the processes they involve. Consequently, they often apply materials in parts of the product in which that particular material could not be applied in reality. Some other students struggled to get a sense of scale and proportion from the 3D models they built.

5.2.3 Order/sequence/progression

The majority of institutions reported that their students learn solid modelling first. The next most frequent answer was that students learn both approaches at the same time. These results were equally confirmed by responses provided by educators via open-ended questions regarding the teaching approach followed in these modules. For example, one of the participants replied "We progress from basic solid modelling into complex surfacing during the 14-week semester."

Figure 5.9 shows the percentage of institutions that follow each of the three different approaches. These results are consistent with information from open ended-ended questions regarding the teaching approach followed in class, and in which participants acknowledged teaching solid modelling first, followed by surface modelling.

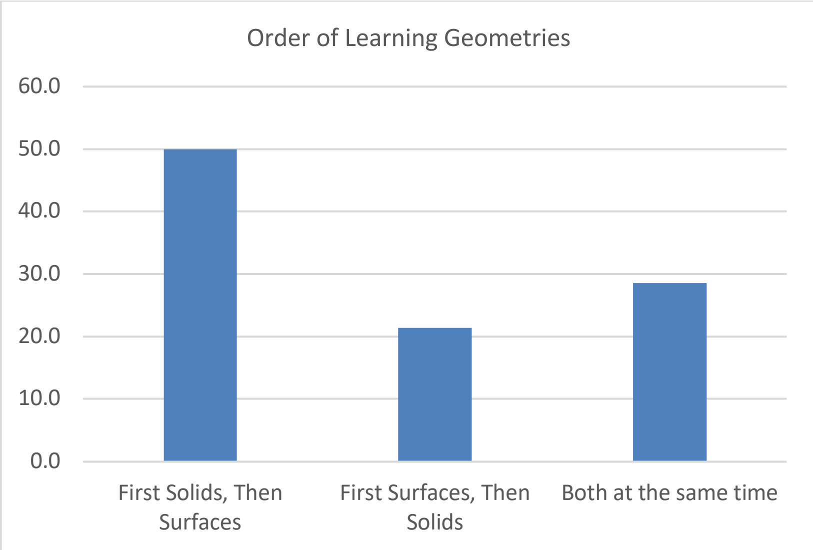

Figure 5.7 Order in which students learn to work with different geometries

Not many institutions articulated a strategy regarding the progression of learning objectives. However, based on the few that, it is possible to perceive an inclination towards moving from simple tasks to tasks of increasingly higher complexity. One of the most descriptive examples, articulated the progression across the different modules is as follows:

> First Year: introduction to surfacing and solid modelling and assembly techniques as well as basic rendering. Second Year: Complex surfacing, engineering drawings of assemblies and rendering as well as 3D printed outputs. Third Year: Modelling and rendering for communication purposes especially portfolio creation.

Considering the information presented in chapter two, the sequence in which the different modelling techniques are learned can have important implications. While there is no theory that supports a particular sequence *per se*, surface modelling has been identified as an approach more closely related to the nature of form exploration that industrial designers engage with, particularly at the beginning of the design process.

Also, as discussed in chapter two, for long there has been a discussion regarding what is the best modelling approach for industrial designers. It is interesting to notice that until the advent of concurrent engineering, industrial designers tended to work primarily with surface modelling programs—NURBS, while engineers tended to work with solids. However early solid modelling programs were not able to make the complex organic surfaces that industrial designers work with. It was until primarily solid modelling programs like Pro-Engineer and Solidworks expanded their surfacing capabilities that they became attractive for industrial designers. This is still not acknowledged by those who have tended to see CAD from the perspective of their own discipline; Schoonmaker for example, argues that CAD users are 'historians' not 'sculptors.' He goes on to say that "there is a natural tendency to view the 3-D model creation as an exercise in sculpting. New users often create a basic shape and then cut away segments of that shape until it has the desired final shape" (Schoonmaker, 2002 p183). However, Schoonmaker assertion in a way confirms that there are substantial differences between how engineers and industrial designers approach 3D modelling. For industrial designers however, having the freedom to sculpt surfaces is key (Bilalis, 2000).

It is also true however, that the fact that industrial designers have to work with engineers, means that they have to work with both approaches. This puts a strain on education, particularly when, as has been seen in chapter two, some educators feel that the curriculum leaves no room to cover any additional material.

On the other hand, based on the opinion of some scholars who agree that working in a parametric environment is seen as more difficult than working in a purely surface modelling one, a progression of increasing challenges, perhaps surface modelling should precede solid modelling seems logical. The observations made on student's works showed that students who work with surface modelling display better 3D models and renderings. Consequently, if there is a better approach to the teaching of traditional modelling approaches, this would probably be to have surface modelling in a NURBS environment first.

5.2.4 Intensity/Depth/Scope

Figure 5.23 shows the percentage of institutions that reported to have either one, two, three or four courses in which at least one of the learning outcomes of the module is to develop computer skills.

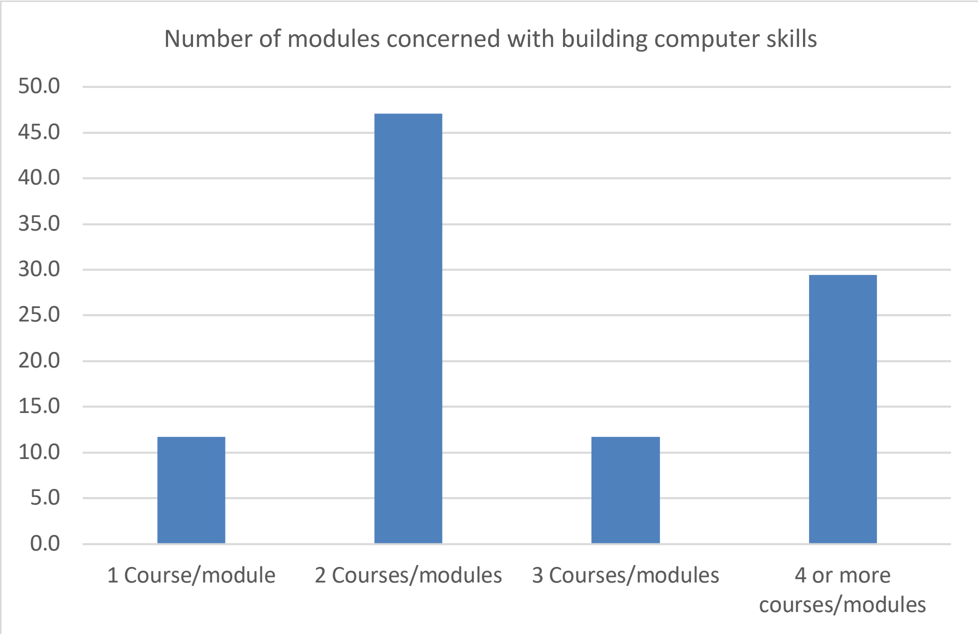

Figure 5.8 Percentage of institutions that reported having 1,2,3 or 4 courses dedicated to building up computer skills.

As can be seen in the chart, a smaller but still substantial number of institutions reported only having one module with specific learning objectives related to computer skills. The majority of institutions reported having two courses. In addition, as can be seen on figure 5.9, most study programs are four years long. But while it was found that in North America and Australasia there is a marked tendency for programs to be 4 years in length, there is an identifiable trend suggesting that the length of studies in some countries in Europe for example is three years. Similarly, it seems that programs labelled 'Industrial Design' tend to be longer than programs 'Product Design.' This however, is likely to the fact that the label 'Industrial Design' as we have seen is more popular in America, thus the relation. More information pertaining statistical description of participating institutions can be reviewed on appendixes I and II.

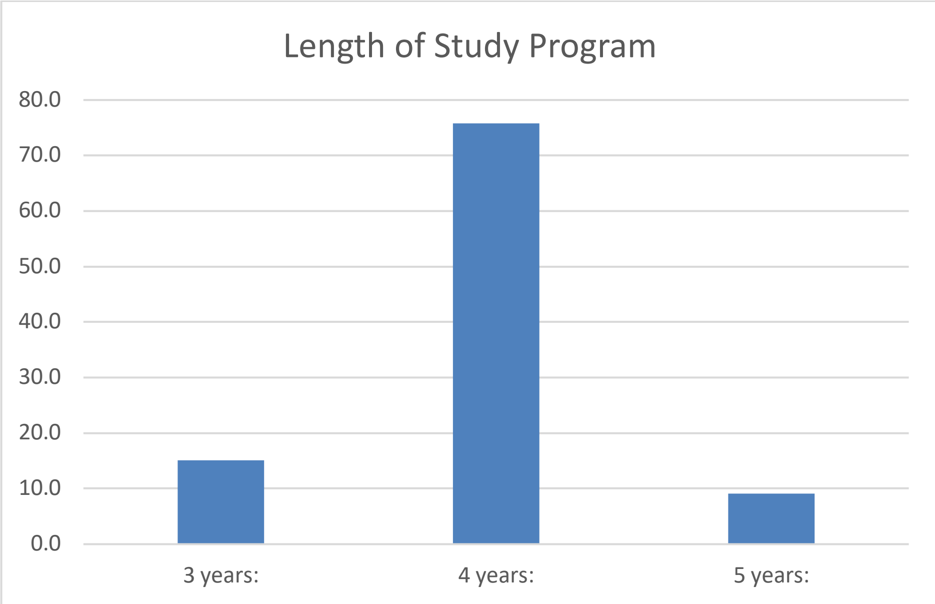

Figure 5.9 Length of industrial design study programs.

This prompts the question of whether the extent/depth to which these computer skills are taught in the majority of institutions is sufficient, or to what extent does it prepare students so that they are able to model their ideas in the computer. This concern was stressed after reviewing final-year project reports. These showed that a number of students repeatedly acknowledged to not have been able to model their products the way they had in mind. This review of reports also showed that students often do not know how to select the right modelling approach: Parametric-Solid, Surface, or Polygonal. For example, a student attempted to build a heavily ornamented jewellery stand using parametric modelling when a simpler approach–at least for representation purposes–would have been to use polygons.

The same reports also showed that only a very small number of students made effective use of the tool for testing different colour schemes. Equally no students were found who used digital 'dummies' to check the scale/proportion of their designs, which is a very easy way to check and communicate ergonomics. Other issues observed included students doing 'basic' mistakes, like not filleting edges of their 3D models before rendering, rendering using 'wrong' projection styles–such as isometric or oblique–and using image editing tools to represent design features that could be very easily modelled in 3D and obtaining much better results.

Bryden (2014) recognizes that it is only when product designers become truly proficient with 3D software that they can visualize in an unrestricted manner, just as they would not be able to explore a concept on paper if their sketching skills are not fully developed. He reminds us that learning a new 3D CAD application can be difficult. Depending on the particular software, this can take months (Bryden 2014 p62; Rogers, 2017).

In "Rulers and Dividers: A Technology of Design," Philip Luscombe (2018), provides a remarkably simple, yet effective example of how different tools can be used to achieve the same goal. This example however, can also be used to illustrate how different tools require that users have different levels of preparation: You can use either; dividers or rulers to design a chair, but you cannot use a ruler without at least a certain level of mathematical literacy. The fact that computer skills in general, are being taught in Industrial Design schools, without assessing and/or providing general IT knowledge, is a mistake.

Although discussing computational thinking, Jeannette Wing (2008) says "we don't want the tool to get in the way of understanding the concepts." The same applies here, and one way of preventing that, is that students learn the concepts before learning how to apply them/relate them to the tool. At the same time however, as she also acknowledges, we do not want to miss the opportunity to sue the tool to reinforce the concept.

The issue should not be underestimated. A collateral activity of these studies, involved the review of portfolios of applicants to the Product Design course at a University in the UK for two years. This review revealed substantially disproportionate levels of CAD skills among those applicants. Professor Ivan Chester highlights that for some teachers it may be hard to believe that 3D-CAD has now been taught in secondary schools for more than twenty years (Chester, 2007; Unver, 2006; Haughton, 2001; Winn & Banks 2012). However, the situation in other countries can be substantially different, while in some countries high school students have been able to work with sophisticated CAD tools—such as Pro-Desktop—for decades, in some other countries high school students have had a more limited CAD training.

This can be a particular problem in light of increasing student mobility that brings ever larger numbers of international students to Universities around the world, but particularly in certain countries in Europe and the US. This makes it difficult for institutions to 'strike the right balance' of computer training in the curriculum. In addition, as Universities have often seen their budgets cut, many of them have lowered or softened their admission requirements in hope of securing tuition money, admitting students with a substantial lower level of skills including computer skills. These students help to alleviate the situation, but on the other hand mean that institutions need—at least should—assume a standard, and lower level of computer skills of incoming students.

The situation within countries is similar. During the interview of prospective students for the product design major at a university in the United Kingdom, substantial variations in CAD skills of applicants in the 2017-18 cohorts were observed. While these differences depended mostly on the type of A-Level's qualifications they had, important variations could also be observed depending on the different schools they came from, even if they had the same type of A-Level's. Sometimes students had some experience using advanced tools such as Solidworks, while some other times they had no CAD experience at all (Winn & Banks 2012). The same was reported by Unver (2006) who also acknowledges that incoming students have a diverse background, some with good graphics and CAD skill, and some without.

It has to be pointed out however, that the results shown in figure 5.8 have the limitation of not providing enough information from which it would be possible to tell the depth to which computer skills are covered. For example; some institutions can choose to have one or two dedicated courses—more intense approach—while other can choose to have four integrated courses—less intense approach—and still offer the same level of coverage.

Having enough space in the curriculum to foster computer skills ought to be carefully considered. As seen in chapter two, some scholars argue that since CAD programs usually sit on 2 or more 'layers' of software to work (figure 5.10), an effective use of CAD requires knowledge of these layers as well (Schoonmaker, 2002). However, most Universities tend to assume general ICT knowledge of incoming students. This is a mistake, a survey conducted in relation to these studies, revealed that higher education students showed gaps in their knowledge of IT.

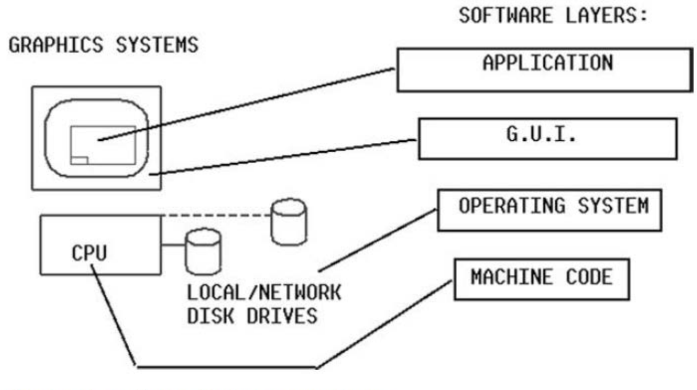

Figure 5.10 Layers of software upon which CAD rests. (Schoonmaker, 2002 p46).

Part of the problem, says Murray and Perez (2014) stems from the fact that academia tends to take the fact that students are constantly immersed in digital technologies as a sign of literacy.

Despite the fact that many students entering the university have a high level of exposure to digital technologies, they are not prepared to cross the bridge between personal and academic use of technology. Instead, technological knowledge is to be gained through structured learning (Murray & Perez, 2014). This contrasts with what Chapman wrote in 1995, regarding the fact that basic computer literacy tended to be in general higher among incoming students than among many staff. On the contrary, he argued that at that time, few students have used CAD before they arrive and even fewer have worked in 3D on a computer. The situation has inverted.

5.2.5 Content

The range of computer skills taught at industrial design schools appears in figure 5.11. This list excludes 3D modelling, which, based on the pilot studies, was considered to be present by default in all institutions. As can be appreciated in the chart, the second most common 'computer skill' taught reported was 'using rapid prototyping equipment,' followed by 'photo retouching and composition.'

It should be noticed that while sketching digitally appears as a skill taught, figure 5.11 does not show any software for digital sketching–current options would include software tools such as Autodesk Sketchbook pro, Corel Painter and Art Rage. The only software on this list of tools reported that could be taken as/work for this purpose although not the optimal, is Photoshop. Something else that needs to be highlighted from this chart, is the lack of computer instruction in areas like research and project management. This information is consistent with the results regarding learning objectives reported earlier in this chapter.

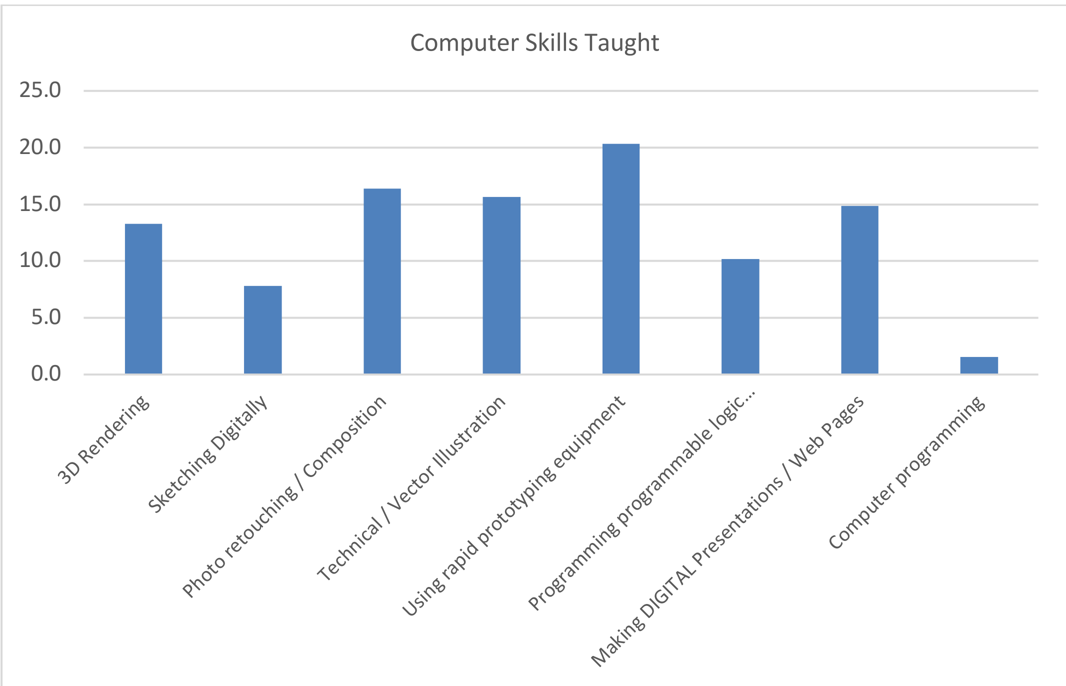

Figure 5.11 Range of computer skills taught in industrial design schools (besides 3D modelling).

In terms of the software tools taught, the survey's results show that Solidworks appears as the most common software tool followed by Rhinoceros. The next most common tool reported is 3D-rendering application Keyshot. As can be seen, the rest of the software tools rank very low in comparison. Something to highlight from figure 5.12 is the difference between Solidworks and Rhinoceros. As results show, the teaching or popularity of the first is considerably higher than the second, this obviously means that some schools only teach solid-modelling. As seen in chapter two however, surface modelling has been found to be more appropriate for the type of form exploration that industrial designers do, particularly at the beginning of the design process. This has important implications that will be discussed in chapter six.

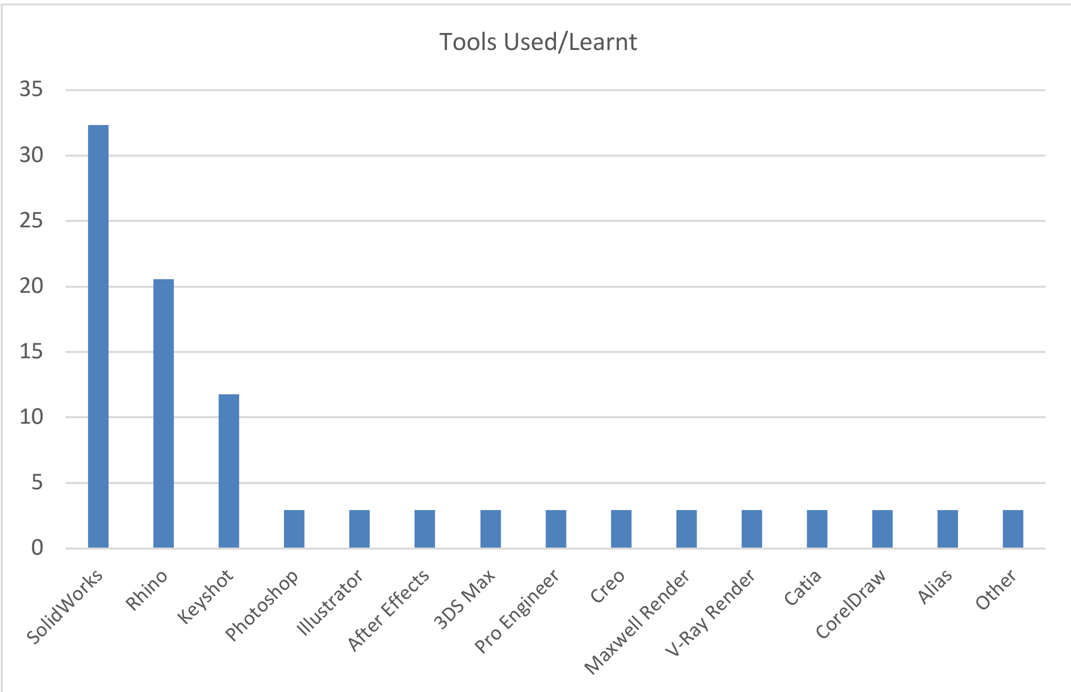

Figure 5.12 Software tools taught in industrial design schools.

Another important point to notice, is the difference between Solidworks, Rhinoceros and 3DS Max, as well as the fact that 3DS Max is the only polygonal modelling tool that appears on the list. As suspected, polygonal modelling approaches are seldom taught in industrial design schools. This finding is consistent with both; the information provided by participants regarding the learning objectives of their institutions, and with the modelling approaches that students learn which appear in figure 5.7. For example, a respondent reported his/her institution had the following as a learning objective: "to understand both solid and surface modelling principles and how of effectively apply them."

Also, to notice is the difference between the responses on figure 5.12 and figure 5.13. While the overall trend is consistent, in figure 5.12 Solidworks is reported as the most popular software tool used. However, in figure 5.13 NURBS is reported as the most common modelling approach taught. The reason why this is an issue is because Solidworks is not a NURBS modelling environment. Similarly, the difference in use between Rhinoceros and 3DS Max that appears in figure 5.12 is of around 12%, yet the difference between surface modelling and polygonal modelling in figure 5.13 is of around 25%. The simplest, most economic explanation of this discrepancy, is that some educators may not have clear the differences between the different modelling approaches, and/or, that the differences between surface models in a parametric modelling environment and those made in a NURBS modelling environment. The fact that Solidworks is reported as the most common software tool in industrial design schools also has implications for teaching generative design methods, this will be discussed in more detail in the following chapter.

The revision of final year project reports revealed that several changes in software vendors had taken place over time–for example from PTC's 'ProEngineer' to Dassault Systems' 'Solidworks.' This type of changes should not be seen in isolation from the large sums that educational institutions can spend on educational technologies, and the 'powerful forces' that such spending invite. Thornburg (2013) warns that software has evolved too to follow education's trajectory, and its dollars (Thornburg, 2013).

The most common modelling approaches that students learn, are: solid and surface modelling. A much smaller percentage of institutions reported incorporating polygonal modelling in the curriculum. The specific percentages of institutions that teach each of the three modelling approaches appear on figure 5.13.

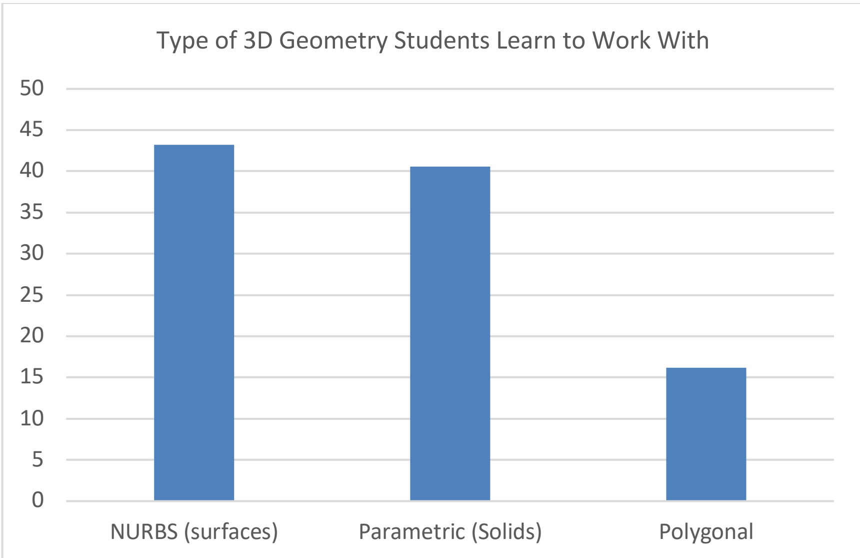

Figure 5.13 Types of 3D modelling approaches taught in industrial design schools.

As discussed in chapter four, advances in additive manufacturing techniques are making it possible to use this type of 3D models in the manufacturing–not just prototyping–of products with complex shapes. At the same time low-cost 3D printers are becoming pervasive in most

Industrial Design schools (figure 5.14). Yet, even though 3D printing is becoming widely used in industry and academia, most institutions do not incorporate the instruction of polygonal modelling in their curriculums.

This situation, as one of the interviewees recognized, 'is paradoxical,' because as he put it; "It's funny that the software we use in ID schools is not good to produce type of geometry we need with 3D printing." This situation warrants the importance of reconsidering the relevance of teaching polygonal modelling in design schools. It also warrants the importance of critically evaluating the dependencies that accompany technological adoption and which can lead to the technological traps previously discussed.

Figure 5.14 Now 3D printers are pervasive in most design schools (July 2015, Workshop, school of design Notre Dame University. Photo by authors).

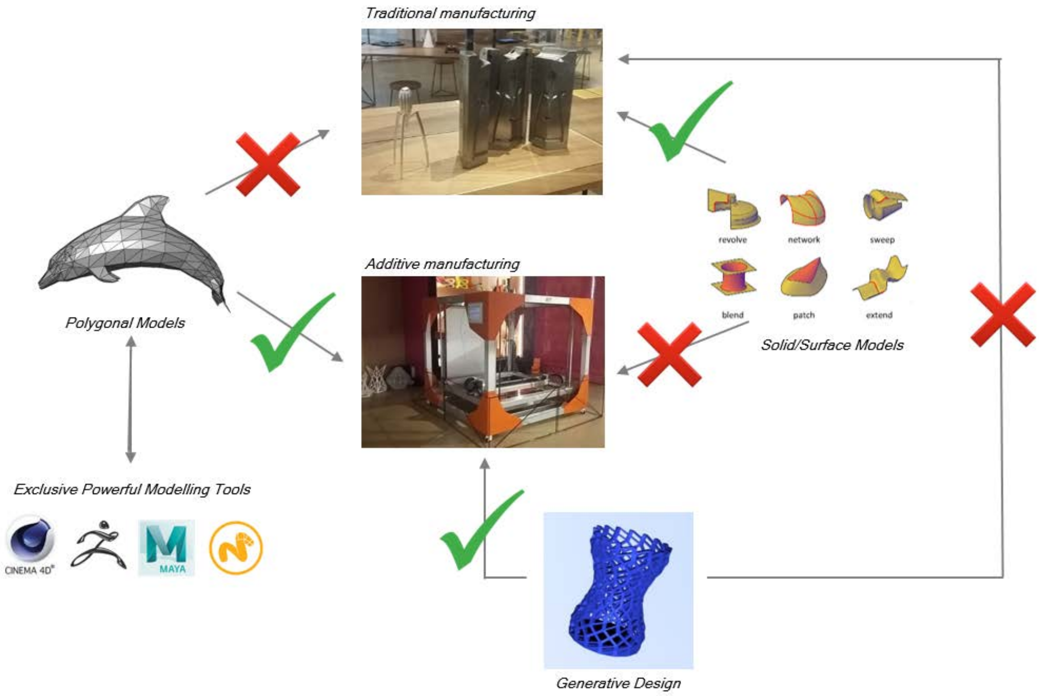

Figure 5.15 Relationships between Polygonal, Solid and Surface models, and Traditional and Additive manufacturing methods.

Despite that Polygonal models are not good for making production molds (such as those used in injection molding), Additive Manufacturing methods such as 3D printing work exclusively with polygonal models. While it is relatively easy to convert either Solid or Surface models into Polygonal models, there are a number of issues with such workflow; first, the resulting–converted–model can have errors, the chances of this happening are directly proportional to the complexity of the model being converted. In order to correct this errors, Polygonal Modelling skills come in handy. Moreover, there is a vast array of powerful modelling tools that work exclusively with Polygonal Models. The reason is, that these tools have been developed for the entertainment–Cinema, TV and Videogames–industries, and in these industries, there are no 'manufacturing concerns' since everything is destined to the screen, and on the other hand, Polygonal

Models in fact offer a lot of flexibility. With the advent of Additive Manufacturing methods, it is now possible to use those tools, to create models that can then be turned into an actual physical object as described in chapter four. Consequently, Polygonal Modelling is becoming an increasingly necessary computer modelling skill in Industrial Design, as is its incorporation in education. These relationships previously discussed are illustrated in figure 5.15.

Something to consider is the definition of CAD. Since there is no finite definition, some do not consider polygonal models–and consequently polygonal modelling systems–as CAD. Bryden (2014) for example considers that CAD modelling is divided into two basic techniques: surface modelling and solid modelling (Bryden, 2014 p13). Equally, as has been shown in chapter two, several others scholars see polygonal modelling as CAD.

The computer models created using polygonal modelling approaches are the type that can be used directly with additive manufacturing methods. While it is true that traditional CAD tools can now produce models that can be used with these manufacturing methods, it needs to be remembered that this is usually done by converting these models, once the geometry is converted, the resulting polygonal model–or 'mesh'–cannot be edited. Polygonal modelling approaches on the other hand offer an entire range of options to work with these polygonal models. Now that 3D printing is becoming pervasive, polygonal modelling approaches and polygonal modelling tools should be re-introduced in the curricula.

5.3 Teaching Methods/Strategies

According to the results of the study, the majority of institutions base the instruction of computer-related courses on a project-based learning approach. This approach usually involves having students make 3D models. The outputs of these modules are usually renderings of 3D models and/or rapid-prototypes of these 3D models.

In the case of institutions that follow a project-based learning approach, a common strategy is to start with instructor-based tutorials, in some cases these tutorials are preceded by one or more introductory lectures, after which students work on a series of exercises. In some cases, the number of assignments is higher, in which case assignments increase in complexity. In this case, once students complete a series of tutorials, they apply their knowledge to their own designs. One respondent for example, mentioned that exercise is based on "the student's sketch." In some other cases institutions reported having only one single assignment developed throughout the term, in which case this assignment is of higher complexity and was often related to the design studio. A small number of institutions also reported to include tests as part of their teaching strategy.

One of the few institutions that provided additional details regarding the teaching approach followed in these courses reported following an "inquiry-based learning strategy." This strategy involves research and having students develop original solutions to complex design challenges.

Another institution following an integrated approach in which computer skills' learning objectives are integrated into a User-Centred Design module presented its teaching strategy as follows:

Active learning experiences where ongoing feedback is provided weekly in all on campus engagements such as interactive lecture sessions, studios and CAD computer labs. The CAD computer workshops are provided to support students in the development of their designs for discussion, and later, presentation in studio. Prior to each CAD computer workshop students will be required to prepare questions for the mentor in relation to the design projects they are working on.

It should be pointed out as mentioned earlier, that at least one institution reported following a 'command by command' demonstration approach.

5.4 Use of computer-mediated education

The results show that institutions do not make effective use of online and computer-mediated education. This is reflected on issues such as not giving online sources to students, not having developed online tutorials, and not having systems to grant students credit when taking online courses.

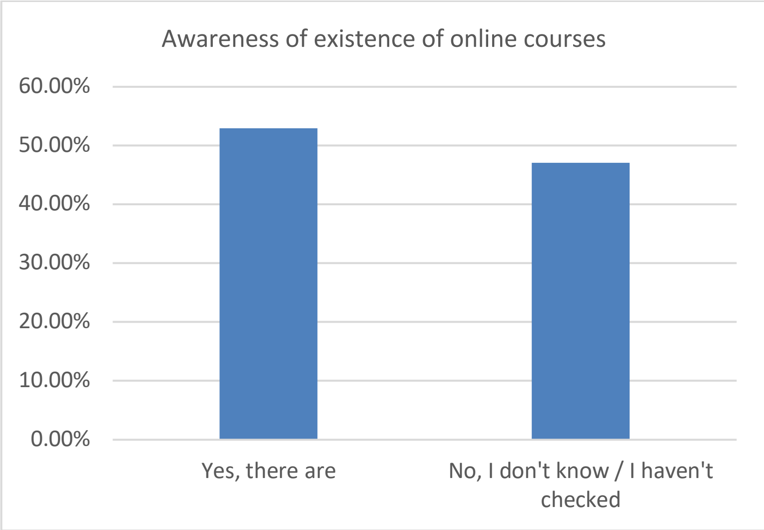

Figure 5.16 Percentage of respondents aware of the existence of online courses covering the same topics taught by them.

Unver (2006) has shown that computer-mediated education can be successfully used to deliver CAD instruction following a blended-learning approach using virtual learning environments–VLEs–like Blackboard (Unver, 2006). However, as can be observed in figure 5.17, more than half of educators are aware that there are teaching materials available online that cover the same topics that they teach. However, as shown on figure 5.20, the majority of educators have not instructed their students to take an online course, and of those who do, they do not fully rely on online sources, but simply use those as reference.

Here, the speed at which instructional material is being created and made available online has to be considered. Universities may not have faced serious competition from online education providers by 2007 as Stallard & Cocker (2001) predicted, but their prediction may have been off just for some years. As discussed in chapter four, the amount of teaching materials specifically tailored for industrial designers on websites like Treehouse, Plural sight, Tynker, Code academy, Udemy, edX, Coursera and more, has grown exponentially over the past decade.

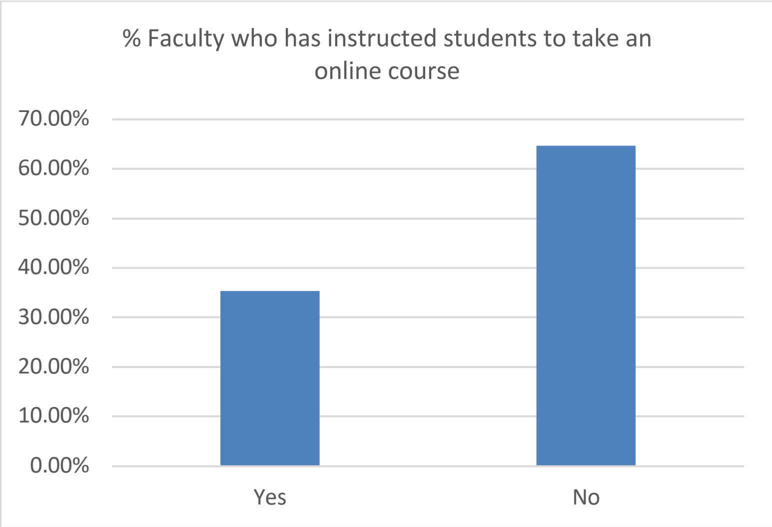

Figure 5.17 Percentage of faculty who has instructed students to review/take an online course.

A possible explanation for the lack of use of online education, is the fact that the majority of educators consider that the instruction they provide is better than that available online, as seen in figure 5.17. However, there is a substantial number who also agreed that theirs and the instruction available online where of equal quality.

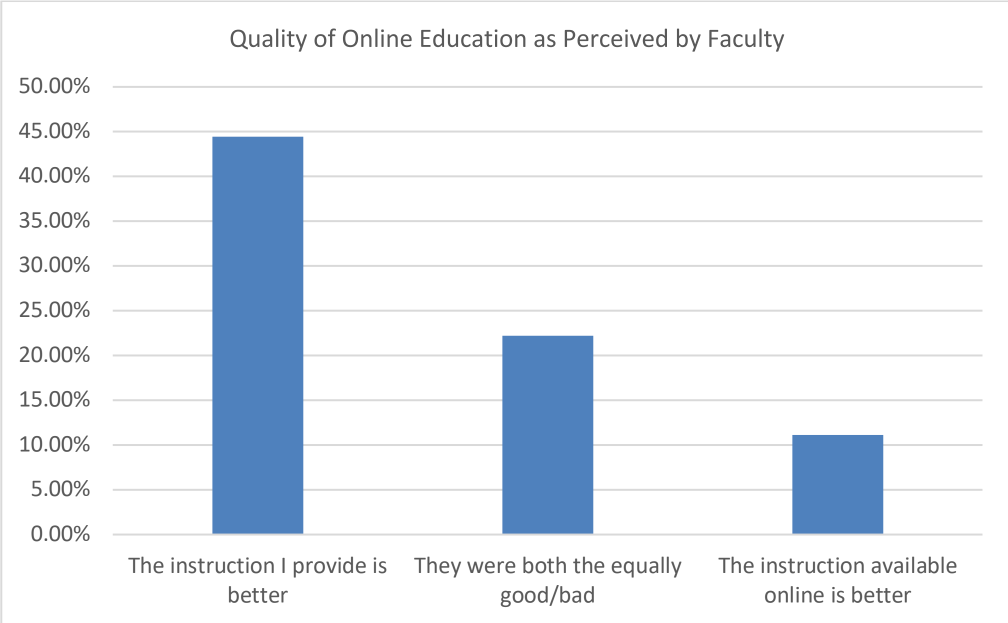

Figure 5.18 Opinions of participants in terms of the quality of onilne teaching materials.

Another element that perhaps has limited the disrupting potential of ICTs is the issue of over-representation carried out by those who profit from selling these technologies. Stallard & Cocker (2001) acknowledge that to a large extent "educators have tried to move forward with bad advice from industry experts," and often succumbed to their "very slick sales techniques" (Stallard & Cocker, 2001, p. 62). Sometimes this over-representation has come from scholars 'with great authority' such as Seymour Papert. Oppenheimer (2003) similarly recalls that Edison predicted 'the end of books,' and 10 years earlier he claimed that with film in education, it would be possible to "achieve one hundred percent efficiency" (Oppenheimer, 2003, p. 3). This situation has led to a chain of "false starts" and an endless cycle of technological upgrades which have made some educators sceptical (Oppenheimer, 2003, p. 18).

The results of the survey also show that the majority of educators have not developed teaching materials to be distributed using computer-mediated approaches. For example, no institution reported using pre-recorded tutorials specifically for the purpose of teaching computer skills. While one did mention the use of online tutorials, it is not clear whether those tutorials were developed by the institution or whether they were just some of the many tutorials available online.

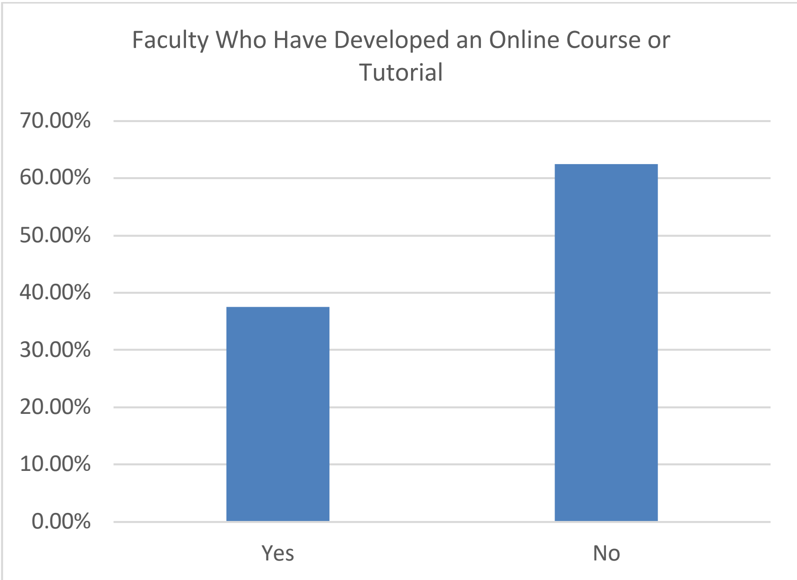

Figure 5.19 Percentage of faculty who have developed an online course or tutorial.

Another result that serves as an indicator to know the extent to which online resources are currently being used, is the percentage of institutions that grant credit to their students when takin an online course. As figure 5.19 reveals, the majority of institutions reported not having a strategy in place through which their students are granted proper credit if successfully completing an online course.

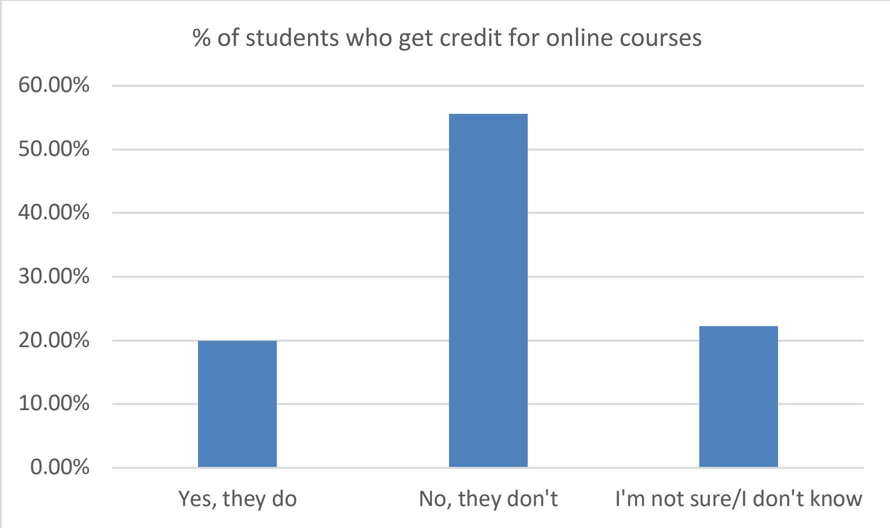

Figure 5.20 Percentage of respondents who replied knowing whether students get credit when taking an online course or not.

The fact that institutions fail to make extensive use of computer-mediate resources, means that blended-mode learning approaches are also seldom used. Consequently, the benefits of computer-mediated education are not exploited. These results are consistent with the approach/strategies reported by participants, in which the use of online/computer-mediated resources is seldom mentioned. They are also consistent with results from a survey conducted in Hong Kong in connection with these studies and which revealed that ICT-mediated education in is substantially under-used.

This is not to say that courses aimed towards building computer skills should be taught completely online, there is a common consensus among educators that personal relations are a fundamental aspect of learning and education (Mahoney, 2001; Mateli, 2012). People interact with one another and engage in critical discussions from which they develop a sense of self-awareness through observation, analysis and reflection. It is through these interactions that meanings arise and learning occurs (Gilchrist, 2001; Crosby, 2001; Chomsky, Gardner, & Della Chiesa, 2013).

5.5 Chapter Summary

This instruction remains very much focused on developing the computer modelling and visualization skills of students, and not much more. In fact, it remains focused on developing just the 3D modelling and visualization skills of students. Other learning outcomes having to do with working with two-dimensional models or with other skills not related to computer modelling were far less common.

Twenty years after McMahon and Browne (1998) noticed that CAD had not found much application at the early stages of the design process, the results of the study show that most schools continue approaching the instruction of CAD as back then. For example, although free-hand drawing computer tablets have extensively penetrated the market over the past two decades, most schools still do not include digital sketching as part of their Computer Aided Design study plans.

While this is not necessarily surprising, what is more interesting are some of the details of how the development of these traditional computer skills is being approached in the curriculum.

As discussed in chapter two, in general, three-dimensional computer models can be classified in three broad groups: Polygonal models, Solid models, and Surface models, each of which has its own advantages and limitations, and each of which requires of its own modelling approach. The results show however that very few schools teach their students to work with Polygonal models. There is in fact, a reason why Polygonal models are not used in Industrial Design, and that is that it is not possible to use them to product moulds, such as those used in traditional manufacturing methods like Injection Moulding. As it will be discussed in the following chapters however, in view of the trends discussed in chapter four, Polygonal models will continue to gain prominence in in Industrial Design. Moreover, the results also show that a good number of schools only teach students to work with either Solid models, or Surface models but not both. As discussed in chapter two, there is a general acknowledgement, that both; Surface and Solid modelling are necessary computer modelling skills in the case of Industrial Design.

There is no definitive evidence that says that the different modelling approaches should be learnt in any particular order, but there is a general acknowledgement that solid modelling is more difficult than surface modelling. Therefore, a logical progression would be to start with Surface modelling and then move on to Solid modelling, however the results show that most schools are going the other way around. This can obviously have an impact on the effectiveness of learning.

Most schools seem to be starting the development of computer skills in the first year. If most learning outcomes are about 3D modelling as mentioned earlier, then that means most students are starting to work with 3D modelling in their first year. This means that, at best, students only have one term–usually one semester–to get familiar with physical model-making before jumping into 3D modelling. When students are not properly acquainted with physical model-making, they struggle with computer modelling, because they have not developed yet a good sense for the attributes of the form: weight, scale, mass, proportion, etc. Consequently, their computer models are often times out of scale for example.

There seems to be a rather strange trend in which most schools either have two or four courses devoted to the development of computer skills. If most programs are four years long, as shown in appendix ii, that means that in the case of those schools that have only two courses or one, the development of computer skills stops at some point. Therefore, in these cases there is no continuity in the growth of these skills, or, it is left up to the student.

An issue that although was not really part of a question, but which could somehow be 'seen through the answers' of some participants, is that there may be issues with que qualifications of teachers. The results show that there were a number of instances in which the same person gave contradictory responses. For example; in one question participants had to say what type of modelling approaches did their students learn? Then, in a later question they had to choose the tools that they used to facilitate the learning of the different modelling approaches they had selected, and in a number of instances, the responses did not match. One way of explaining that, is that perhaps, there is actually not a solid understanding on the part of the lecturers of what the differences are.

This finding (issue with teacher's qualifications) can be confirmed if one reviews carefully the literature about CAD out there. For example; in a 2013 paper published in Procedia SBS, the author seems to confuse Surface and Polygonal models, furthermore classifies Autodesk's 3DS Max software as a 'surface modelling' program.

The results of the study show that neither the title/definition of the study program, nor the location of the institution hosting it have a significant impact on the overall approach that institutions follow towards the instruction of computer skills. In consequence, neither is a good indicator to determine how an institution approaches such instruction. Descriptive statistical analysis of surveyed institutions revealed that most programs are hosted in a school of design, followed by programs hosted in a school of engineering and then programs hosted in school or architecture, this was a rather unexpected finding because, as it has been discussed earlier, it could be said that industrial design is a discipline that branched-out of architecture. More details about this analysis are available in appendix II.

Chapter Six

6 EXTENDED/NEW COMPUTER SKILLS

Having presented the results regarding traditional computer skills in the previous chapter, this chapter presents the findings regarding computer programming and generative design. Implications of these findings are discussed against the background of the literature review, the opinions of educators and experts acquired from qualitative methods, and the context presented in chapter four. The discussion also draws from the previous' experience of the author as industrial designer and industrial design lecturer. The chapter also discusses opportunities to improve the instruction of the computer skills.

6.1 Objectives

The results of the study show that despite most educators support the incorporation of computer programming in the curriculum, the majority of institutions do not to offer opportunities for students to acquire this skill. These results are consistent with the skills and tools that most institutions reported to foster in chapter five. They are also consistent with the overall objectives of computer instruction that institutions reported to have, and with the objectives that most educators said such instruction should have. The difference in opinions of participants regarding the need to incorporate computer programming in industrial design education can be appreciated in figure 6.1

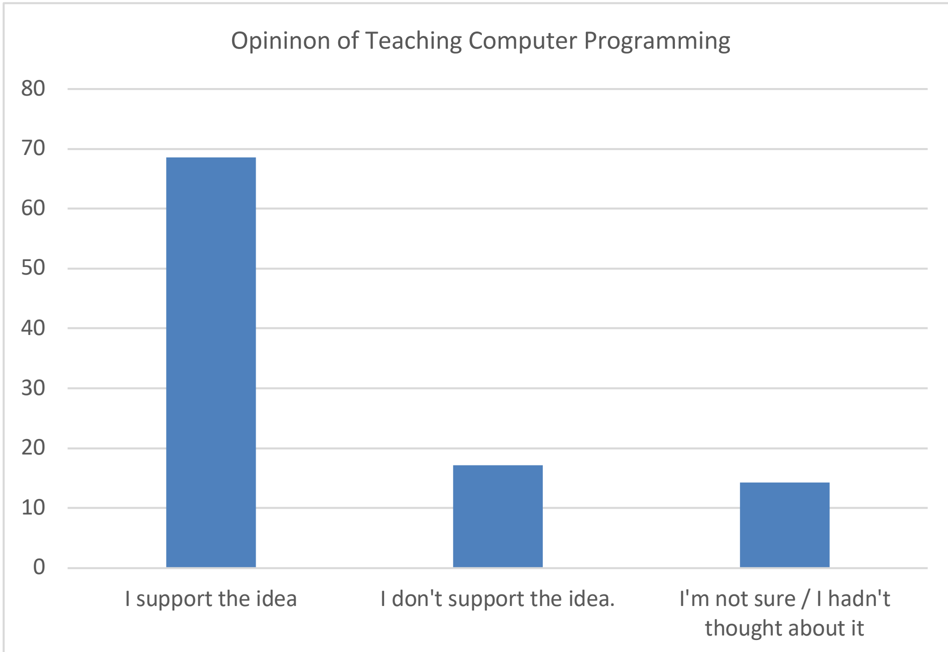

Figure 6.1 Opinion of participants regarding the need to teach computer programming.

Out of the few institutions that reported supporting the incorporation of computer programming, in some other cases the support was made on the basis of the connection between computer programming and working with programmable controllers/microprocessors such as Arduino. One respondent said: "Yes: We teach processing and programming of Arduino. As we embrace technology in order to produce interactive prototypes."

These arguments, to an extent reflect an awareness of the connection between computer programming and computational thinking, although not precisely articulated in those terms. As an example, another respondent said that computer programming "Strengthen students' logical ability." They also reflect an awareness of the connection between computer programming and generative design. Both of these will be discussed in greater detail later.

Those participants who replied not supporting the idea of incorporating computer programming into the curriculum, provided arguments such as that there is not enough time available, and/or that computer programming "is for computer science students, not designers. Designers need tools, not to create them." Perhaps, part of the problem is that ready-made computer programs 'mask' the fact that; behind every icon in a CAD tool, there is a computer program as Krish (2013) and McMahon & Browne (1998) recognize. If seen this way, computer programming becomes more relevant; the question then is whether to use existing computer programs or to make them.

Figure 6.2, shows the percentage of educators who support the incorporation of computer programming versus the percentage of institutions that actually incorporate this instruction into their curriculums.

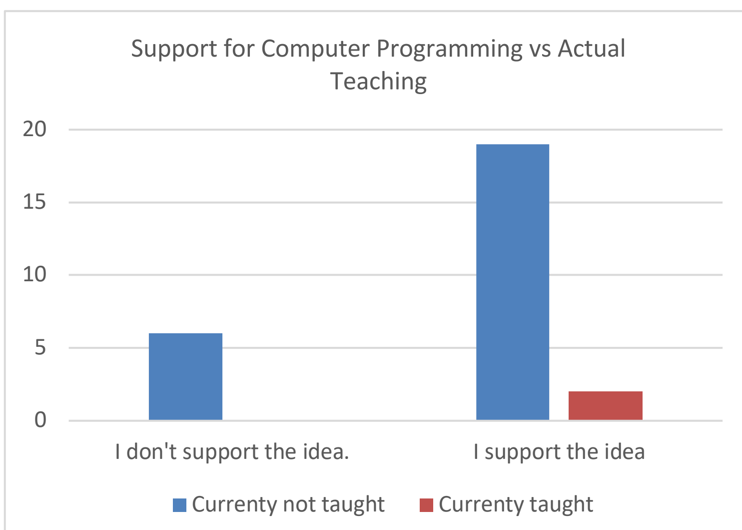

Figure 6.2 Respondents who support incorporating comptuer programming vs actual percentages of institutions teaching computer programming.

In some cases, institutions that reported not teaching computer programming also reported teaching their students how to work with Arduino (fig 6.9). Since working with Arduino is highly likely to involve some type of computer programming, an explanation is that that perhaps some educators are unaware of it. That would explain why they do not identify this as involving computer programming and/or as a course through which students are already learning computer programming.

In contrast, the pertinence of computer programming has been discussed/acknowledged for long in architecture schools. Chase (2004) for example shows that although in some cases diminishing, computer programming has been part of the architecture curricula in many schools, and in some cases, it has been used as an instrument to teach generative design since the 1990s (Chase, 2004 p692). Sivam (2010) too, acknowledges that many architecture students struggle over whether or not they should learn computer programming (Sivam, 2010). This is a question that is only starting to emerge in industrial design.

In addition, computer programming has been highlighted as a creative endeavor that fosters problem solving and inventiveness (Vaidyanathan, 2012), valuable skills for design students. The ultimate objective of incorporating computer programming into the curriculum is not so much to have students build software themselves, but to help them develop a sense of computational thinking. This was expressed by some of the interviewees who said: *"Computational thinking is far more important than programing. They* (students) *should know what they can do with computation, no need to be experts at it, but should understand what can be done. Only those interested need to learn* *more."* However, as discussed in chapter four; computer programming is the best vehicle to develop computational thinking skills.

As discussed in chapter four, there is an increasing number of creative projects that could not have been achieved without their creators employing some sort of computer programing. Moreover, the pioneers behind these projects have usually been people from creative disciplines who have sought to acquire computer science knowledge in order to take advantage of the computer medium (Boden, 1998; Boden & Edmonds, 2009; Cornock & Edmonds, 1973). This supports the pertinence of teaching of computer programming in design disciplines.

There is similarly, an increasing array of tools to aid designers with advanced modelling, which depend on designers having good notions of computational thinking and computer programming. Context Free for example, provides a useful online tool to do generative design, however it can only be used by typing code.

Miles Berry points out that coding is not incorporated in elementary education because the world needs more computer programmers, but because computer programming is the vehicle to develop computational thinking skills. That in turn allows students to understand and change the world (Berry, 2015). Since digital design technologies rely on software, and as more of the tools industrial designers use, and as more and more of the products they create involve digital technologies, computer programming can allow students to not just access a wider variety of tools, but also to understand the products they are designing better. Having a sense of computational thinking and basic notions of computer programming can make a difference in helping designers to make more effective use of computers (Norman, 2015; Shein, 2014). During a visit to the school of design of the Hong Kong PolyU in December 2014, Don Norman ratified the need to bring design education closer to computer science.

6.1.1 Computational Thinking

The results of the study show that most institutions do not have Computational Thinking as an objective of their curriculum (figure 6.3). In this respect, the shift from ICT towards computer science that has been taking place in basic computing education does not seem to have permeated yet into higher education of industrial design. This may be partly due a lack of awareness about the trend amongst educators.

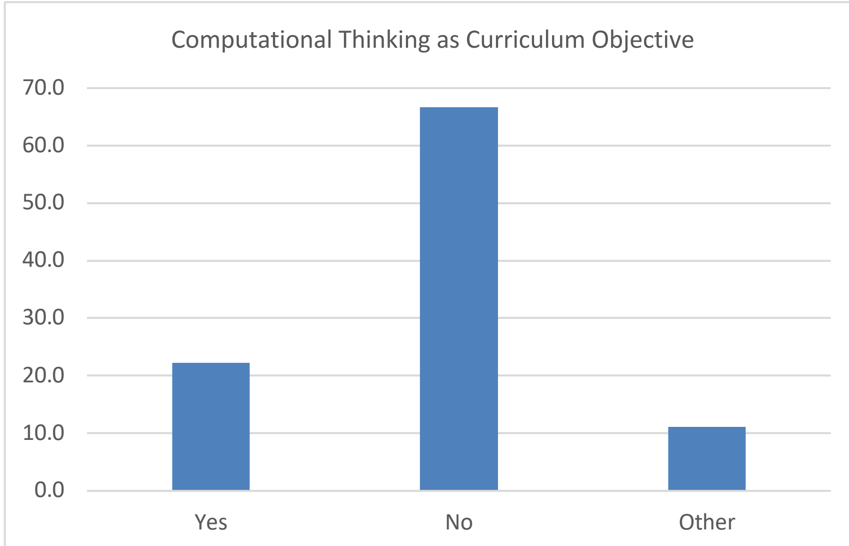

Figure 6.3 Percentage of institutions that reported having computational thinking as a curriculum objective

As can be seen in fig 6.4, computational thinking tended to appear as a curriculum objective more frequently in study programs located in America or in Western Europe. However, most of the few institutions that reported that their students learn computer programming were actually located in Asia. A possible explanation is that perhaps, some participants understood the term computational thinking in different ways—such as meaning: to develop computer skills, for example.

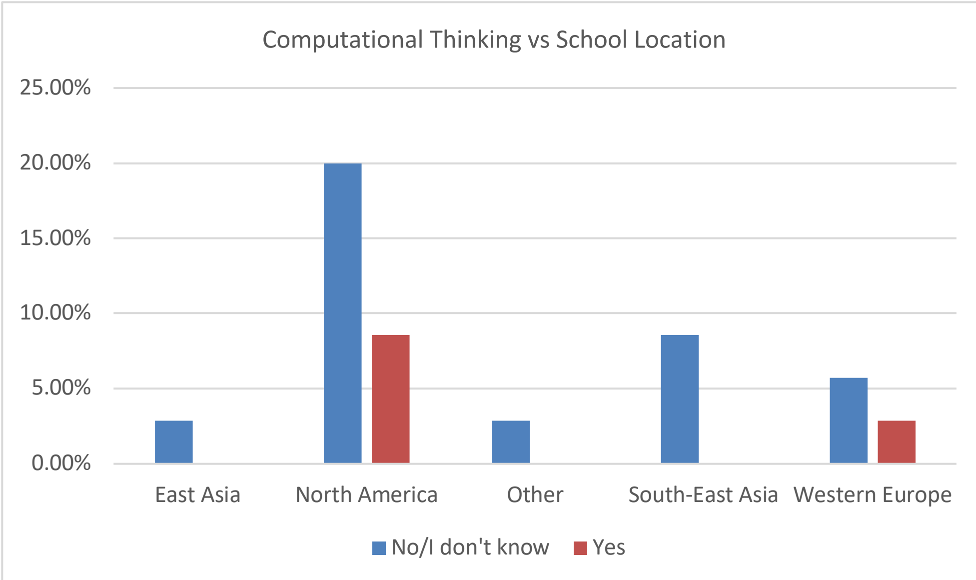

Figure 6.4 Computational thinking as curriculum objective vs location of institution

6.2 Approach

Out of the small number of institutions that reported to incorporate computer programming in their curriculum, there is a clear majority of cases in which this is done through an elective course (figure 6.5). This however, means that all students gain exposure to the subject. Wagstaff (2012) argues that in order to truly favor the creative economy, computer programming must be part of mainstream education. Similarly, to truly have an impact, computer programming should be part of the mainstream curriculum. Similarly, Murray & Perez (2014) point out that, *"the clearest articulation of an educational institution's approach to digital literacy manifests in its curriculum"* (Murray & Perez, 2014).

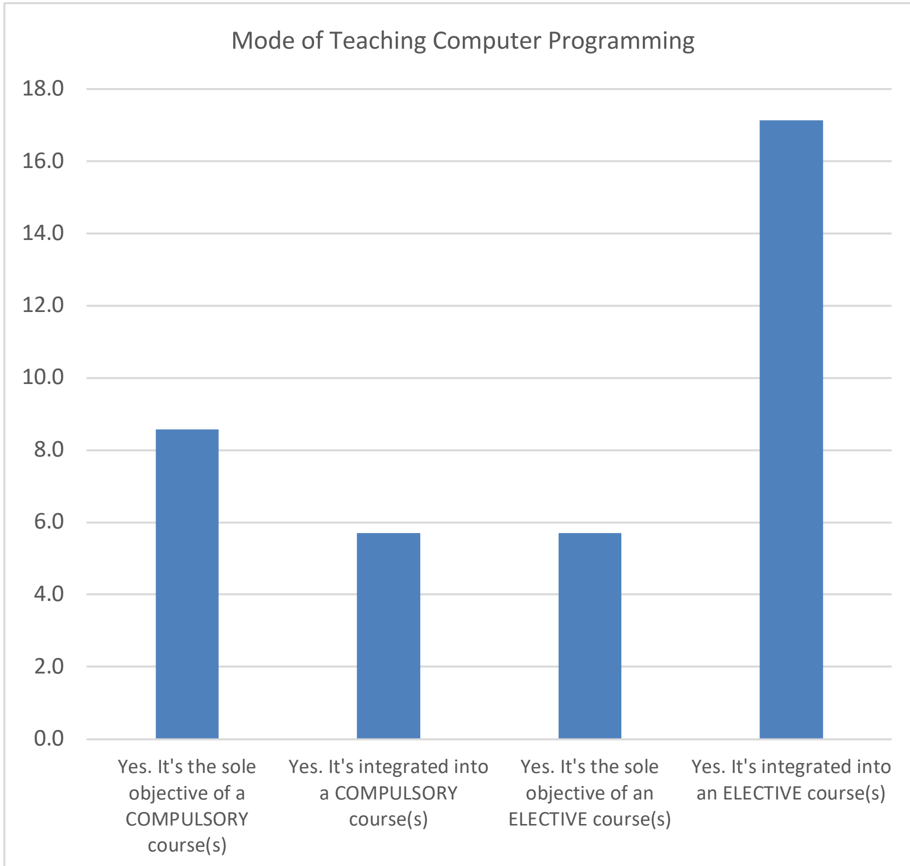

Figure 6.5 Mode of integrating computer programming into the curriculum.

Figure 6.6 shows the percentage of institutions that incorporate computer programming into their curriculums, versus the percentage of educators who support the incorporation of computer programming. These percentages are shown for each of the different modes in which this instruction is actually integrated into the curriculum.

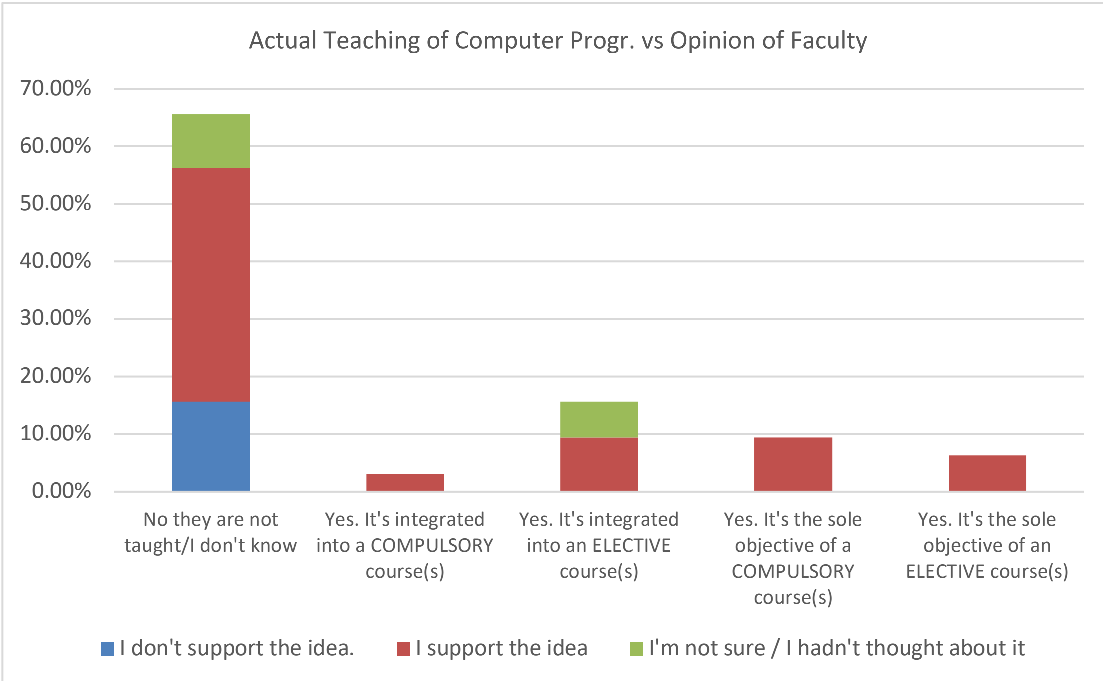

Figure 6.6 Actual teaching of computer programming vs support of respondents.

As can be seen on figure 6.7, out of the small number of institutions that incorporate computer programming into their curriculums, the most common programming languages taught are languages related to web technologies–such as HTML, and Flash–followed by Processing. This information is consistent with that of figure 6.8, in that programming languages currently being taught, reflect the opinions of educators regarding which languages they think should be taught–with the exception of 'Arduino.'

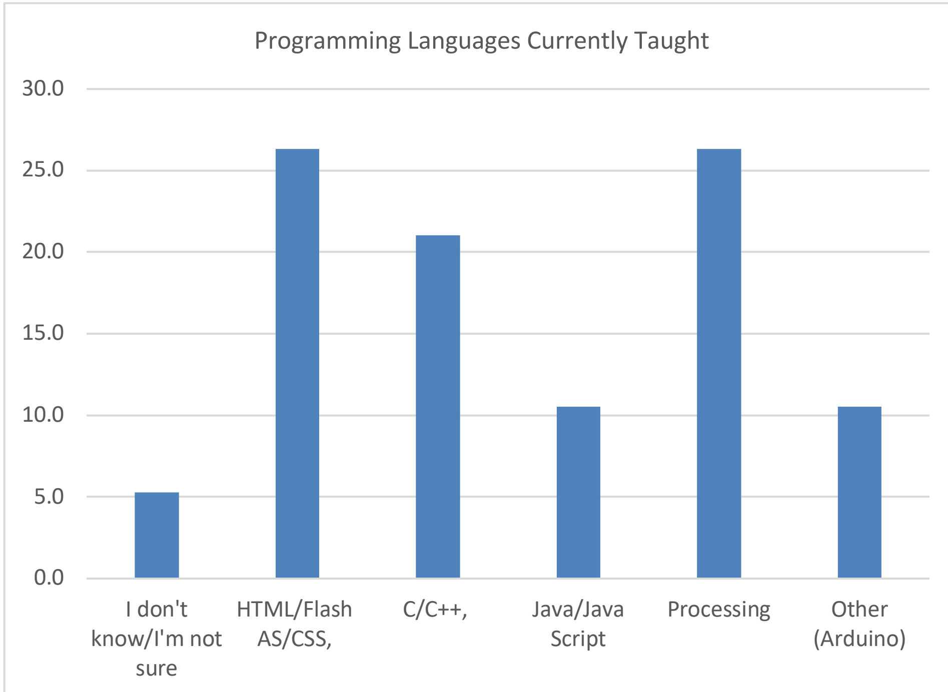

Figure 6.7 Computer programming languages currently being taught.

These results however, contrast with the rationale given by participants who said they support having computer programming as a learning objective of the curriculum discussed earlier. While HTML and Processing are the programming languages most frequently recommended by educators, the rationale they provided to support the incorporation of computer programming, is mostly related with it being an instrument to support advanced modelling approaches and/or working with microprocessors like Arduino. While processing is undoubtedly a tool for creative programming, languages around web technologies—such as HTML and flash—do not fit the rationale provided by participants. Moreover, neither of these languages is a good candidate for working with the APIs of common CAD packages in industrial design. An API, or Application Programming Interface, is), is a set of computer code provided by the vendor of a software (the application) and which can be used to manipulate that same application or a process within it automatically. These APIs are usually based on languages like C++, Visual Basic, LISP, Python or Java.

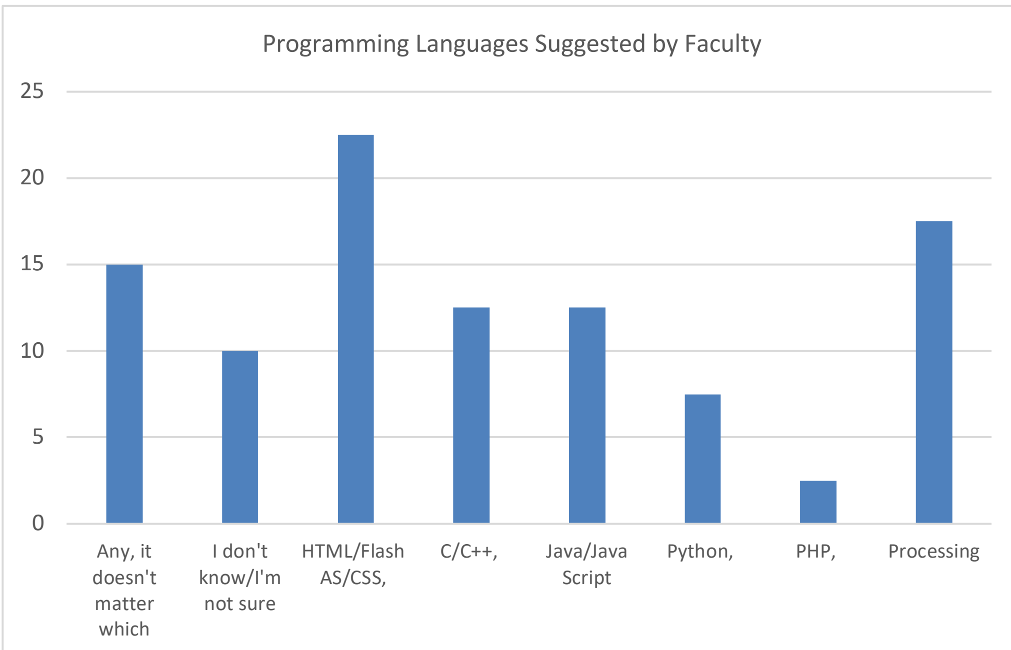

Figure 6.8 Programming languages suggested by participants as options to be incorporated into the curriculum.

If the main objective of incorporating computer programming in the curriculum is to help students gain computational thinking skills, since computer programming is considered the main vehicles to achieve this objective, it is important to consider the selection and number of computer programming languages to be included in the curriculum. As one of the interviewees noticed: *"Just as learning foreign languages helps you see patterns in the usage of language in general, learning different programming* *languages (preferably from very different families) help you understand how solutions to problems can be expressed in many different ways."*

6.3 Generative Design

As with computer programming, the results show that only a small percentage of institutions have learning generative design as an objective of their courses. In addition, the few institutions that do, aim mostly to provide a basic exposure awareness, rather than at developing a level of expertise. One respondent provided the following rationale for not supporting the incorporation of generative design methods in the curriculum.

> My initial opinion is that I don't support the idea–of incorporating generative design. Often students rely too much on digital tools at undergraduate level. This opinion is based on considering current computational tools as I believe students become masters of the tool but do not become well educated on when and how they should use the tool.

As discussed in chapter four, digital mediums like those opened through computers, have "*freed the image from traditional concepts of representation. We no longer represent discrete shapes in the conventional paper -based sense"* (Oxman, 2006 p39). This is having an impact on notions related to classic forms of representation, such as static space, and introduces new concepts, such as dynamic space. Generative design can be a good vehicle to develop spatial and transformational thinking, and its tools can be useful aids to teach these and other design principles to new design students (Chase, 2004 p691-p696).

Not only studies about computer applications in architecture date back to the early 1960's (Gun, 2012). Varinlioglu et al. on their part, point out that over the last decade, digital technologies including algorithmic design tools have changed and evolved traditional teaching methods in architecture (Varınlıoglu, 2015). Krish, on his part acknowledges that until recently, generative design has mostly been in the domain of research labs, and mostly in architecture or art disciplines (Krish, 2010). One of the respondents expressed this trend as well:

> Generative design is taken a little more seriously in architecture, which has a greater intellectual tradition than product design courses dominated by stylists. Product design schools and product designers are not that open to new ways of thinking about how they design. Please read my discussion in core77 you will discover how much they hate any form of thinking different to their own.

6.3.1 Tools

According to the results, the number of institutions that incorporate generative design in their curriculums is very low, thus, it is not possible to identify trends regarding the use of tools. However, the input from interviewees provided some useful insights; One for example, noticed that generative design tools based on 'patch-based flow-control environments'—also known as 'visual programming languages'—such as Grasshopper and the new version of Nodebox, do not require that designers know how to code, however, a sense of computational thinking is highly advisable. As seen in chapter four, in this type of tools **algorithms can be arranged/programmed using a visual interface rather than by typing code (Whitley, 1997).** In addition, these tools use 'reactive programming,' in which each action has an immediate reaction, providing the type of feedback that designers are used to when modelling using traditional mediums.

Based on their user interface, a classification of generative design tools is proposed in figure 6. can be classified in tools featuring visual programming interfaces or VLPs, tools in which the user needs to type code, and tools that feature a graphical user interface–GUI–where different parameters can be easily adjusted.

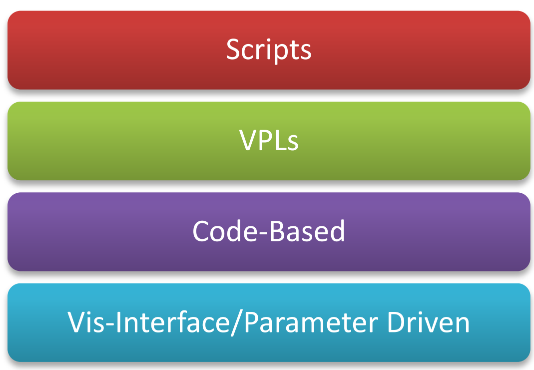

Figure 6.9 Proposed classification of generative design tools

These tools have a considerable advantage to engage 'visual thinkers'–category on which most designers fall according to Gardner's theory of multiple intelligences–and have greatly helped designers to explore generative design. As this expert says, these tools create an experience "less like solving a Rubik's cube and more like shaping a pot," however interviewees acknowledged that despite being more approachable, these tools are still intimidating for designers precisely due to a lack of computational thinking.

In general however, the consensus among interviewees, was that while it is not necessary to know computer programming to do generative design, it is highly recommended. Chase (2004) in addition, recognizes that a more effective use of generative design tools could be achieved if students acquired foundations in computer programming (Chase, 2004 p697). Tools that allow designers to do generative design without requiring coding skills are more limited in the outputs they can produce. Conversely, more powerful tools do require the designer to have some knowledge of coding. Another expert articulated it this way:

> …it depends on what you want to do, so one cannot say that it is always necessary. What is clear is that if you do not know how to code, it will be very difficult for you to innovate. When you are also limited by the capabilities of a software, you are limiting the questions you ask yourself.

The same idea was proposed by another expert who made the distinction between what he categorizes as 'explicit' and 'implicit' generative design. 'Explicit' generative design is that which is based on generating outputs based on 'existing objects' and 'static environments.' While 'implicit' generative design is that which deals with systems that self-generate and involve changing environments and uncertainty. For the first, he said, a node-based flow-control environment is enough, for the second however, it is necessary to code.

Overall, interviewees agreed, that when the main goal is to develop computational thinking skills, rather than taking the fastest route to using, making and/or modifying generative design tools, some programming skills in languages like Python, Java and JavaScript would be recommended. Some interviewees pointed out however, that factors such as the tools and programming environment are probably more important than the particular languages chosen. In this regard, Processing stands out, not only there are extensive learning materials online, but it can even be used in several modes. These modes allow users to type code in any of the languages previously mentioned, plus Ruby.

One of the aims of Processing, was specifically to make it easier for anyone to make interactive art (Runberg, 2015). That alone makes it an attractive option for industrial design, but there is another reason why Processing is an attractive platform in the case of industrial design education; the fact that the programming language and development environment for the Arduino platform are based on Processing. Learning Processing can potentially facilitate working with Arduino and vice-versa. Processing was a second option by educators who support the incorporation of computer programming into the curriculum (Figure 6.7).

One other tool suggested by interviewees as an option for beginning designers, is Context Free (www.contextfreeart.org). Despite being code-based, this tool was regarded as a good option for the generation of randomized outputs (Figure 6.10), however because it is its own language. The programming skills they learn while learning to work with Context Free, are not as transferable as when learning to use other tools such as Processing.

An advantage of Context Free, is that there are plenty of examples online, and "newcomers to programming find that modifying existing sketches is easy and fun." In addition, the system provides almost instant visual feedback. These characteristics make it a viable option to teach generative design in industrial design schools.

Figure 6.10 Example of problem-space exploration done in Contextfree.org (above). Example of generative graphic output created with Contextfree.org (below).

Both; Grasshopper and Processing were the most common tools highlighted as good options by interviewees. In the case of Grasshopper, one of the reasons is likely the fact that it is a plugin for Rhinoceros (Rhino), one of the most popular CAD tools in industrial design. Rhino has also become very popular among architects interested in working with organic shapes and also because of its generative design capabilities when combined with Grasshopper. As one of the interviewees put it; since both architecture and industrial design are mostly concerned with modelling in 3D, and Rhino is a 3D modelling environment, the “Rhino plus Grasshopper provides a powerful entry.”

The previous must be considered in light of the fact that Solidworks is reported to be the most common software tool in industrial design schools. While the API of Solidworks offers many possibilities, there are currently no tools in the form of plugins, that enable Solidworks to do generative design.

6.3.2 Teaching Methods/Strategies

Just as with traditional CAD skills, interviewees agree that teaching generative design methods should be underpinned by the general theory behind them. For this purpose, and just as with traditional CAD, it is important to maintain a balance between digital and manual methods. One of the interviewees for example, pointed out that often times, generative principles can be explained without using computers at all. This confirms Chase’s claim that the teaching of generative design in architecture schools, sometimes focuses more on helping students understand the principles of form generation and the logic behind, rather than on attempting to produce a 'finished design piece' (Chase, 2004 p693).

6.3.2.1 Tinkering

Some interviewees suggested that tinkering would be particularly useful when using code-based generative design tools. The idea is, to find existing scripts code, and then playing around with them and seeing the results. Varinlioglu et al. (2015) have been successful in following this approach in architecture schools. During this process, the rationale behind the results of the modification can be explained.

6.3.2.2 Application Programming Interfaces

Another possibility to combine computer programming and generative design in a way that is relevant to industrial designers, is through the APIs of the most popular CAD packages used in the field, most of which offer this possibility nowadays (Sivam, 2010). This opens the possibility for industrial design students to practice/apply computer programming in a more meaningful way. This however, depends on the selection of computer programming languages taught at schools.

6.3.2.3 Simple textures

As it has been discussed, one advantage of generative design methods is to be able to easily create models with very complex forms 'easily.' However, these models are often detached from "real life constrains" (Krish, 2010), and are often difficult, if not impossible to manufacture using traditional mass-manufacturing processes. While it is true that additive manufacturing processes are slowly increasing the feasibility of these methods, it is also true that only in low-volume production.

While for the purpose of learning generative design it is not necessarily to know the manufacturability limitations of the models created, it is advisable that students are aware of these issues. Consequently, the best scenarios to integrate generative design methods in industrial design education, would be in the design of product—or parts—with little manufacturing and/or ergonomic constraints, such as patterns or textures on small or flat surfaces. It would be then advisable to explain students the manufacturing limitations of the models created, and how these can post-processed in order to make them suitable for mass-production.

6.3.2.4 Sequence/Progression

Based on the insight provided by interviewees, a progression in the instruction of generative design methods can be devised. This progression could start by introducing students to theory and basic concepts, using simple tools. As one of the experts said; "Just getting started is the most important." These tools could range from those based on tweaking a number of controls—

like Autodesk's Shapeshifter—to using tools based on node-based flow-control environments, such as Nodebox or Grasshopper. In the end, students could move on to code-based tools like Processing.

This instruction could be pursued following an integrated or discrete approach, and has the advantage that students would develop first, a degree of computational thinking skills while experimenting with algorithms in tools like Grasshopper. Once students are more confident with the different algorithms/nodes in a node-based flow-control environment, they could start looking into how these algorithms work, and even making their own.

As part of a course at the University of Sussex called 'Creativity Foundation Two,' two groups of foundation year design students of the 2016 and 2017 cohorts, were asked to work on a quick exercise to produce a form that could be used as a flower vase. For this exercise, the tool used was Autodesk's generative design tool 'Shape-shifter.' The output of the students can be appreciated in figure 6.11. Chase (2004) concluded that a good generative design tool for beginning designers should allow to quickly explore a number of designs and generate designs that may not have emerged by using hand—or traditional—methods (Chase, Scott 2004 p696), Shapeshifter has all of these characteristics.

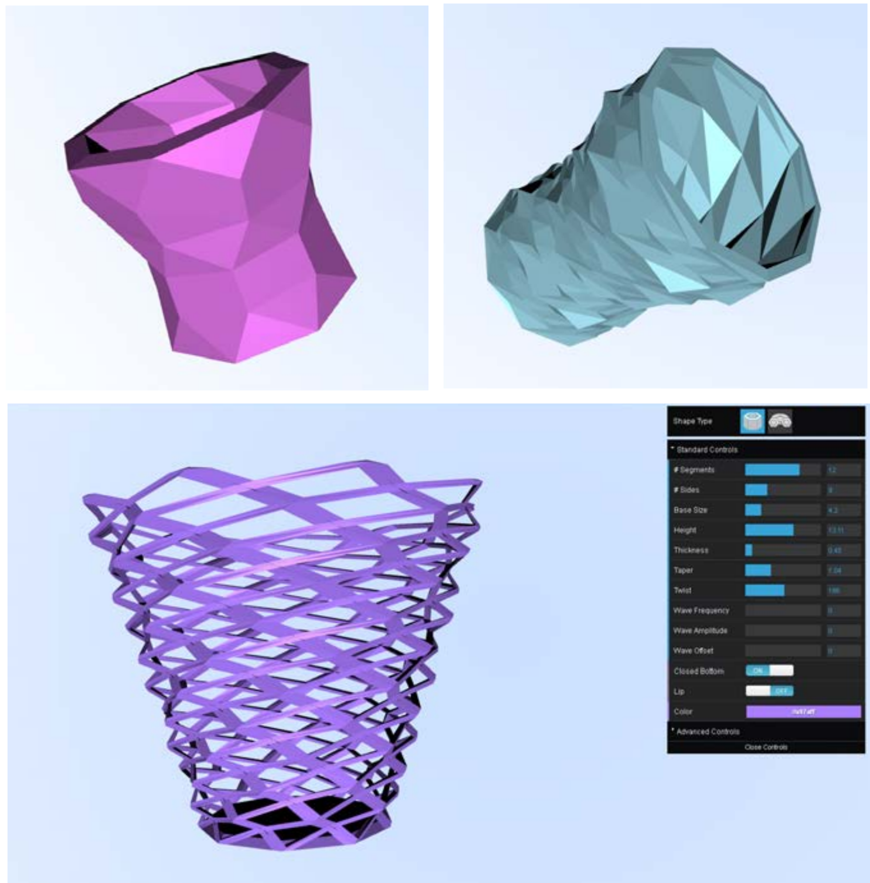

Figure 6.11 Work of students using the 'Shape-shifter' generative design tool.

The exercise was preceded by an introduction to generative design and a quick demo of the tool. After that, students had 30 minutes to experiment with the tool and create their vases. After the exercise students were asked a few questions regarding their experience. All students thought the tool was easy to use just with the information provided. Similarly, all except one, thought the tool was useful, however, two said that they felt limited with the number of options—patters—available in the tool. Nonetheless, they all agreed it would be a useful tool, in the design process, and agreed that if nothing else, they could take inspiration from the output of this process. All students thought that even a short amount of time such as thirty minutes, was enough to get familiar with the tool and produce a simple shape. Another example from a different course is shown in figure 6.12. In this case however, the design of a lamp served as the basis for the brief. This type of result can be easily obtained using scripts for Adobe Illustrator, and large number of variations can be explored.

These exercises, shows that simple tools, can serve to expose students to generative design at an introductory level. These types of tools, also allow students to produce something without previous knowledge of generative methods, however, the exercise also confirms that the simpler the tool, the more restrictive in the variety of models it can produce.

Figure 6.12 Mutually facilitating relationship between computer programming, generative design and computational thinking

6.3.3 Symbiotic relationship

An important consideration, is that computer programming and generative design facilitate each other. They maintain a symbiotic relationship in which generative design facilitates understanding of computer programming and vice versa. And since computer programming facilitates computational thinking, this symbiotic relation between computer programming and generative design, in turn facilitates computational thinking (Figures 6.13 & 6.14). This relationship is supported by media theory, which states that different mediums are often mediums/facilitators of each other (Feenberg, 2012).

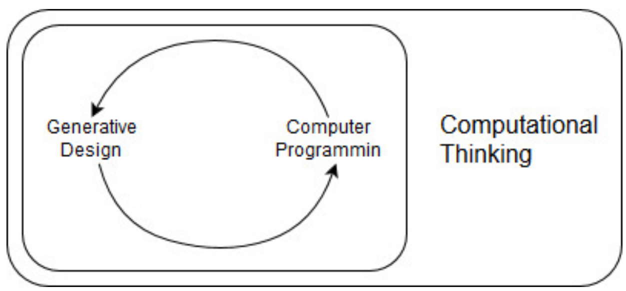

Figure 6.13 Mutually facilitating relationship between computer programming, generative design and computational thinking

One of the interviewees put this relation like this:

> My own take is that data and algorithms are to a generative artist what paint and brushwork is to a painter. Just as a good painter needs to know a lot about paint (where to get it, how much it costs, how to mix it, which

hues you need to create different effects). A generative artist must somehow develop a facility and comfort level with algorithms, with understanding other people's algorithms, with creating her own, with developing a toolbox of techniques to accomplish various effects, etc.

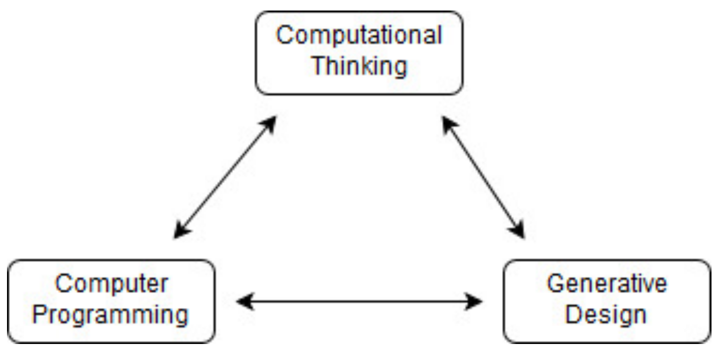

Figure 6.14 Mutually facilitating relationship between computer programming, generative design and computational thinking

As it has been discussed in this chapter, while most experts agree that it is not necessary to know how to code to do generative design, they also agree that not knowing how to code limits designers substantially. If not computer programming specifically, experts agree that a sense of computational thinking is necessary for anyone interested in generative design. As one respondent put it:

The reason seems to be that a certain basic set of computational thinking skills is essential to these tasks. There seems to be no way around it. No clever or magical UI will eliminate the need to decompose a problem into smaller parts, to debug, and to think abstractly in a way that seems foreign to anyone bewildered or traumatized by basic algebra.

This is of particular importance in the case of industrial design education, because one of the reasons why generative methods have not been employed in industrial design as much as in other fields like architecture, is that the products industrial designers create have a different set of constraints. These constraints, often around design for manufacture and ergonomics, require that designers have a higher control of the design process, which in turn requires the use of more sophisticated tools and methods, which in turn, demand that designers know, if not coding *per se*, that at least they have notions of computer programming.

Since computer programming and generative design maintain a mutually facilitating relationship, a generative design exercise can be an ideal vehicle to introduce concepts about computer programming and vice-versa. In addition, Chase (2004) has noticed, that generative design can provide a relevant medium for students to gain exposure to concepts like structure, reiteration, and conditionals, which can help them to develop computational thinking skills (Chase, 2004 p691).

6.4 Chapter Summary

The results of these studies show that only a small number of institutions have computational thinking as one objective of their computer-related education. Computer programming and generative design methods also do not appear as learning outcomes in most cases. This is true in general, and regardless of the number of computer-related courses an institution has.

Comparing the rationale for supporting the incorporation of computer programming in the curriculum with the selection of programming languages that educators suggest, it is unclear whether educators understand why this is important. The reason why computer skills are important in industrial design education, is not because there is a need for 3D modelers, or computer programmers—although it may well be—but because these skills are part of how students explore design, express themselves, and ultimately learn. This however, is a vision about computer-skills education that seems to be more assimilated in other fields like architecture. In contrast, several interviewees recognized that industrial design tends to lag behind, and is a field often less open to assimilating new perspectives.

Advanced modelling techniques like generative design expand the potential of industrial designers to create computer models, but only if students learn how to use them. The variety and complexity of both; computer models and computer modelling approaches available to industrial designers are increasingly higher, since neither needs to be 'passive,' but can automatically change, both; computer programming and generative design are relevant computer skills for industrial design students.

Based on the results of these studies, two conclusions in terms of non-traditional computer skills can be made:

1. A transition towards a more encompassing objective for computer instruction in industrial design education, has in general not taken place, nor there are any indications that this is happening at the moment either.
2. As a consequence of the previous, computer programming has not been set as a learning outcome of most institutions.
3. Generative design methods have not been in general, incorporated in most curriculums, and similarly, there are no indications this is currently occurring.

Based on the results of these studies, it could be argued, that the initial adoption of computers in industrial design, has led to an understanding of CAD that has remained mostly unchallenged, and which tends to reproduce over time (figure 6.15). This can be deducted not just from things like the type of computer skills that schools foster, but also from the definitions of CAD provided by participants. It is beyond the scope of these studies to prove this relation; however, it is fair to assume, that this understanding of computer aided design must have an impact over the overall approach that institutions have toward the instruction of computer skills. As an example, it is clear from the results, that most lecturers do not consider things like teaching students how to program Arduino boards, part of the instruction of computer skills they deliver.

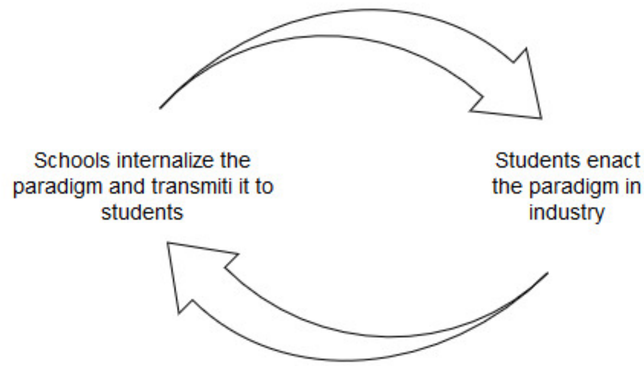

Figure 6.15 How understanding of CAD is perpetuated over time

Even if design education succeeds at training students in traditional computer skills, it cannot be said that it succeeds at training students to make effective use of computers. To improve this instruction, it is necessary to move beyond the traditional understanding of computer aided design, and promote the vision that computers are computation tools, capable of incorporating automation at different stages of the design process. For this, it may be necessary to revisit the lexicon used when talking about computer skills; if for most people the term CAD has a narrow meaning, it may be better to talk about computer skills in industrial design schools using other terms.

Dankwort et al. (2004) acknowledge that CAD education is more than just teaching 'solid or surface modelling.' Students must learn about the complete product development process under the lens of Computer Aided Product Creation. Modern instruction of computer skills in industrial design education should aim to not only equip students with more and better skills, but also to help them understand the full potential of the computer as a tool in the design process.

Because, as philosophers of technology have pointed out, technology frames our way of thinking, it is in a way contradictory, to claim that as opposed to vocational schools, Universities are concerned with 'forming thinkers.' Thus, 'technological education is not important.'

The implementation of standards in education is a hot debated topic. On the one hand, proposals are often met with fierce criticism (Tierney, 2013; Tierney, & Sablan 2014), and at the same time, some academics support the idea that all college students, regardless of their discipline, complete a basic core-curriculum that provides a solid cultural basis (Meacham, 2013). With a core-curriculum standard, only one part of the curriculum is standardized (National Governors Association, 2010). Such a policy can substantially aid in ensuring that instruction of a particular subject is always covered up to a minimum (Platt, 2011). Curriculum standards have a number of advantages, such as: shared expectations, focus, efficiency–teachers can share the same teaching aids and assessments–and lastly, quality (Porter et al., 2011).

The use of standards in education and higher education is in fact not new (Sams, 2013). In the case of design education, most of the best design schools in the US, in fact originally shared a common curriculum; the one inherited from the Bauhaus and later established in Chicago by Lazlo Moholy Nagy (Budd, 2011). The results of these studies show no indication that any standards regarding instruction of computers skills exist in any of the countries/institutions surveyed.

Like in any system, both: private and public higher education must be properly regulated. While freedom and institutional autonomy are undoubtedly important attributes in higher education, intentionally or not, they make it possible for very wide variations in the extent and depth to which they train students in a given area such as computer skills.

The results show that by far, most schools do not teach their students computer programming. This represents a different in comparison to other disciplines like Architecture or Engineering, in which the teaching of computer programming is more common. Consequently, most schools surveyed, do not have computational thinking as a learning objective that the curriculum aims to achieve. Interestingly, most participants support the incorporation of computer programming in the curriculum. This suggest in turn that perhaps there are issues to find qualified teachers, and to develop updated curriculums.

Very few schools teach their students Generative Design as well. In fact, out of the 38 surveyed programs, only one provided more detailed information in this regard. There is a general consensus among experts in the field however, that, while tools that allow designers with little or no knowledge of computer programming to do generative design are starting to appear, to really take advantage of this approach, it is necessary to have a good foundation of computer programming. There seems to be therefore, a mutually facilitating relationship between Computer Programming, Computational Thinking and Generative Design, thus it is hard to get one without the other.

Chapter Seven

7 CONCLUSION

These studies have investigated the current instruction of computer skills in Industrial Design Education with the aim of identifying knowledge gaps and generating insights that can help us understand how well it prepares students and find ways of improving it. The studies set out to: 1) Determine the range of computer skills being currently fostered, the extent to which they are fostered, and how this is translated into the curriculum. 2) Determine the extent to which these skills address challenges and opportunities posed by trends in basic computing education, generative computer modelling methods, rapid prototyping, and computer-mediated education. 3) Develop a framework that can help Industrial Design educators to develop study plans aimed at developing computer skills. 4) Make a series of general recommendations on how to improve the instruction of computer skills in Industrial Design education.

To attain these objectives, these studies have made a revision a number of Industrial Design undergraduate programmes, gathering data using surveys, interviews and other instruments. The insight of faculty on issues such as the need to incorporate computer programming in the curriculum have been identified, and their implications have been discussed against the background of a number of trends mentioned earlier and discussed in detail in chapter four.

7.1 Regarding the Range, Extent and Overall Curricular Approach

The findings of these studies suggest that the instruction of computer skills in Industrial Design education offers room for improvement. The range of computer skills fostered, as well as the approach followed to foster them, have not substantially changed since computers were widely introduced in schools two decades ago. This instruction, focuses mostly on developing computer modelling skills—mostly three-dimensional—and it is integrated in the curriculum using a discrete approach. It could be argued consequently, that schools often teach students how to use CAD rather than computers, this is a mistake.

It has been argued, that the understanding of CAD that prevails in industry and academia, may have a role to play; the definitions of CAD given by faculty, show that the prevailing view, is that 'Design' in 'Computer Aided Design' means modelling. In some cases, course outlines even focus on teaching students how to use a particular software title—i.e., SolidWorks. While it is acknowledged that solid/parametric and surface modelling techniques, are each important computer skills for Industrial Design students, the results show that many schools only teach students one or the other, but not both.

The results also reveal issues with teacher's training; concepts fundamental for the instruction of computer modelling, such as the different types of computer models, and their corresponding modelling approaches, were found to be not of the dominion of faculty. Moreover, the objectives of computer-related courses, were often found to be vaguely articulated and/or unknown to faculty. Thus, there is a risk of course outlines and teaching plans being poorly designed. In general too, this instruction fails to train students to an extent that makes them feel confident to model the designs that they have envisioned in their minds.

Neither the specific denomination of a program, nor the hosting department a program is attached to, were found to be indicators of substantial differences in how the instruction of computer skills is approached.

It was also found, that institutions fall short in providing enough foundations. For example, most study programs do not offer descriptive geometry classes to students, nor it is a pre-requisite for CAD courses. Although positive in terms of exposing students to the benefits of computer modelling,

The results suggest that traditional modelling skills have tended to be replaced by their computerized equivalents. This has deprived students from learning foundational modelling concepts. For example; almost all schools surveyed, have eliminated 2D drafting and/or descriptive geometry from their curriculums, thus, students have less time and instruments to develop spatial thinking skills.

This has important implications for both; learning CAD, and operating without it. The same applies to basic computer skills foundations; it is becoming increasingly acknowledged that Universities have made the dangerous assumption of thinking that just because incoming student spend all day glued to their smartphones, they do not need general ICT education. That is not the case.

This 'substitutional' approach, is like giving basic education pupils a calculator and stop teaching them how to do maths. No one would advise that, not even on the basis of productivity and accuracy. Furthermore, this increases the risk of walking into a technological trap, which, if not carefully considered, can have catastrophic consequences whenever technology fails.

7.2 Regarding Addressing Trends

Current instruction of computer skills in Industrial Design Education, also offers room for improvement in terms of addressing the trends discussed in chapter four. For example; despite that the amount, quality and accessibility of online educational materials–particularly related to computer subjects–has risen exponentially over the past decade, the results show that Industrial Design schools either ignore the existence of these materials, choose not to use them, and/or have not developed their own.

Apart from modelling, the range of computer skills developed in Industrial Design schools, generally falls short in helping students to apply computer in other areas such as Project Management or Design Research. Likewise, while education in other creative disciplines like Architecture, have incorporated computer programming and novel approaches to computer modelling like Generative Design for a long time, the relevance of such content in Industrial Design Education, is just starting to be realized.

The advantages of a blended approach, combining in-classroom and online instruction have been discussed. A blended approach is particularly suitable for computer-related courses, because the medium of delivery and object of study are the same. Computer mediated education, can also aid in mitigating the challenge posed by the big differences in the level of computer skills of incoming students. To take advantage of this approach however, it is necessary to find suitable online teaching materials, and/or, developing them. Moreover, it is challenging to integrate both effectively due for example, the difficulties to prevent students from cheating when completing online tutorials.

Computer programming is important for several reasons, for example; although computer programs specifically developed for industrial designers are widely available and continue to evolve, industrial designers need not be restricted by the capabilities of 'ready-made' programs, they can develop their own, and/or modify existing ones. However, the results reveal that the number of schools that currently teach Computer Programming is low, and the few that do, only integrate it into elective courses. In addition, while academics in Industrial Design schools feel that computer programming is important, they do not know exactly why.

7.3 Recommendations

Based on the results, it is fair to say that the average Industrial Design curriculum prepares students to be computer-modellers, but not to be successful computer users. There is a fundamental difference between a curriculum that seeks to develop computer skills, and one that only seeks to develop CAD skills. Similarly, there is a difference between a curriculum that seeks to develop CAD skills, and one that just seeks to develop computer modelling skills.

The same can be said between a curriculum that seeks to develop computer modelling skills, and one that only seeks to develop skills with a particular CAD software vendor's software. Therefore, the notion that the only type of computer skills that need to be fostered in Industrial Design schools are computer modelling skills needs to be revisited, the same applies to the notion that only certain types of computer modelling techniques need to be taught. These relationships are illustrated in figure 7.1

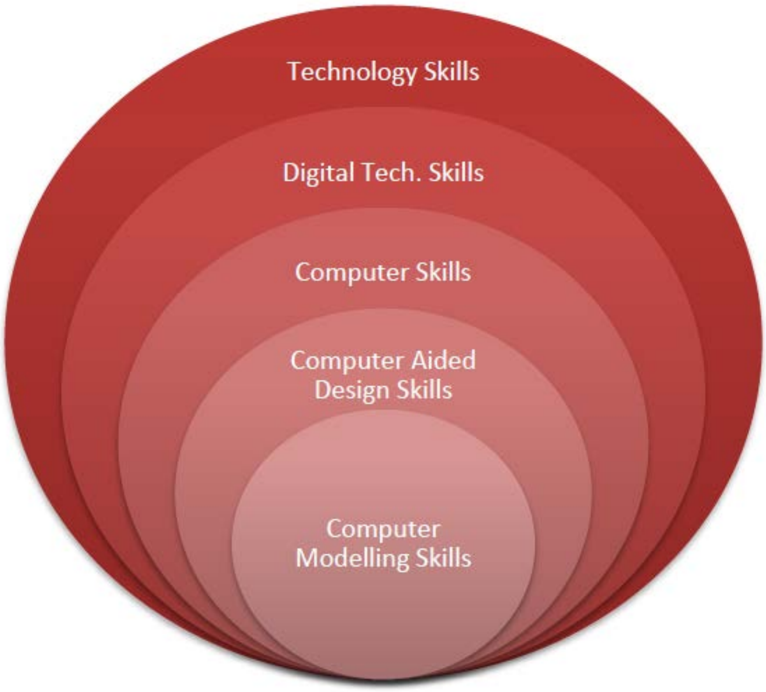

Figure 7.1 Different levels of technological/computer-skills instruction

Industrial design schools need to develop a broad, formal and systematic technology education strategy (figure 7.4). Based on this, the level and scope of computer skills that they aim to develop should be determined.

7.3.1 Traditional computer skills

Skills not related to computer modelling, but relevant for other stages of the Industrial Design process, such as design communication–i.e., animation, video-editing or photo-retouching–need to be included in study plans.

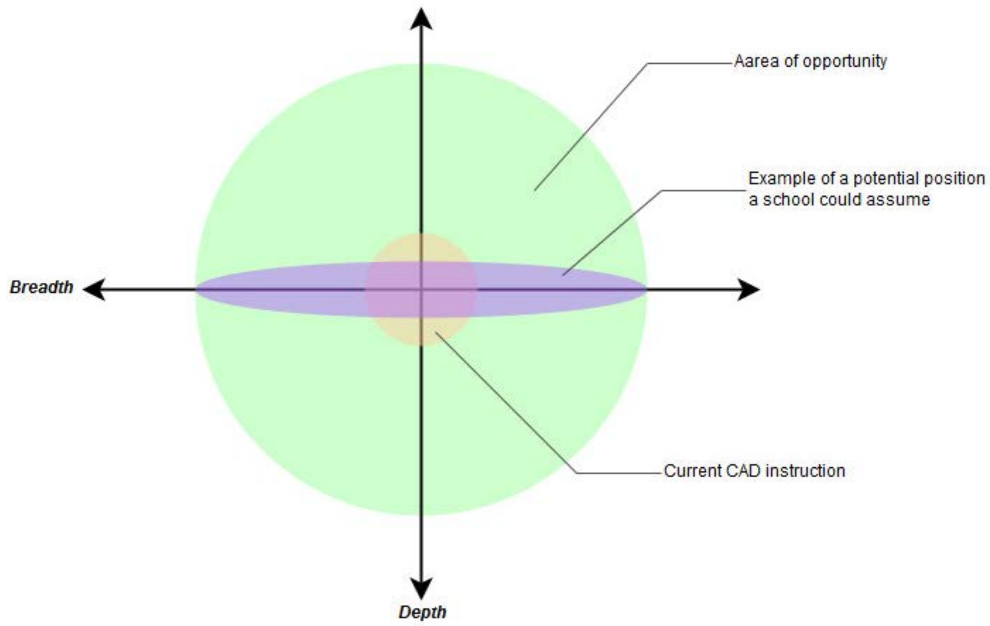

Figure 7.2 Expanding computer skills in Industrial design education

This expansion can take place in two dimensions as illustrated in figure 7.2. The vertical axis–representing depth–and the horizontal–representing breadth or range of different skills, while the green area represents the areas of opportunity for expansion. The purple are exemplifies the position that a particular school could assume, and the Orange represents the extent of skills currently covered in an average school.

7.3.1.1 Reverting the substitutional approach

Because many of the changes in the curriculum implemented in the past, have replaced the instruction of traditional skills for computerized versions, it is necessary to revert those changes to ensure that students acquire proper foundations.

For example; re-introducing Descriptive Geometry, to give students more opportunities to develop spatial thinking skills. The same applies to Orthogonal Projections and Physical Model-making. Ensuring that students make physical models first, helps them understand basic design concepts such as the attributes of form in 'the real world.' Afterwards students can move on to 3D CAD environments and apply their knowledge in a virtual environment.

7.3.1.2 Provide exposure to all 3D modelling approaches

Proper traditional computer modelling education in Industrial Design schools, should include exposure to digital free-form sketching, 2D drafting and illustration, surface modelling, solid modelling, parametric modelling, polygonal modelling, and 3D rendering. This core instruction should also give students the opportunity to practice the acquisition of 3D models, such as with 3D scanning.

7.3.1.3 Provide specialization paths

Traditional computer skills education, should also provide specialization opportunities in areas such as Finite Element Analysis, Animation, and

Project Lifecycle Management (Figure 7.3). These specialized courses can come in a variety of forms such as electives, and could be delivered using an in-classroom, online or a blended approach.

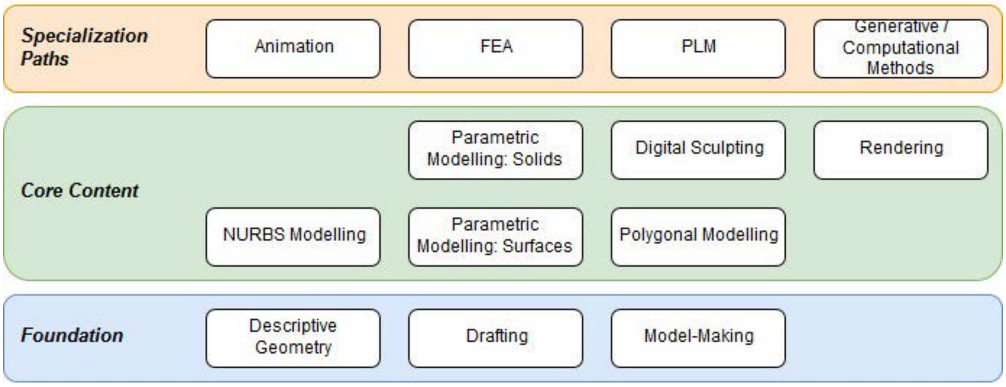

Figure 7.3 Example of traditional computer skills implementation in a curriculum

It has been discussed that, while Polygonal models and polygonal modelling, have been neglected in Industrial Design in the past due their incompatibility with traditional manufacturing methods, advances in Additive Manufacturing are starting to rival the feasibility of traditional mass-manufacturing methods, particularly in low-volume production.

7.3.1.4 Generative Design

Improving computer skills education in Industrial Design schools, implies not only expanding beyond computer modelling, but also within. The results of these studies however, show that the majority of schools only teach students to work with digitally hand-crafted or 'passive' computer models. Schools should offer students the possibility to work with modelling techniques such as Generative Design. Some possibilities could include: design briefs involving java scripts to create 2D patterns/textures, or using simple 3D tools specially developed for this purpose–such as those described in chapter six–to create models of simple products like a lamp shade or a flower base.

It has been argued that, since Computer Programming and Generative Design maintain a mutually facilitating relationship, a design project involving Generative Design can be an ideal vehicle to introduce concepts about Computer Programming and vice-versa.

7.3.2 Integration Approach

The instruction of computer skills can be integrated in the curriculum following either a discrete or, an integrated approach, or a combination of both. Each has advantages and disadvantages; in the case of the first; the main drawback is that instruction can be de-contextualized, and thus become 'meaningless,' and leading to poor student engagement. In the case of the second, the specific learning outcomes in terms of computer skills can be overlooked within the overall objectives of the course. Moreover, the assessment of these outcomes, can be neglected.

It is important therefore, that for each of the computer skills that schools aim to foster, the right integration approach is devised. In general, an integrated approach in which students learn computer skills around their studio projects is ideal.

7.3.3 Computer Programming

In a landscape in which the race for innovation is so intense, and in which products increasingly rely on software, computer programming has emerged as a basic literacy in the twentieth-first century, and a desirable skill for university graduates from all disciplines. Since Computer programming is also the best route to develop Computational Thinking skills.

Just like basic computer education has shifted from teaching pupils ICTs, to teaching them computer science, Industrial Design Education ought to embark on a similar journey, in which the aim is not just to develop computer modelling skills, not even CAD skills, but at its deepest level, Computational Thinking skills. At the moment however, Computational Thinking is not an objective in most schools. Once again, this paradigm shift has slowly been taking place in other creative fields like Architecture for a while, but not so much in Industrial Design.

7.3.3.1 Shift towards Computational Thinking

If Computer Programming is the best vehicle to achieve Computational Thinking, having it as a curriculum objective is of the utmost importance. Some ways of doing this, is: incorporating the use of code-based Generative Design tools into study plans, using 3D CAD packages through their application programming interfaces (APIs), and using modelling tools based on computer coding such as OpenSCAD. There seems to be an agreement amongst experts that it is not necessary to code in order to do simple Generative Design, however, there is a recognition that coding skills are necessary to do more sophisticated work. Another possibility, is to have students work on projects that involve developing electronic hardware, such as in the design of electronic or interactive products, and integrating the use of programmable logic controllers like Arduino.

Making this shift may require reviewing the terminology used to refer to the instruction of computer skills. Trying to change the perception around the term CAD may not be feasible, instead, it may be that it is actually better to resort to other terms such as 'Computer Aided Product Development,' or simply 'Computer Skills' in order to start conveying the idea that this instruction is about more than just traditional computer modelling. This does not mean of course, that skills like computer modelling should be neglected, but rather, that instruction of computer skills needs to aim for more than it currently does.

Summary of recommendations:

1. Develop an overall computer skills education strategy, with clear objectives at each of the levels shown in figure 7.1 At the 3D modelling level, at the CAD level, and above.
2. Develop a comprehensive curriculum, growth of computer skills all the way from beginning to end, and which offers opportunities for specialization. This curriculum could rely on computer-mediated education.
3. Carry on with the trend of basic computing education, and make a similar shift from teaching students how to use software (ICT), to teaching them how they can use computers and computation to achieve their goals through computer science and computational thinking.
4. Ensure that students have proper foundations. This in turn requires:

a) Either assessing and/or providing general ICT knowledge.

b) Re-introduce in the curriculum content that has been replaced rather than enriched, and which is necessary for the development of computer skills–i.e., Descriptive Geometry and Technical Drawing.

c) Leaving enough room/time for students to get acquainted with non-computerized modelling/design processes.

5. Strengthen basic Computer Modelling Skills by:

a) Expose students to all three basic computer modelling approaches–Solid, Surface and Polygonal.

b) Expose students to advanced computer modelling approaches like Generative Design methods.

6. Facilitate the learning of Computer Programming, by using project briefs involving Generative Design.

7. Look closely at teacher's training/qualifications, and re-consider the terminology used when discussing Computer Skills education. If the term CAD has acquired the connotation of being just about computer modelling/3D modelling, it may be better to use other terms to avoid the trap of developing study plans that only focus on those areas.

7.4 The PRODISST Framework

The implementation of the previous set of recommendations can be facilitated with the use of a framework. This framework is proposed in figure 7.4 and can be applied at any of the levels that appear in figure 7.1, from the 'macro' level of overall technology skills, to the 'micro' level of specific computer modelling methods. This framework can serve as a guide to help Industrial Design educators develop and/or review curriculums.

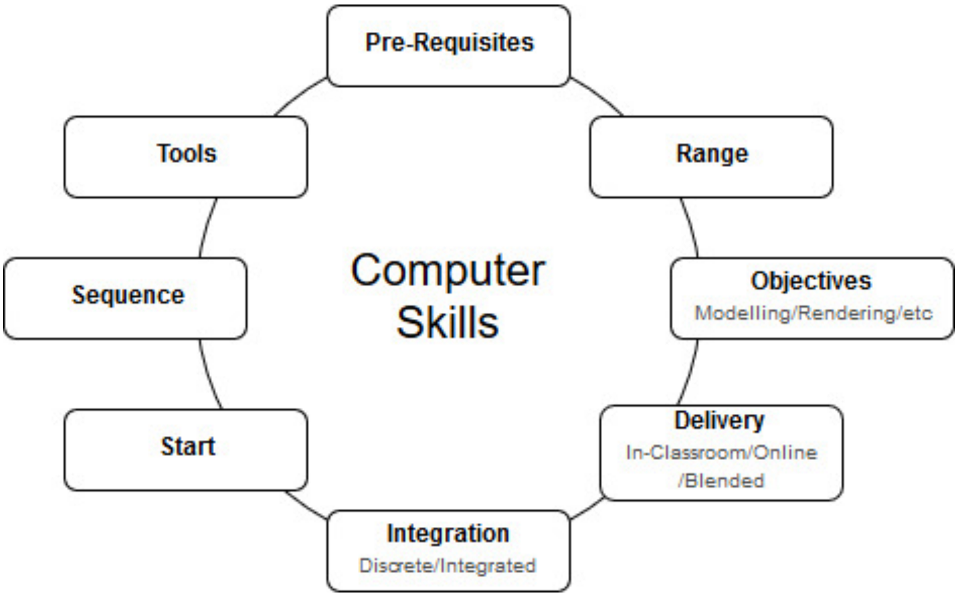

Figure 7.4 PRODISST Model to introduce computer skills in Industrial Design programs

The framework has all the elements that need to be considered when putting together study plans. It can help educators to ensure that none of these elements is left out, and to think about the different options each can be best approached. The first element in the framework are the Pre-requisites, which refers simply to the knowledge that students need to have before starting with the development of their computer skills. Range refers to the different skills that students are supposed to learn. Objectives, refers to what exactly, about each of those computer skills students are supposed to learn. In other words: to what depth they should learn each computer skill. Delivery refers to whether the instruction is delivered using a traditional classroom setting, online or using a combination of methods. Integration refers to whether the instruction is included into another course–generally the Design Studio Project–or, whether it is a separate course. The Starting Point refers to the time–in the study plan–in which each of the different computer skills is supposed to take place. Sequence refers to the order in which each of the different skills should be learnt, and finally, the Tools refers simply to the selection of computer tools that will be used to facilitate the learning (fig. 7.4).

At a low level of the computer skills education strategy–3D modelling for example–the Range, could include skills like Polygonal Modelling and Solid Modelling. At this level, the Objectives in each case could include; learning how to increase and decrease the density of the mesh–in the case of Polygonal Modelling–and learning how to use Boolean operations–in the case of Solid Modelling. At a higher level–i.e., General Computer Skills–the Range could include skills like Computer Modelling and Computer Programming. At this level, the Objectives in each case could include; learning about conditional operators–in the case of Computer Programming–and learning about 2D and 3D Modelling–in the case of Computer Modelling.

The combination of these factors will vary depending on the conditions of each school and/or program. For example; a school could choose to follow a discrete integration approach, a classroom delivery mode, and basic objectives. Another however, could choose to follow an integrated approach, using classroom delivery methods, and advanced skills as objective.

While this framework is developed specifically with the development of computer skills in Industrial Design schools in mind, it is possible that it could be partially applicable in other design disciplines and beyond. It is likely, that study plans developed with this framework, would be less prone to show some of the issues found with current study plans, and that it would also make it easier to implement some of the recommendations previously presented.

7.5 Limitations

As the studies unfolded, finding the right focus and scope of the research emerged as the main challenges. At the extremes, at one point the study aimed to research Digital Design Technologies in general, while at another, nothing more than 3D Modelling. As the process evolved, this was inevitably reflected on the overall research aims, objectives and research questions. This in turn, had several repercussions; at some point the studies where more philosophical, at another more pragmatic. Eventually it became clear that what the studies really were looking at, was a range of computer skills. It was partly due to this realization, that two broad groups of computer skills were identified, namely Traditional and Non-traditional.

The studies relied on information gathered from academics in Industrial Design schools. Other stakeholders however, were less taken into account. This means that the results reflect the opinion of academia disproportionately. Likewise, while schools from different parts of the world have been consulted, certain regions, are unequally represented. An effort has been made to minimize this issue with the triangulation of data, however, the results, may not evenly represent, or apply to all parts of the world in general.

In addition, a substantial amount of empirical data was collected using surveys. Part of the questionnaire was tested during the pilot study; however, a number of questions were added afterwards, and there is always the risk that a participant does not understand or misunderstands a question. As an example; it is likely that many survey respondents did not understand what it was meant by 'Computational Thinking.'

There are as well, other novel modelling approaches, such as the so called 'digital clay,' and/or 'virtual-reality 3D modelling,' whose learning in Industrial Design education should be studied, and which these studies did not look into.

7.6 Contribution and Future Work

Within its limitations, it is fair to say that the studies do contribute to answering the questions they set to answer, and have achieved the objectives they aimed at. More specifically, the studies are believed to contribute in the following ways:

1. By generating knowledge by:

   a) Providing a current, detailed, and for the most part non-existent, account of the computer skills being developed in Industrial Design education and how their development is approached through the curriculum.

   b) Collecting and analysing the insight of experts about Generative Design.

   c) Giving examples of how that insight can be applied, and how Generative Design can be approached in the Industrial Design classroom.

   d) Collecting and analysing other relevant information, such as level of computer skills of incoming students.

2. By highlighting relations between closely intertwined trends in Education and Technology, which are relevant for the development of study plans aimed at developing computer skills within Industrial Design Education.

3. By shedding light upon past research, confirming and questioning previous findings, and showing whether issues highlighted in the past have been addressed or not. For example; and in contrast to Dankwort et al's. (2004) claim that education of computer aided design tends to be broad but superficial. The results of these studies suggest that computer education is in general not only superficial, but also narrow and traditional.

7.6.2 What next?

A most immediate future line of research, is to test the implementation of the recommendations and framework proposed. For example, exploring the scenarios to integrate Computer Programming, and/or testing the different Computer Programming languages that have been identified as the best options for students to learn. Further research could also look into exploring the possibilities of using the APIs of software tools to do Generative Design. There is also a need for studies that address whether there is an optimal sequence when learning the different types of traditional computer modelling methods–surface modelling, solid/parametric modelling, polygonal modelling.

Another possibility, is to look into the long-term consequences of technological adoption. Computer skills education does more than building technical skills, technology frames our way of thinking, thus, it is important to look at this in greater depth. It has been discussed for example, how institutions can change the software tools they use over time. This raises questions about the actual motivations for such changes, and the relations between educational institutions and the creators of technology. The dangers of 'hidden technological agendas,' such as technological exclusion for example, can be mitigated with the use of an increasingly expanding and powerful array of free, open-source tools, however, this requires of research to test the actual capability of these tools.

Another issue which deserves further attention, is the important differences in the level of Computer Aided Design skills of incoming students. This is particularly problematic in countries where universities tend to receive students from different types of secondary education systems. Therefore, it is important to study the issue in greater depth, and try to come up with ways of solving it. This problem is exacerbated by increased student mobility in recent years, as well as Universities' increased dependence on tuition money, which often results in lax admission requirements.

7.6.3 Shifting Field or Not

In chapter one, the position of these studies in before the wider discourse of Design Education was presented. It should only be added, that at the end of the studies, that position is held. The creation of 'intangible products' or services is not, the central concern of Industrial Design. But if there are in fact, other design disciplines for which such products are the central concern, even in such cases, the development of computer skills–and even computer modelling skills specifically–is of crucial importance. If anything, the types of models used in those cases would be of a different kind.

7.7 Chapter Summary/Final Remarks

The relevance of computer skills in Industrial Design education goes beyond mere efficiency, often a tool can be difference between achieving a goal or not. For the simple fact that modelling skills are essential in Industrial Design, and much of the modelling process is now carried on in the computer, the computer skills that industrial designers acquire in college are no matter to be overlooked.

Since computer skills education does more than building technical skills, providing students with broad and deep computer skills, is to expand their thinking. The skills taught in Industrial Design schools, as well as the approach followed to teach them, suggests that a revision of such instruction is necessary. We are now moving towards a stage in which being able to modify and/or create their own computer technologies can be an important asset for industrial designers in the workplace. As we move into these technological futures, skills like computer programming are becoming essential.

Edward de Bono explains the power of technology to expand human potential using a bike as example (de Bono, 2012). Like any other technology, computers can help Industrial Design students 'reach further,' but only if they know 'how to ride them.' Universities have the responsibility to address the development and assessment of the technological skills of their graduates; there is no justification to neglect pursuing a better computer education.

Far from being a thing of the past, the instruction of technological skills in general, and of computer skills in particular, must be a constant concern in

Industrial Design education, always in evolution. It is thus hoped, that these studies are a contribution in this respect.

Appendices

APPENDIX I (List institutions surveyed)

For more information regarding the profile of participating/responding faculty please see Appendix II.

ASIA	
Institution	Country
Tongji University	China
Hunan University	China
SHU-TE University	Taiwan
Central Academy of Fine Arts (CAFA)	China
Nanjing University of the Arts	China
National University of Singapore	Singapore
MIT Pune	India

OCEANIA	
Institution	Country
Univ. of Tech Sydney (UTS)	Australia
Swinburne Univ. of Technology	Australia
Auckland University of Technology	New Zealand

EUROPE	
Institution	Country
Middle East Technical Univ. (METU)	Turkey
Univ. of Limerick	Ireland
Oslo School of Arch. and Design (AHO)	Norway
Oslo and Akershus University College (HiOA)	Norway
Lund Univ.	Sweden
Univ. of Twente	Netherlands
Bournemouth University	UK
University of Huddersfield	UK
Strate Ecole de Design	France
Creapole Ecole de Design	France
Pforzheim University	Germany
Loughborough Univ.	UK

NORTH AMERICA	
Institution	Country
University of Cincinnati	USA
Rhode Island School of Design	USA
Art Center College of Design	USA

Auburn University	USA
Virginia Tech	USA
College of Creative Studies	USA
Stanford University	USA
Georgia Tech	USA
Purdue University	USA
BYU (Brigham Young) University	USA
Savannah College of Art and Design	USA
Massachusetts College of Art and Design	USA
California State Univ. Long Beach	USA
University of Notre Dame	USA

APPENDIX II (Profile of participants and their institutions)

All values expressed in percentages unless otherwise noticed.

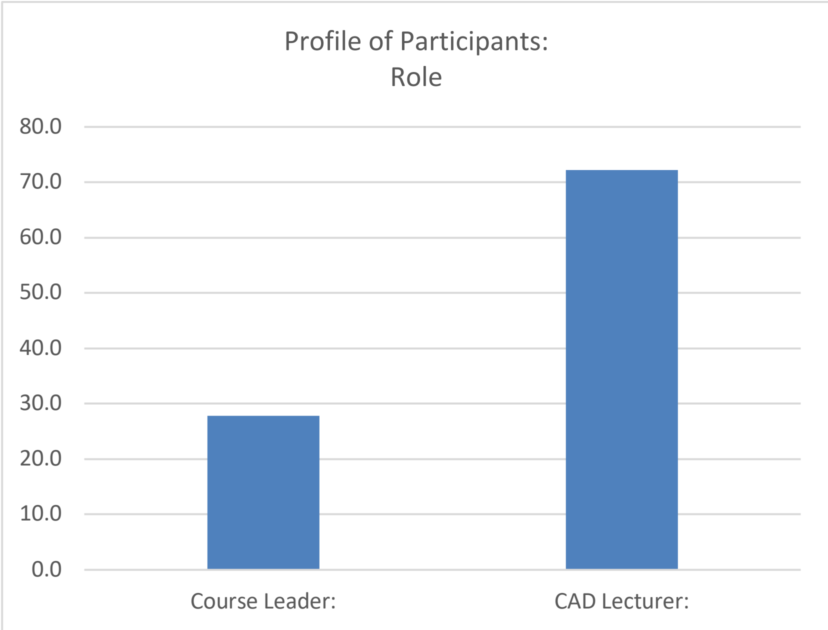

Figure A2.1 Role of participants

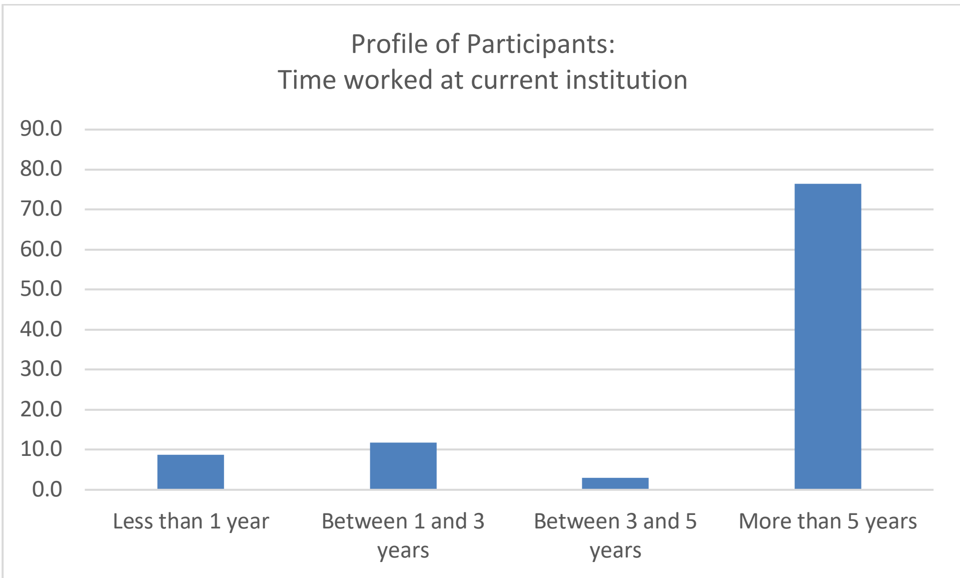

Figure A2.2 Teaching experience of participants

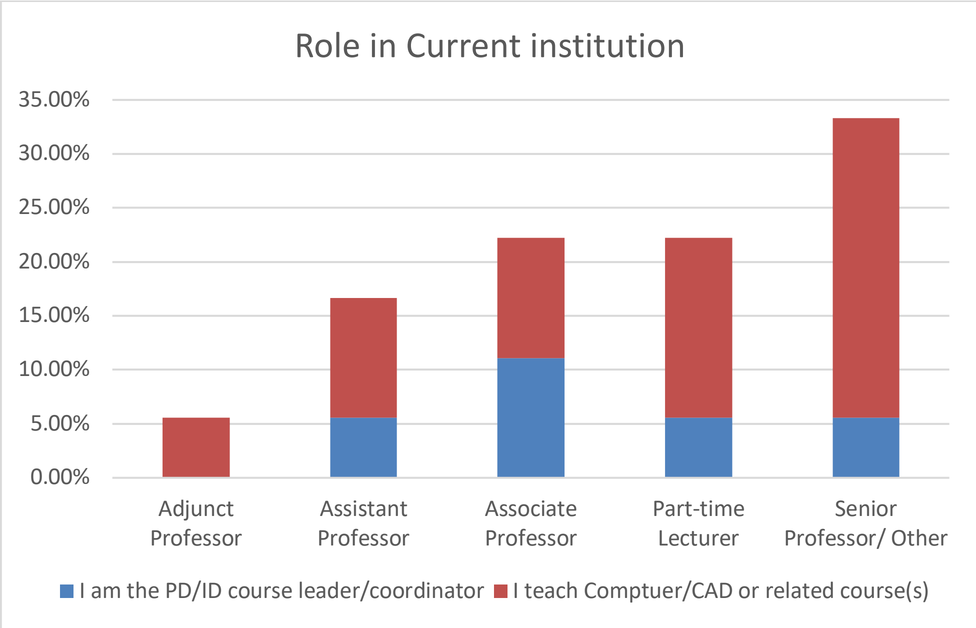

Figure A2.3 Type of appointment of participants vs their role

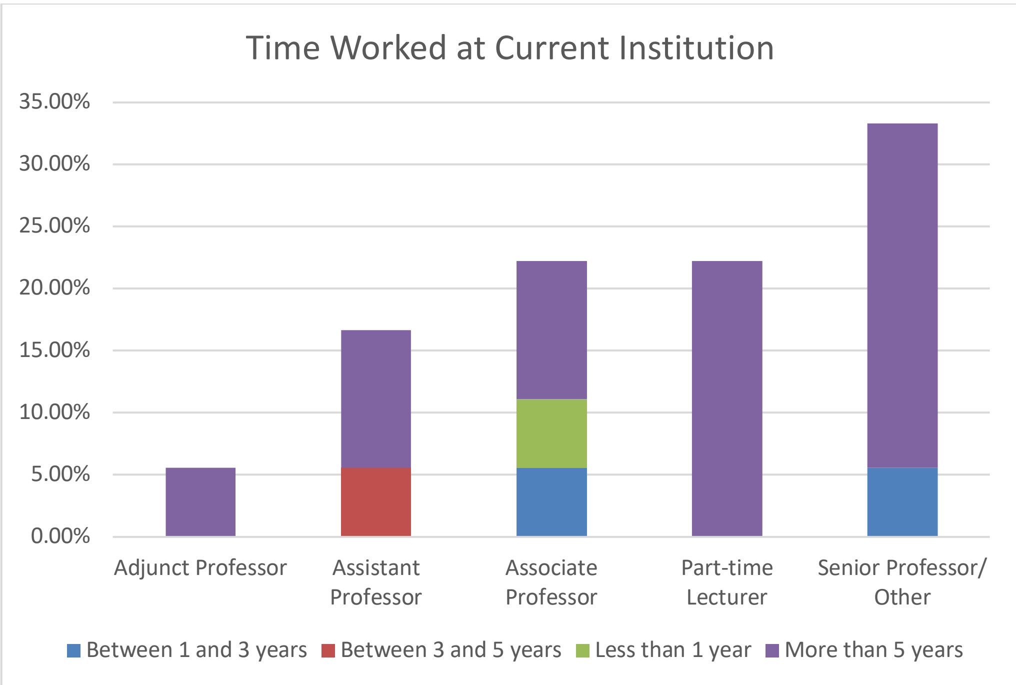

Figure A2.4 Type of appointment of participants vs their teaching experience

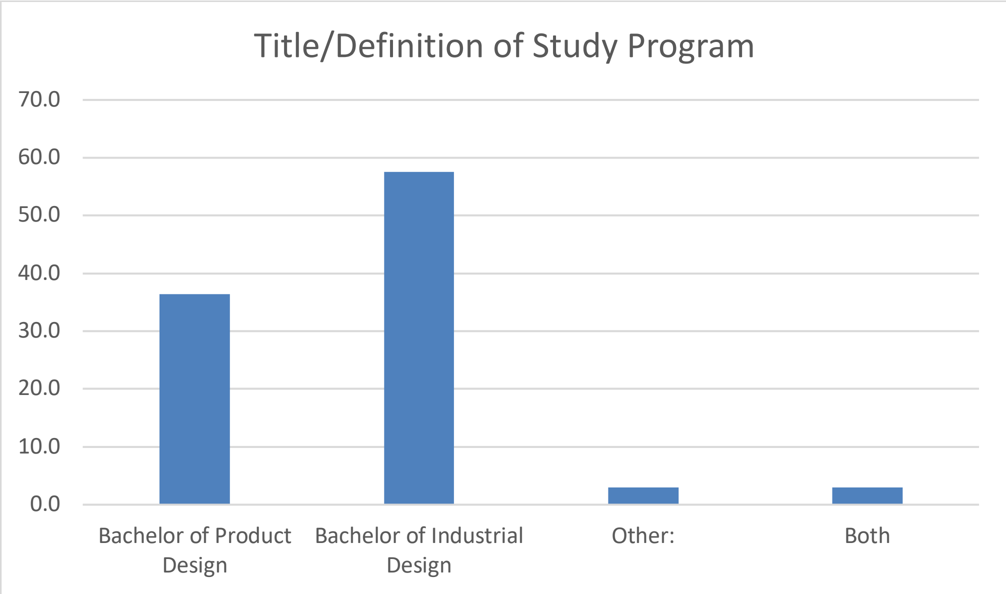

Figure A2.5 Title of study programs of surveyed institutions

Whether the program/course is hosted in a certain school or the other, seems to have no relation to the title. In North America and Australasia there is a marked tendency for programs to be 4 years in length, while in Europe the difference in the number of programs being 3, 4 and 5 years in length is smaller, with a slight

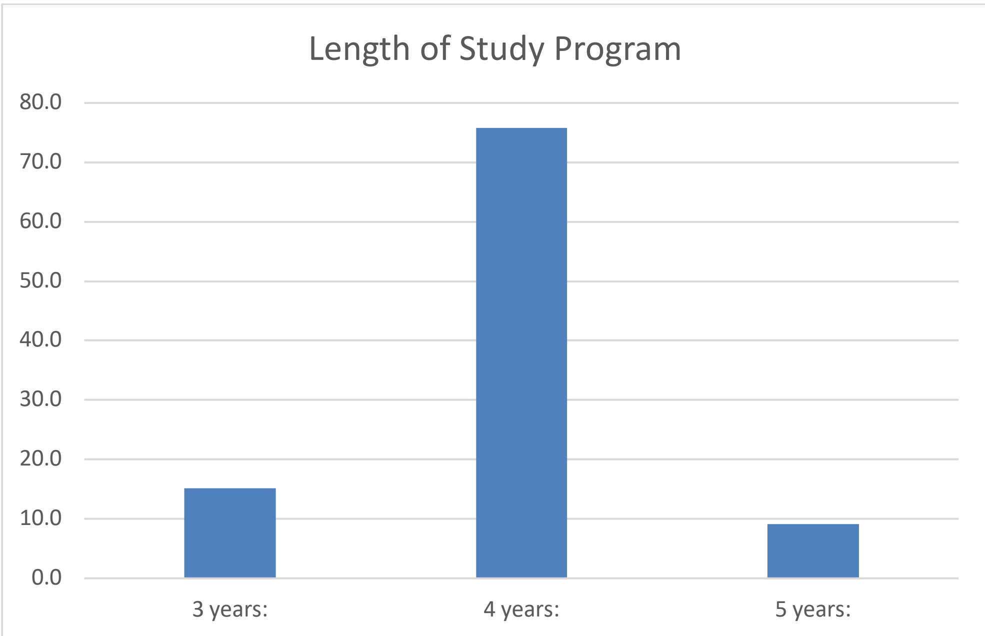

Figure A2.6 Length of studies of industrial design programs at participating institutions

It seems that programs labelled 'Industrial Design' tend to be longer than programs 'Product Design.' but this is likely to the fact that the label 'Industrial Design' as we have seen is more popular in America and as we have seen programs in North America tend to be longer than in Europe.

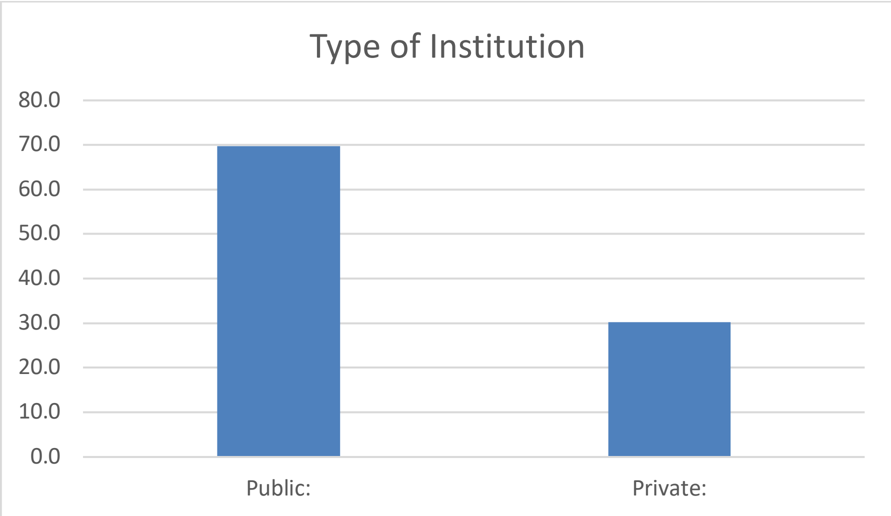

Figure A2.7 Percentage of participating institutions that are private or public

Most programs are hosted in a 'school of design' followed by a 'school of engineering' and then a 'school or architecture.' Still very few programs are independent (their own department or independent of other department)

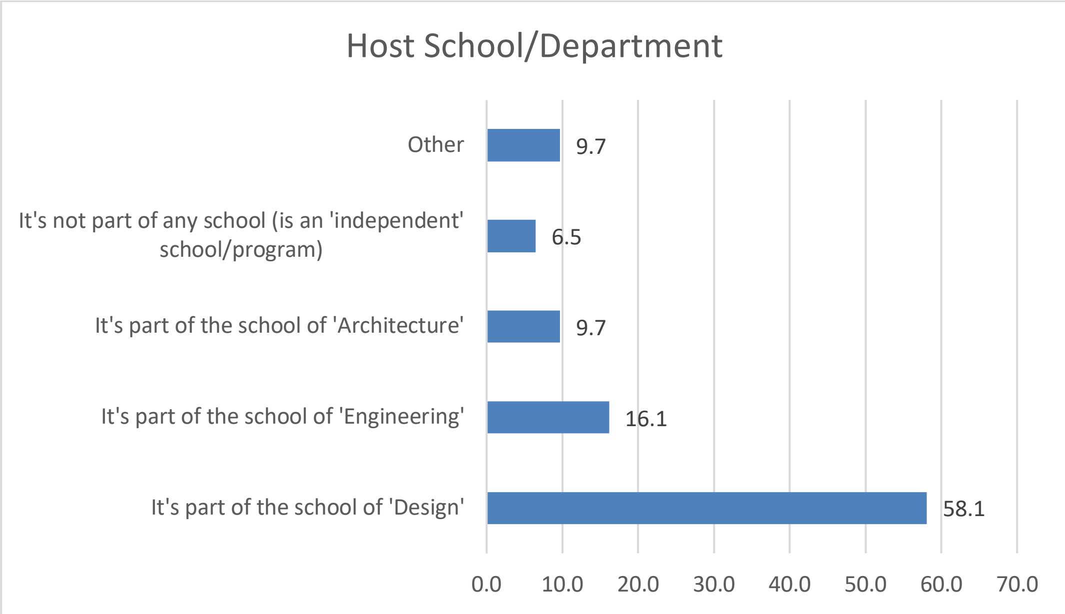

Figure A2.8 Hosting schools/departments of industrial design programs of participating institutions

The institutions were distributed regionally as shown in figure A2.9. There was no intention of focusing on certain regions, however this is where the responses tended to come from, most likely due the fact that the survey was in English.

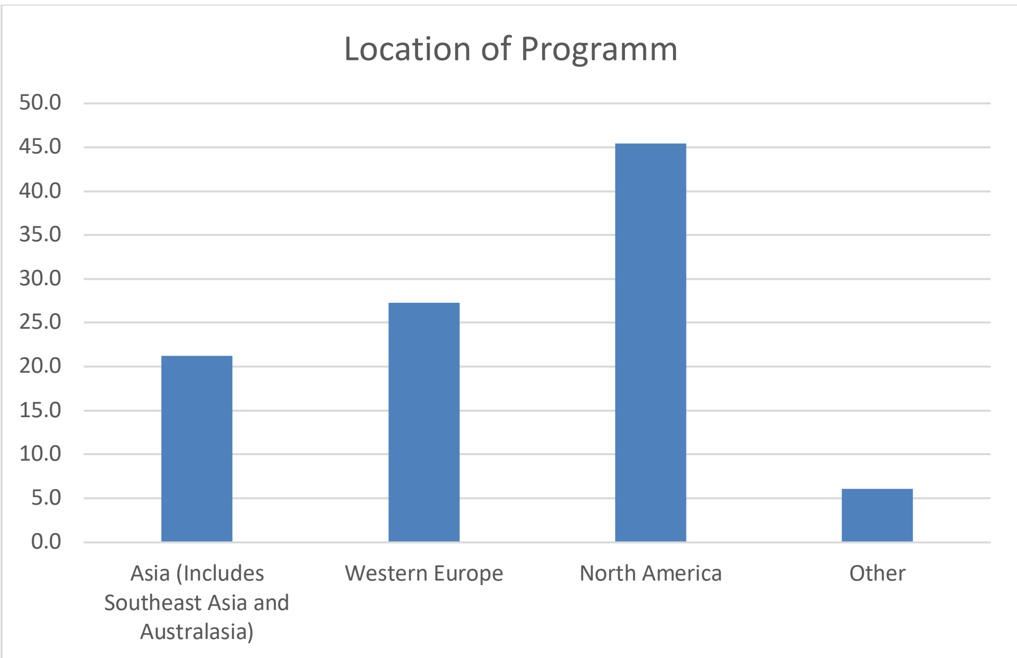

Figure A2.9 Location of participating institutions

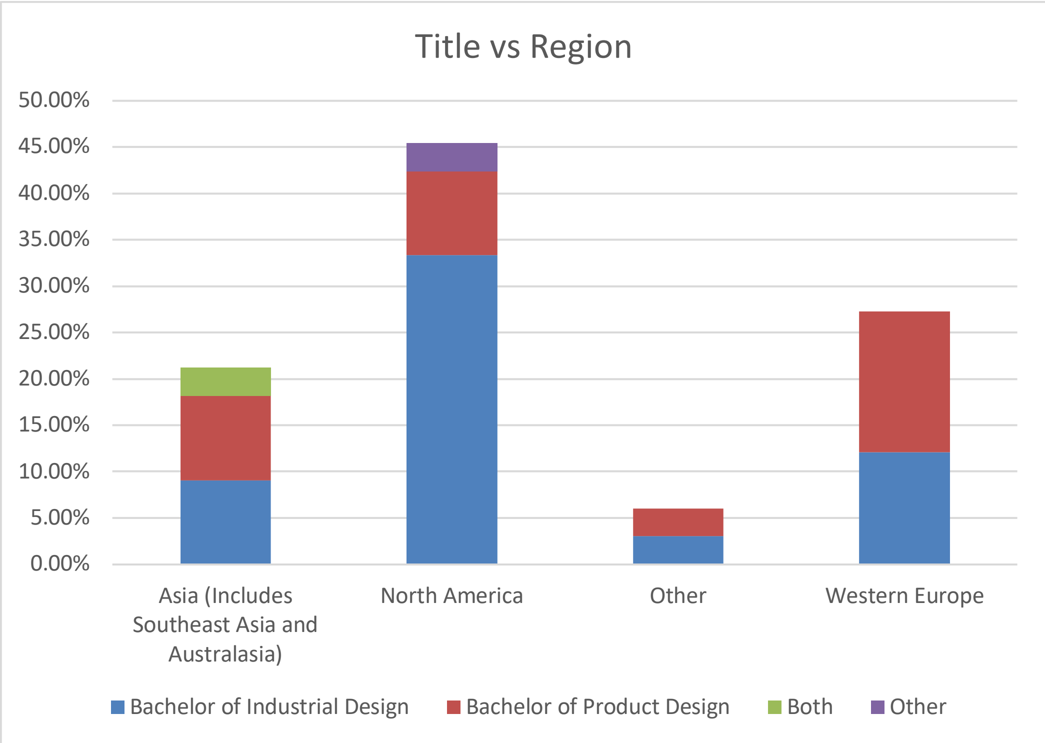

Figure A2.10 Title of study program vs location of participating institution

There is a marked tendency for programs in North America to be title Industrial Design. Whether the program/course is hosted in a certain school or the other, seems to have no relation to the title.

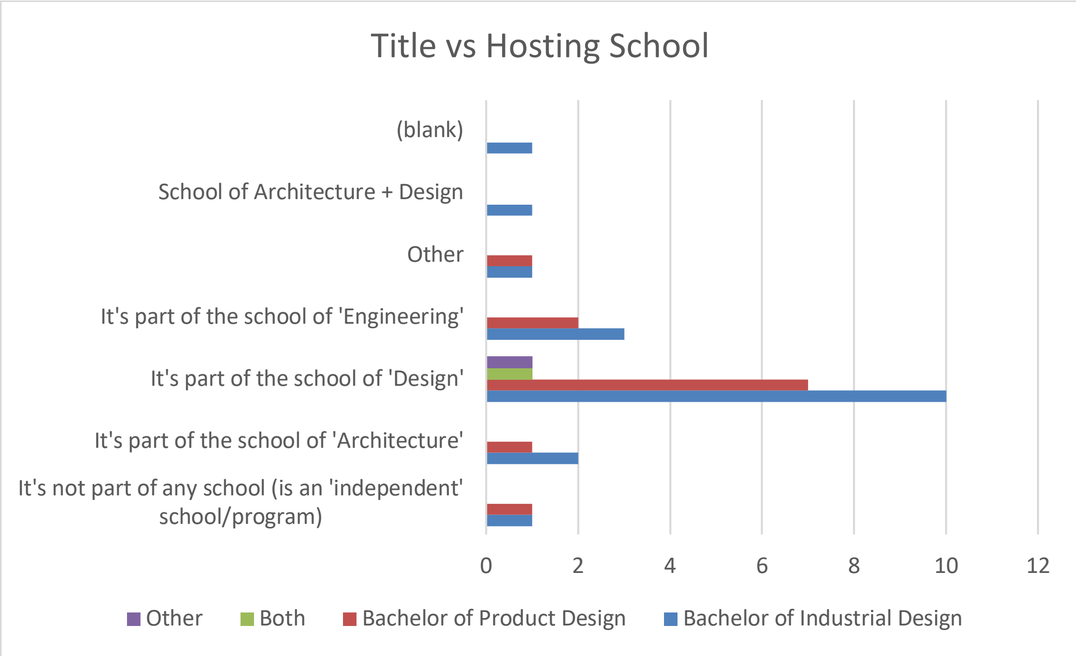

Figure A2.11 Title of study program vs hosting school/department

These results suggest that there is an 'embrace' of industrial/product programs at schools of Engineering and a lack of growth in schools of Architecture, despite that the discipline branched-out from Architecture.

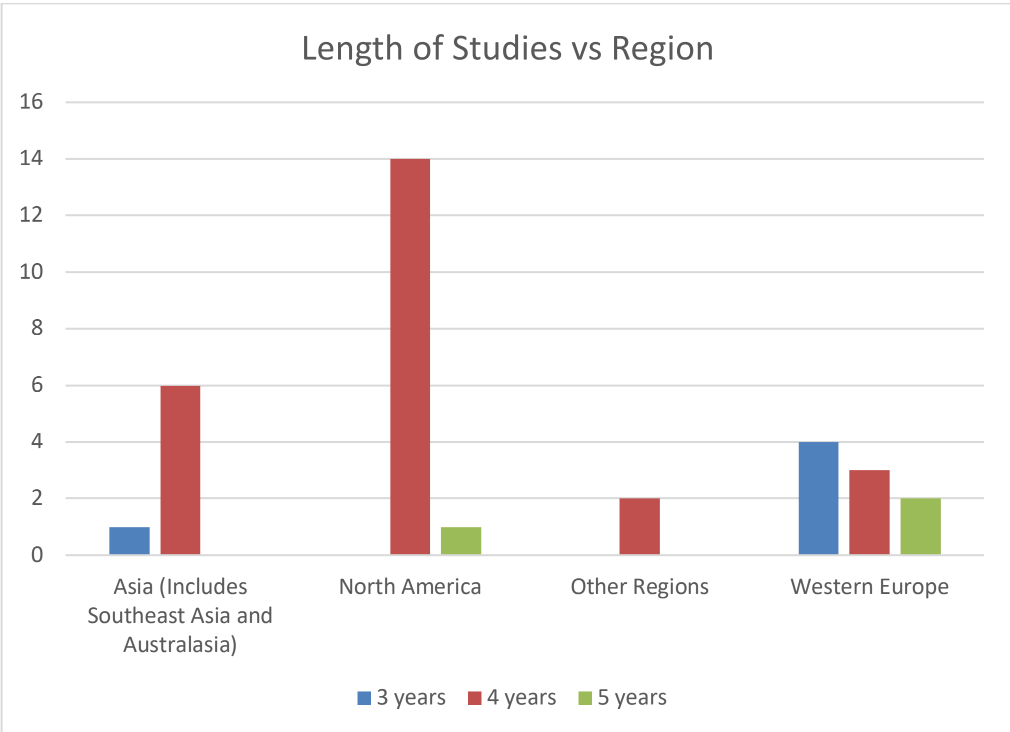

Figure A2.12 Length of study program vs location of institution

Most study programs are 4 years long. In North America and Australasia there is a marked tendency for programs to be 4 years in length, while in Europe the difference in the number of programs being 3, 4 and 5 years in length is smaller, with a slight majority being 3 years. Overall programs in Europe are shorter than in America. This can be explained by the fact that the title Industrial Design is more popular in America, and programs in America are longer.

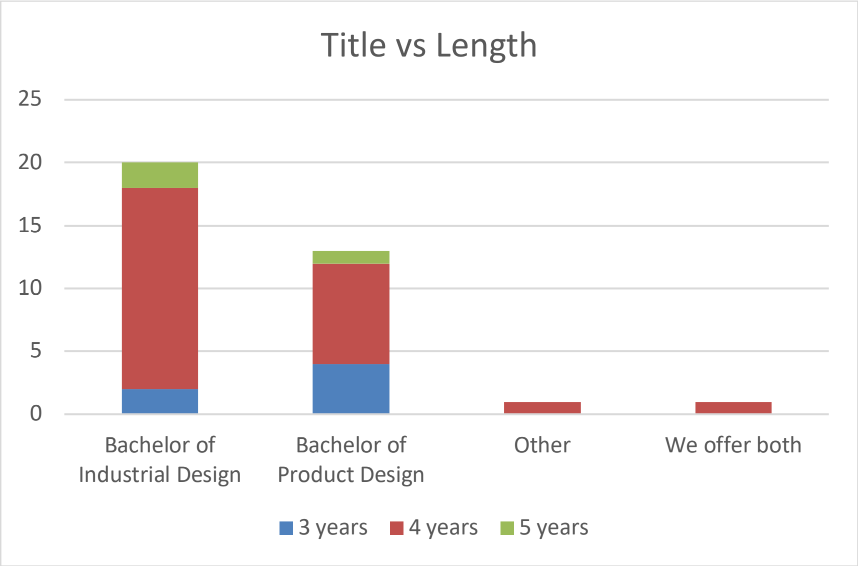

Figure A2.13 Title of the study program vs their length

Other regions outside Western Europe, North America, East Asia and Southeast Asia have been consolidated under 'Other regions.' All responses under East Asia and Southeast Asia were consolidated under 'Asia'

Overall programs in Europe are shorter than in America. This suggests that there is no relation between the title and length of the program and the how CAD/computer education is being approached. It seems that programs labelled 'Industrial Design' tend to be longer than programs 'Product Design.'

APPENDIX III (list of experts consulted)

Bernat Cuni: Product designer specialized in digital fabrication with experience in the fields of design research, eco-design and design entrepreneurship. Bernat Cuni collaborates with the Instituto Europeo di Design (IED) giving a course on Design Entrepreneurship at the Master in Design Management in Barcelona.

Boris Brawer: Boris is a product designer and lecturer. He has his own design studio: Made where he works as creative director. Prior to founding Made, Boris worked with international design consultancies in Germany, Switzerland, China and Taiwan. His passion for consumer electronics led him to Shanghai where he worked at ZTE Corp. one of the largest smartphone manufacturers in the world. During this time he was also a part-time Product Design lecturer. In 2010 he relocated to Taipei where he also works as lecturer of industrial design at Shih Chien University.

Clement Noury: Professor Clement Noury is the responsible of the product design department at Creapole Ecole de Design in Paris.

Daniel Gonzales Abalde: Daniel is a designer, graduate in artistic jewellery and expert in generative design using Grasshopper. He is the developer of Peacock and PhyloMachine, two plugins for Grasshopper and Rhinoceros. He collaborates with other designers and gives online courses of Grasshopper.

Dominique Sciamma: Dominique is the director of Strate School of Design in Paris. A well-known schools due its product-design program. Dominique created its Multimedia Department in 1998. In 2007, he created the "Interactive Systems and Objects" department. In 2010, he created Strate Research, the Research Department. Holding a DEA and a Maîtrise in theoretical Computer Science and a Maîtrise in Mathematics, and with a thorough career within international companies, Dominique has led, until his integration at Strate, a double activity, as a pluri-media author, and a consultant.
As an expert in AI, he spent two years in Singapore, between 1989 and 1991 where he successively worked as researcher in AI, product manager, and

Marketing director in the same department, and he eventually created the South-Eastern Asia department. His specialties are: Research, development, Interaction design, Editorial design, Interface design, Multimedia authoring, Communication and Creativity

Ernesto Spicciolato: Ernesto Spicciolato is senior teaching fellow at the Hong Kong Polytechnic University School of Design. Born in Italy, graduated at the University of Architecture, Florence, in 1983. With Santina Bonini he founded design-bs.com office in Milano in 1990. The office based its work on active research in the fields of the new technologies, evolution of artificial materials, utilization of natural material for industrial products, social evolutions. His works are in the permanent collection of Cooper Hewitt Museum (New York), Museo del Design della Triennale (Milano), MOMA Museum of Modern Art (New York), and Centre Georges Pompidou (Paris). Among the others, he designed for Bic Italia, Colombo Design, Covo, Dae-Young Computer, Gedy, Glas, LG Chemicals, LG Electronics, Viceversa, Zanussi-Rex.

Frederik De Bleser: Frederik is co-founder of Experimental Media Research Group (EMRG) at St. Lucas School of Arts in Antwerp, Belgium. He is the author of NodeBox, a well-known generative design tool developed by Frederik De Bleser and Tom De Smedt. He owns a private design company called Burocrazy. His PhD research studies the link between computers and art: the impact of procedural graphics applications on graphic designers.

Ian Campbell: After graduating from Brunel University in 1985 with a BSc in the Special Engineering Programme, Dr Campbell worked as a design engineer, first in Ford Motor Company, and later in the Rover Group. In 1989, he was appointed as a Senior Teaching Fellow for CAD/CAM at the University of Warwick. This gave him the opportunity to raise his awareness of CAD/CAM technology and practices. In 1993, he obtained a lectureship at the University of Nottingham where he was a member of the Rapid Prototyping Research Group. His principle area of research was the integration of rapid prototyping into the design process and he gained his PhD in 1998. He moved to Loughborough University in October 2000. His publications include four edited books, thirty academic journal articles and over seventy other research outputs. Funding sources for his research have included EPSRC, the Royal Society, EC Framework V, the Commonwealth Scholarship Commission and the National Research Foundation of South Africa.

Ilpo Koskinen: Professor Ilpo Koskinen joined the ranks of the School of Design in August 2014. He has been interested in industrial design and information technology, and lately service and community design, always from a research point of view. He has published well over 100 papers, conference papers and books. Koskinen is an experienced teacher, project leader, and thesis supervisor especially at MA/MSc and doctoral levels. His former students work in several countries in Europe, North America and Asia. Ilpo Koskinen is a sociologist by training, but has worked as a professor of industrial design at Helsinki's University of Art and Design (now Aalto University) since 1999. His main research interests have been in mobile multimedia, the relationship of design and cities, and interpretive design methodology. He is the author of Design Research through Practice: From Lab, Field, and Showroom, a book on constructive design research (Morgan Kaufman, San Francisco, September 2011). This book explicates recent developments in contemporary design research by focusing on their methodological foundations, whether they come from the sciences, the social sciences, or art and design.

JiaYi Young: Jiayi Young is an Assistant Professor of Design at the University of California, Davis. Her inquiries lie within the emergent and experimental field of digital media with an emphasis on the cross-disciplinary areas of design that integrates the arts, the sciences with cutting edge technology. Her current research and creative work are focused on constructing data-driven sensor-enabled interfaces, installations, real-time projection graphics, participatory performances, and immersive environments in cultural and public places with a goal of creating generative energy to engage the public in social dialogue. Jiayi Young has published and exhibited nationally and internationally, including Ars Electronica, the International Symposium of Electronic Art (ISEA); the Leonardo Electronic Almanac (LEA); Hall of Science, New York; the United Nation's Fourth Conference on Women, Beijing, China; the Hermitage Museum, St. Petersburg, Russia; and Moltkerei Werkstatt, Cologne, Germany.

John Cartan: Designer, inventor, and writer currently working as a Senior Design Architect in the Oracle User Experience Emerging Technologies group. John is concerned with making complex information easier to understand and enterprise applications easier to use. He started as programmer but now work exclusively as a user experience design architect and inventor. He has managed UX teams and enjoys being a mentor and team leader, but prefers to work as an individual contributor. He is a designer at heart. As design architect for the emerging interactions team, he enjoys spotting trends early and finding ways to apply new technologies to real-world use cases. His specializations are: Visualizations, Interactives, NUI, Tablets, Information Architecture, Interaction Design, Usability Testing, Graphic Design, User Experience, and Analytics.

John Frazer: John was born in Lancashire, England. Educated at Stowe School and then the Architectural Association and Cambridge University. He trained as an architect and then taught and researched in both architecture and art and design schools. He started lecturing at Cambridge Department of Architecture where he co-founded the Technical Research Division (now part of the Martin Centre) He subsequently went back to the AA again to teach and then to Ulster where he was the Head of the School of Art and Design Research. He founded the Ulster Centre for Computer Aided Design and was awarded a Personal Chair at the University of Ulster in 1984. He also founded the award winning Autographics software development company. He moved to Hong Kong again as Swire Chair Professor and Head of the School of Design embracing Environmental Design, Fashion, Graphics and introducing Interactive Design and a Global Virtual Design Studio and the Design Technology Research Centre. Then he worked for Gehry Technologies as International Research co-ordinator of the digital practice ecosystem. From there he went to be Professor of Design Science and Head of School of Design in Queensland Australia where he founded a Centre for Complex Urban Systems Design. He also holds Honorary Professorships from the Universities of Dalian, Fudan and Shandong and the Beijing Institute of Clothing Technology, is Visiting Professor at the Universities of Salford and Brighton, and Fellow of the Royal Society of Arts. John is author of over 200 book chapters and papers. Public recognition includes having been a Board Member for the Design Council of Hong Kong, and Art Advisor to the Art and

Heritage Museum Hong Kong. More recently served on the Action Agenda for the built environment and other working parties for the Australian Government. He is Chair of the John and Julia Frazer Foundation for Accelerating Architecture.

Michael Elwell: Mike Elwell is a product of the University of Notre Dame Industrial Design undergraduate program. After graduation, Michael was employed at Coachmen Recreational Vehicles, Radio Flyer, and the design firm Process4. He worked on a wide range of products from automotive to toy design before attending graduate school at the University of Illinois at Urbana-Champaign. His M.F.A. thesis, a safer infant crib, creatively utilized materials to reduce the chances of Sudden Unexpected Infant Death from accidental suffocation. He was honoured with the 2011 Teaching Excellence Award at the University of Illinois before returning home to Notre Dame. His current research focuses on social entrepreneurship. Along with being the 2005 IDSA Merit Award Winner, Michael also won third place in the International Housewares Association Student Design Competition. His patented product, a prescription pill container opener and label magnifier, is now licensed as the "Magnifying Medi-Grip." The product also was an Honouree in the 2009 Design Defined awards and won Best in Category at the 2010 Housewares Design Awards.

Mick Geerits: Mick is co-owner of 'abnormal' (www.abnormal.design) a studio specialized in computational design and art based in London. Mick has a background in Industrial Design and is an expert in computational/generative methods. He studied at the Royal College of Art and his passion for technology led him to diving into code and electronics. His interests revolve around emergent behaviour, decentralized systems, electronics and growth algorithms.

Oluwaseyi Sosanya: Oluwaseyi Sosanya, is the co-founder of Gravity Sketch and the inventor of the 3D weaver loom. He studied mechanical engineering at Oregon State University. He graduated from a double master's course by the Royal College of Art and Imperial College, London, titled "Innovation Design Engineering" in 2014. After graduation from college, he worked as a

CAD draftsman building homes in Oregon. He moved to Taiwan to learn Chinese. He returned to Oregon to co-found Grove as a design engineer. He then moved back to Taiwan to work as material specialist at Pegatron, a well-known-design consultancy. After graduation from RCA, he was recruited as innovation design leader at Jaguar Land Rover where he worked for two years. Since then he is focusing on turning his graduation project 'Gravity Sketch,' into a company.

Paul Goetz: Paul is currently Senior Vendor Manager at Amazon Automotive. He directed the design and development team at Gnosis product development in Shenzhen China. He worked closely with the consultancy's partner factory Ngai Kwong International, a roughly 2000 person contract manufacturer in Shenzhen. He focused on the manufacture of electromechanical products from vacuums to hair styling tools.

Richard Malachowski: Richard is an industrial designer born and raised in Lima Perú, from Polish and Italian family heritage, he moved to Barcelona and do a Master in Transportation Design through which he collaborated with companies like Smart Design, Seat, and Cannondale. In 2011 he moved to Shenzhen, China where he worked at Gnosis product development first as product designer and later as design leader. He now works in Barcelona, leading his own design studio 'Play Industrial Design.'

Sivam Krish: Dr. Sivam Krish is an education innovator and a globally recognized expert in generative design. He works at the University of Canberra. He is developing PhoneLabs, an open source, student led, hands-on, educational initiative, being adopted by schools and universities worldwide. He has global teaching and research experiences across the diverse disciplines of design, engineering, technology. In addition to his current role in leading PhoneLabs, he consults on creating Maker Spaces and in developing student led learning schemes. He is now busy creating technologies for authoring the next generation of online learning experiences.

Thomas Hvid Spansberg: Thomas is leader of the course at the School of Communication and Culture of Aarhus University. The program seeks to prepare students for understanding digital culture. Thomas is an expert on computation education and teaching computer programming.

APPENDIX IV (Pilot Study)

Design students, educators and practitioners were asked a series of questions geared towards finding out what is the use they make of computers and the role that they see them having in the design process. The aim was in part to get a first-hand impression of what people in the field understand by Computer Aided Design.

LinkedIn was found to be particularly suitable for this purposes because it features specialized professional groups formed by professionals of very specific fields or professional interests such as industrial design or even industrial design education. The number of respondents was of 60 (N=60).

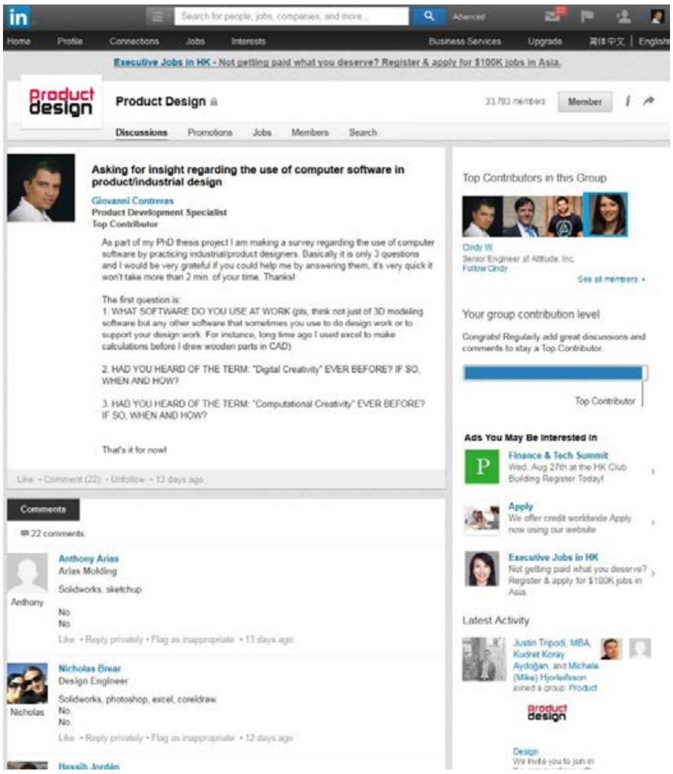

Figure 5.1 Sample Screenshot from the PD discussion board on website LinkedIn

The intention was to leave the question open to see whether participants reported all the software they used and not just 'design' software. In reality what we wanted to know was if they were conscious of all the software they use to do–aid–design or not.

As it was expected, practically every respondent reported using software like 'AutoCAD' and 'Photoshop,' that is, software for modelling and visualization purposes. Only around half reported using 'productivity' software. It is hard to believe for instance, that only half of the respondents use any productivity software, yet they do not consider it–it seems–part of the toolset with which they aid their design process. Although participants were not asked whether they use 'productivity software' as such, they were explicitly asked to list all the software that they use on a typical design project. Besides, they were asked to think of not just typical design software.

Less surprising however, was the fact that not one of the respondents replied using other type of programs usually associated with facilitating creativity or innovation such as mind-mapping tools or even team-collaboration software. As expected, almost all of the respondents–97% of the total, and mostly product designers–reported using some kind of modelling software. See Figure 5.2.

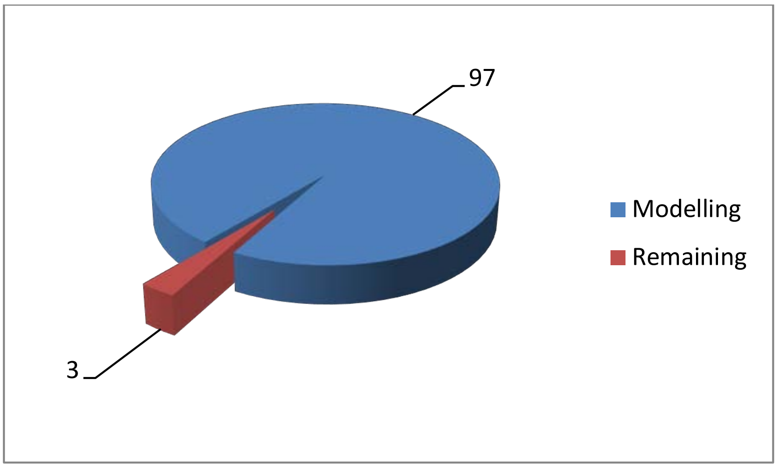

Figure A4.1 Percentage of participants who use computers for modelling or other purposes in a typical design project

56% from the total reported to use other type of tools besides using modelling tools, however most of these additional tools were productivity type of tools–Microsoft Office for example. This would make the use of productivity tools the second most common use of computer tools for product designers. See figure 5.3.

In regards to using 'typical' design software such as using 3D modelling packages for 'alternative' or different purposes, only a few replied engaging in alternative/exploratory practices when using those tools. Perhaps this goes back to the issue of having fixated perceptions about what the computer is for a long time and not challenging its role. It is hard to believe for instance that only half of the respondents use any software that would fall into the 'productivity' category, yet they do not consider it—it seems—part of the toolset with which they aid their design process. We did not ask whether they use 'productivity software' as such, however we wanted precisely to see whether they would mention software that would fall into this category when thinking about all the software they use in a design project.

Participants were also asked them if they had any idea what the terms '*computational creativity*' and '*digital creativity*' meant. In terms of the level of awareness about these terms, practically no one replied knowing what it meant. A similar response was obtained when they were asked if they knew what the term '*VPL*' was and what was the connection with computational creativity—VPL is an acronym for '*visual programming language*.'

Basically not a single respondent—up to the time this report was revised—reported using any type of computer tool for creativity-assisting purposes, the closest response in this regard was from a participant who reported to use mind-mapping software. See figure 5.4.

When it came to the question "have you heard of the term *digital creativity*" before? Only 22% of the total of respondents reported knowing the term. See figure 5.5. This suggests that indeed participants are mostly unaware of other possible uses they could make of the computer tool.

The use of digital design technologies remains highly concentrated on modelling and visualization tasks, while the other stages of the design process remain mostly 'virgin' to the use of digital tools. Surprisingly for example, no-one replied using a project management software, which, having worked in industry for a number of years, I can attest, it would be a very useful skill to have. Now, it is not that the tools are not there, because there are tools, but these are not being used. Which, suggests, it is a matter of perception. This is preventing us from assuming a more proactive role in looking at where are the limits of our technology.

Regarding the term "have you heard of the term *computational creativity*" before? The knowledge of the term was even lower, only 22% of the total of respondents reported having heard of the term, further confirming that the level of awareness about the possibilities of using the computer as a partner in the creative process is very low amongst product designers. See figure 5.6.

None of the respondents replied using other type of programs such as mind-mapping tools or even team-collaboration software. As part of this pilot-study, participants were also asked what they understood by '*computational creativity*' and '*digital creativity*.' The intention was to know their level of awareness about these terms. Practically no one replied knowing what they meant. A similar response was obtained when they were asked if they knew what the acronym '*VPL*'–visual programming language–was and what was its connection with computational creativity.

From the responses, it is possible to deduct whether the use and/or teaching of computer skills in industrial design remains primarily focused on modelling or not. Design Educators on the other hand, were asked to list all the software tools that students learn at their respective institutions. Once again they were specifically asked to think of all software, not just 'design' tools. Educators were also asked to mention if they made use of tools for any 'uncommon' purposes–i.e., if they used Solidworks for something not related to 3D modelling. Equally, only a few replied engaging in alternative/exploratory practices when using 'typical' design software such 3D modelling packages for 'alternative' or different purposes.

Glossary

21st Century Literacies: Sometimes also referred to as 'New Literacies,' is the range of abilities and competencies that a person is believed will need to succeed in the 21st century. There is no definitive articulation of what these literacies are, however the National Council of Teachers of English for example, include the following: developing fluency with the tools of technology, building cross-cultural relationships with others to solve problems collaboratively while strengthening independent thought, designing for, and sharing information with, global communities for a variety of purposes, managing, analysing, and synthetizing multiple streams of information, creating, criticizing, analysing, and evaluating multimedia texts, attending the ethical responsibilities of the environment.

3D Printing: A rather colloquial way to refer to additive manufacturing methods. Most of the times it is used to refer to additive manufacturing based on extrusion or fused deposition modelling (see additive manufacturing). Much fuzz has been generated around 3D printing in past years, with some people arguing it could deeply transform production methods, and having profound social and economical implications. Many of these claims are often exaggerated by uninformed enthusiast. It is true however, that because 3D printers build products layer by layer, they can produce physical objects without the constraints of traditional manufacturing methods. For more information about 3D printing, see Campbell et al. 2011.

Additive manufacturing: Also referred to as 3D printing, is the fabrication of a physical part of object using a 'layer by layer' approach. Additive manufacturing has gotten much attention in recent years, with mixed reactions from business leaders. Some of the different additive manufacturing methods are: material extrusion or_'fused deposition modelling,' (the most common, used in popular 3D printing machines such as the 'makerbots'), powder bed fusion (at one point made popular by Z-Corps machines, and probably due which the term '3D printing' became popular), binder jetting, material jetting, vat photo-polymerization, directed energy deposition, and sheet lamination. Each of these has its own advantages and disadvantages, including materials available, build volume, construction speed, quality, and amount of post-processing needed. Nowadays 3D printers range in size from desktop printers to printers capable of building much larger parts even houses. Common plastic materials used in the case of extrusion machines include: polycarbonate (PC), acrylonitrile butadiene styrene (ABS), polyphenylsulfone (PPSF), and other PC-ABS and PC-ISO blends. Additive manufacturing processes usually require computer files in a stereolithography (STL) format, a de-facto standard for 3D printing machines. In this process, the CAD model is approximated by triangles and sliced into thin layers.

API: An Application Programming Interface (API), is a set of computer code provided by the vendor of a software (the application) and which can be used to manipulate that same application or a process within it automatically. APIs often enable the communication between two more computer programs. APIs have become particularly important online, because they allow different websites to share information with each other. For example; an API allows a website like Instagram to fetch information from a user's Facebook account automatically. In this case, an API serves as interface between the two services. Not to confused with 'Academic Performance Index,' another term also used in the context of education.

Architectural Design: Refers to the design of habitable spaces, such as houses or any type of building destined to be inhabited. It can also include the design of urban landscapes, although the latest also overlaps with the fields of Urban Design and/or Urban planning. Architectural designers hold similar skills as Architects, but in some countries Architectural Design is seen as a lower type of qualification.

Arduino: Open source computer hardware and software company. Well known for its single-board microcontrollers and kits for building digital devices that can sense and control other devices, thus often used in the design of interactive products. Distributed as open-source hardware and software, licensed under the GNU Lesser General Public License. In recent years, and due increased interest in interactive devices and interaction design, a number of product design schools have incorporated projects involving Arduino in their study plans. To learn more about Arduino, visit: ww.arduino.cc.

Associativity: Used in the context of Computer Aided Design, to indicate that one drawing or 3-D model can have a relationship to other entities or models. If dimensions between two entities are associative, then if a feature of one of these entities is changed, all other features associated to this feature being changed will also change. Associativity is a common characteristic of parametric models/modelling.

Bitmap Image: A bitmap image, is one where there is no pure mathematical definition of what is to be seen in the hardcopy. Instead, the image is created by directly 'dumping' individual bits of data into a file straight from graphics memory (thus the name). The computer file generated contains all the needed data to display the image, such as the colour of each pixel. A bitmap then, is just a file that has a map of what each bit looks like. Common examples of this type of files include the formats: TIFF, GIF, JPEG, and BMP.

C++: Computer programming language in development since the 1970s and widely used in the development of a variety of PC applications. C++ is also a popular language for programming microcontrollers. C++ is an enhanced version of 'C,' with the addition of object-oriented programming concepts. Applications written in C++ must be interfaced with a graphics library such as OpenGL in order to display images on a computer screen.

CAD Model: A CAD model is the set of information, usually mathematical expressions of geometric bodies, expressed in computer code, and which represent a geometric entity/body, whether two or three-dimensional. This information is contained in a computer file, and can be made manifest in different ways, such as when displaying an image of the model in a computer screen, or when being physically built such as when using a 3D printer. Since CAD models are mathematical abstractions of geometry, they are always 'correct,' because at their core there is a mathematical formula.

Courseware: Term given to materials which are part of an educational course or class. Although courseware refers to all types of materials, it is mostly associated with technology-based materials particularly software, and often times is found as synonym with educational software in general. The term is particularly common when discussing training for computer programs or IT industry's certification programs.

Constructivism: Constructivism is a philosophical understanding regarding the nature of knowing, and suggesting that we (humans) build knowledge from what we experience. For the same reason, it has a direct application to education and learning in general. This philosophical stance, mostly epistemological, was popularized by renowned psychologist Jean Piaget in his multiple works about child development.

Computer Aided Styling (CAS): A concept derived of Computer Aided Design (CAD). It is used to denote the application of computers to the process of styling of a product. It is a more specific term than CAD, concerned mostly with the design of the exterior surfaces (also known as Class-A surfaces) of a product.

Computer Aided Concept Design (CACD): Another concept derived of CAD. It refers specifically to the application of computers to the conceptual design of a product. It is used to denote specifically an area of CAD that has to do with the conceptual stage of the design process.

Computer Aided Aesthetic Design (CAAD): Similar to the previous' two, it is a term used to describe the area of CAD concerned with developing the aesthetics of a product, generally these are determined by the exterior surfaces and in this sense it overlaps substantially with Computer Aided Styling (CAS).

Computer Aided Technologies (CAx): Umbrella concept used to refer to the multiple computer systems involved in the New Product Development process such as CAD, CAM, CAE etc. Advanced tools may combine several of these aspects into one package.

Computer Aided Instruction (CAI): Used to refer to the process of supporting instruction of any kind with the use of computer technologies. Tools such as virtual learning environments (VLEs) are examples of computer aided instruction, but the term CAI is more encopasing.

Computer Art: Art in which a computer plays a role in the creation of the art piece. Unlike Generative Art, the term Computer Art, does not necessarily imply that the computer will be used to introduce a degree of randomness into the art piece, simply that the computer is somehow used to produce the art, whether in the process, or the art piece itself.

Concurrent Engineering: Approach to new product development 'in parallel' as opposed to a 'sequential' approach in which certain activities are carried one after the other. It has the advantage of reduced 'head times.' Multi-disciplinary teams work in parallel at the same time, developing different parts and features of each part of a product.

Computational Creativity: Area of knowledge in charge of studying the capacity of computers to create "by themselves," show/ emulate human creativity. It attempts to reproduce a "human kind of creativity" in computers through the use of artificial intelligence. The creations generated can be of any kind, including thoughts or ideas.

Computational Thinking: Term coined by Computer Scientist Jeannette Wing, and which refers to the ability to think by drawing from fundamental concepts of computation. It is a type of analytical thinking sharing some characteristics with mathematical thinking, engineering thinking, and scientific thinking. Wing and others argue that Computational Thinking will be a skill that every person will need by the middle of the 21st century. In support of this argument, it is said that Computational Thinking is already having a strong impact in all disciplines, the most widely cited example is the sequencing of the human genome. A fundamental aspect of computational thinking is abstraction, the ability to know how to translate life phenomena into information that can be computed. Computational Thinking has gained much attention in the educational arena in recent years. For example, a new basic computing education curriculum was unveiled in the United Kingdom in 2013, precisely with the aim of fostering computational thinking skills and moving away from simply teaching students ICTs.

Computer Literacy: Refers to the ability to use computers with fluency. It involves a wide range of skills, from elementary use, to programming and advanced problem solving. In its broadest sense, computer literacy requires some understanding of computer programming and how computer hardware works. Computer literacy is considered an essential literacy for the 21st century, both in order to compete in the workplace, and in terms of being able to access services and be fully empowered in society.

Creative Coding: Term used to refer to the user of computer programming for 'creative' purposes. While coding can be seen as a creative activity in nature, the term 'creative coding' refers more to using coding in projects carried out in creative fields like art or design disciplines.

Design Creativity: Term used in recent years to discuss creativity specifically within the context of design. Although its definition is also a work progress, it may represent a viable alternative to avoid the definitional issues of creativity widely discussed.

Design Research: Originally understood as research into the process of design, the concept has expanded to include research through that process. Design Research aims to understanding and enhance the design processes and practice in general terms, generating insights that can be applied to any particular discipline (i.e., Industrial Design or Graphic Design). It can be understood too as the research that designers do when working on a design project. It has been pointed out that in comparison to other fields, design research is very new and lacks its own methods as well as a clearly defined research agenda.

Digital Clay Modelling: Refers to a 3D modelling approach which resembles working with modelling clay, but in the computer. In this approach, the modelling process starts by creating a 'lump' of clay with has a basic form (like a semi sphere/egg) and then using tools to start sculpting it as if working with actual clay. Most of the times, this modelling approach needs to be accompanied of special 'sensing' pointing devices which provide tactile feedback in order to truly exploit its potential. The most popular of such type of devices is 3D system's 'Touch.' This modelling approach is based on polygonal models, and utilizes polygonal meshes with densities of several million polygons. The approach is particularly suitable when working with sculptural/organic bodies. The approach was pioneered by the entertainment and toys industries, particularly for the modelling of figure characters, however in recent years it is being employed in the design of products/vehicles. For more information about digital clay, see Alcaide-Marzal et al., 2013, and Zammit & Munoz, 2014.

Digital Design Technologies: Any technology used in design (creative design) and which is digital in nature (including but not limited to what is commonly known as 'Computer Aided Design' or CAD) (including but not limited to what is commonly thought as CAD).

Digital Hand-crafting: Refers to the process of building a model or artifact using digital technologies, mostly computers, as a medium. The concept of

Digtial Hand-crafting, is that the digital medium (computer) is still treated like any other medium (i.e., paper or clay), that is, the designer has a more or less clear idea of what he/she wants the model or artefact to look like, and crafts it using the medium (and its associated tools) according to the mental image she/he has.

Digital Imaging: Refers to the acquisition, creation and/or manipulation of images using digital technologies. Typical examples would include the acquisition of an image using a digital scanner or a digital camera, the manipulation of those images using an image-editing software such as Adobe Photoshop, and/or printing those images using a digital printer. In any instance of digital imaging, an image is the result of the process.

Digital Creativity: Refers to the idea of using digital mediums in creative projects. It applies to all fields where it is possible to create using digital medium, however since all design fields such as graphic or product design nowadays use digital medium it is commonly associated to those fields. Not to confuse with computational creativity, in which computers create by themselves or aid the designer to create using a certain degree of autonomy or 'intelligence.'

Digital Literacy: Is the knowledge, skills, and behaviours used in a broad range of digital devices such as smartphones, tablets, laptops and desktop PCs. Digital literacy initially focused on digital skills and stand-alone computers. A digitally literate person will possess a range of digital skills, knowledge of the basic principles of computing devices, skills in using computer networks, an ability to engage in online communities and social networks while adhering to behavioural protocols, be able to find, capture and evaluate information, an understanding of the societal issues raised by digital technologies (such as big data), and possess critical thinking skills.

Digital Morphogenesis: Term used predominantly in architecture, refers to the process by which a form is transformed using digital technologies–mostly computers–and which often resembles methods found in nature. The resulting forms are typically suitable for structures. It could be said that Generative Design is a form of Digital morphogenesis.

Digital Sketching/Digital Hand-Sketching: Refers to the process of doing hand sketches using a drawing tablet (whether a tablet-computer or a drawing tablet connected to a computer) instead of sketching directly on paper. Wacom ® has been a pioneer in the development of this technology and still dominates most of the market of drawing tablets.

Distance Learning: Refers to learning in which the tutor and the learner do not share the same space. It is in general agreed that the first forms of distance learning were correspondence courses where the printed materials were sent to learners by mail. As technology evolved, these materials also evolved, giving place to audio cassettes, VHS tapes, Audio CDs, and DVDs. With the advent of the internet, and advancements in streaming audio and video, sending materials by mail has become less popular, and now most exchanges occur completely online. Most of the times e-Learning can be considered a form of distance learning, and often times the terms are used as synonyms, but are not the same. Also, while in most distance learning programs the teacher and the learner do not share the same time, strictly speaking, distance learning only implies a separation of space, not necessarily of time (i.e., online distance learning in which the teacher and learners are in contact with each other at the same time through video chat).

E-Learning: Term used to designate the type of learning that occurs through electronic devices although it is commonly used to refer to distance educational programs that take place over the internet. It is also often found as synonym with several other terms in the educational technology jargon such as distance learning; however the second does not necessarily imply the mediation of electronic devices or the internet.

Engineering design: Engineering design is a systematic, process to generate, evaluate, and specify devices, systems, or processes whose form and function meet a specified function/objective, while at the same time, satisfying a set of design criteria or constraints. The process involves a methodical series of steps. This is also a highly iterative process in which certain tasks or stages often need to be repeated several times. In this process, basic sciences are applied to transform resources in an optimal way. Fundamental elements of the design process include: establishing objectives and criteria, synthesis, analysis, construction, testing and evaluation. Design engineers exist in all engineering disciplines, including: civil, mechanical, electrical, chemical, aerospace, nuclear, manufacturing, computer, and structural engineering. Design engineers work with other engineers and designers, to develop product concepts and specifications that ensure that a product actually works and fits its purpose. Design engineers often work with industrial designers and marketers to ensure proper design direction and that the customer needs are met.

Feature Based Modelling: Approach to creating computer models used in industrial/product design by building features. Typically the sequence of features built is recorded on what is commonly referred to as “feature tree” of “history tree.” Features can be updated at any point and the model is automatically updated.

Finite-element Analysis (FEA): Refers to a numerical analysis process widely used in engineering disciplines. FEA is used to study the performance of an object by dividing it into small building blocks (finite elements) that form a model of the actual object. FEA is widely used in Mechanical Engineering to predict such things as a structure's stresses and deflections.

Formal Education: Refers to all education provided by the state, typically divided into basic, secondary and tertiary (higher education). In most countries, formal education is compulsory up to the secondary level. It is often offered by public and private parties. Formal Education represents the three stages that a person would normally have to go through in order to earn a professional qualification or degree in most countries.

Generative Art: Term used to describe art in which the artwork, or part of it, is created through a process not fully under the control of the artist. The process for creating the art can be based con computer algorithms that generate an output (often times random), but can also be completely computer-free, such as when using mechanic or organic elements to transform an art piece.

Generative Design: Term used to refer to the process of using computer algorithms to create or transform existing forms (whether in 2D or 3D). The process often yields unpredictable results. The resulting output sometimes resembles forms found in nature. It could be said that generative design is a form of computational creativity. Arguably, Generative Design does not have to be computer-based, yet when the term is found, most of the times it refers to a computer-based process.

Genetic Algorithms: Term used to denote computer code inspired by Darwin's theory of evolution. Often used to create variations of an existing universe of elements or 'population' by means of making combinations among the different members of a set, as if combining chromosomes, hence the name. Genetic algorithms are sometimes used to generate design variations. Genetic algorithms include an examination step in which the "fitness" of the variations created is tested and only the best 'individuals' are kept for further reproduction.

Generic CAD System(s): Within the context of the design and engineering disciplines, it refers to Computer Aided Design program packages that can be applied in a wide variety of industries and for different purposes. Examples of such systems include Autodesk's AutoCAD, Trimble's Sketchup (formerly Google Sketchup), TurboCAD, FreeCAD and FormZ among many others.

Graphical User Interface: Or 'GUI,' term used to refer to the set of graphic elements through which a person interacts with a machine or computer program. In the early days of computers, programs did not have a GUI, consequently, users operated them by typing text commands using a keyboard and a command prompt.

Grasshopper: Is a graphical algorithm editor/visual programming language developed by David Rutten. It comes as a plug-in for the popular CAD software Rhinoceros 3D. Models are created by dragging components onto a canvas. The outputs to these components are then connected to the inputs of subsequent components. Grasshopper is probably the most popular tool among Architects and Product Designers interested in Generative Design.

ILS: Acronym for 'Integrated Learning Systems,' essentially is a set of tools–most often computer-based and on line–used to organize educational materials and curriculum content for a class, so that students have access to them as the progress through the course. ILSs may include information or curriculum content, evaluation mechanisms, communication or discussion boards, and other tools. The term is used in tandem with LMS or 'Learning Management Systems,' although the second seems to be gaining popularity. These days there are powerful free and commercial ILS systems; an example of a well-known free ILS system is Moodle. Its commercial counterpart–and probably best known–is Blackboard.

Industrial Design: Discipline mostly concerned with giving form to products in order to provide the best possible value to consumers. In order to accomplish this, industrial designers determine not just the appearance of a product, but also its functions and mode of operation. It could be said that Architects are to the design of the house, what Industrial Designers are to the design of the products inside. Industrial Design emerged as a professional discipline in the 19th century. Ever since, businesses have grown aware of the importance of Industrial Design to gain a competitive edge. Pioneers in the field include people like: Charles and Ray Eames, Henry Dreyfuss, Alvar Aalto and Dieter Rams. Well known contemporary industrial designers include Phillipe Stark, Karim Rashid, Jonathan Ives, Yves Béhar, and Patricia Moore. For more information about Industrial Design, see: The Industrial Designers Society of America (www.idsa.org), the China Industrial Design Association (chinadesign.cn), and/or the British Industrial Design Association (www.britishindustrialdesign.org.uk).

Informal Education: Is any form of education guided by an informal curriculum or study plan, and which may or may not involve a tutor. Typically, all education that is not considered formal, afterschool or personal tutoring would fall into this category.

Informal Learning: Refers to all learning that is not acquired through any form of formal education or schooling. It can be intentional or not; such as when something is learnt without expecting it, or self-driven; when something is learnt purposefully but without the help of any tutor or formal study guide.

JavaScript: Computer programming language originally called LiveScript, and developed by extinct web browser Netscape. Initially developed to enhance Web pages. JavaScript is the default scripting language for Adobe's software; Illustrator, Photoshop, etc.

Knowledge Economy: Refers to the notion that knowledge can be used to generate tangible and intangible products with commercial value. It is in general well accepted that with the rise of 'the information society,' the world economy is in a state of transition to become a 'knowledge economy,' in which case, the importance to education is greater than ever before.

Learnscaping: Term coined by informal learning specialist Jay Cross, it refers to the process of transforming an environment in order to make it more conducive to learning. This transformation can be physical or cultural, although normally they should come together.

Macros: Computer programs that capture a user's actions within a computer system, and which can be reproduced later when necessary. The process starts with the user starting the recording of the Macro, then performing all the functions that need to be captured, and then stopping the recording. Once recorded, the user can run this macro whenever needed. Macros allow the automation of complex functions of a software. The most important aspect of macros is that anyone can automate complex processes without being a professional programmer. Macros are available in most CAD programs, and offer a simpler (although less powerful) alternative to automate complex tasks than by using APIs.

MOOC: Acronym standing for 'Massive Open Online Course.' It refers to an online course aimed at 'massive' participation via the web. Besides filmed lectures, readings, and problem sets, MOOCs often include user forums to support discussions and interactions among students and others. They also provide feedback, quizzes and other assignments and assessments. MOOCs were first introduced in 2006, but only started to gain traction until 2012-14. Popular websites offering MOOCs are: Coursera, Edx, and Udemy. Most MOOCs offer the possibility to 'audit' the course for free, however if the student wants an official recognition, this must be paid for. For more information about MOOCs see Koller, 2012.

Neutral File Formats: Term used to refer to computer file formats over which no CAD vendor holds rights. These file formats are often created by non-commercial organizations. Examples of neutral file formats are: DXF (2D) and IGES (3D). Neutral file formats are often used to transfer CAD models from the software of one vendor to another.

Parametric Modelling: An approach to creating computer models used in industrial design by using parameters and constraints. When these parameters and constraints are updated, the model updates automatically.

Physical Model-Making: In the context of Industrial Design, refers to the process of making physical models/mock-ups using materials such as foam, clay, wood, cardboard, paper, etc. These models are built at different stages of the design process of a new product in order to test different things aspects of it.

Polygonal Models: In the context of Computer Aided Design, a Polygonal Model, is a type of three-dimensional model, defined by a collection of two-dimensional polygons (generally triangles or quads), each of which forming a surface, and arranged next to each other to form a three-dimensional mesh. Polygonal models are the most basic type of models, and because of their versatility, polygonal models have been extensively used the entertainment industry. Common 3D-CAD modelling packages based on polygonal modelling include: Autodesk's Maya and 3DS Max and Maxon's Cinema 4D.

Procedural Generation: Term mostly used in the fields of computer graphics and videogames. It refers to the process of generating forms automatically using computer algorithms rather than following a Digital Handcrafting approach. It is a term closely related to Digital Morphogenesis and Generative Design, but precedes them in time.

Processing: Processing is a text computer programming language designed to work with images. Processing aims to offer a good balance between simplicity and capabilities. Processing has become popular in recent years, particularly among the community of artists interested in digital and computer art, and can be used to teach computer graphics and human computer interaction techniques. There is an extensive amount of resources to learn Processing online, and also a large support community. In addition, there is a vast array of libraries that can extend Processing's ability to work with sound, import/export 2D and 3D files. For more information about processing see: Casie and Fry (2007), and/or visit: Processing.org

Product Data Management (PDM): Refers to the process of collecting, storing, analysing and disseminating information generated by the multiple computer systems involved in New Product Development such as CAD and CAM.

Product Lifecycle Management (PLM): Refers to the process of managing the entire lifecycle of the product (often called a 'cradle to cradle' approach). Includes Design, Research & Development and Manufacturing as well as Marketing & Sales and Recycling (Cradle to Cradle).

Rapid Prototyping: refers to the processes of quickly making physical parts from CAD files for the purposes of testing and research and development. Rapid prototyping encompasses a variety of processes used to fabricate models and prototypes of parts for products using data from computer models. Most commonly refers to the creation of three-dimensional parts, but can also apply to the creation of two-dimensional parts from sheet materials. Rapid Prototyping processes can be substractive, such as when cutting from a block of material (i.e., CNC milling), or additive, such as in the case of 3D printing. The literature often (and mistakenly) uses the terms Rapid Prototyping and 3D printing as synonyms, however 3D printing is just one type of Rapid Prototyping. Also not to be confused with Rapid Manufacturing, which is the use of Rapid Prototyping techniques and equipment, but for the purpose of low-volume production rather than for prototyping and testing.

Rendering: In the context of three-dimensional CAD and Computer Graphics, refers to the generation of an image from a three-dimensional computer model, and is used to create product visualizations. Among other, the process involves; choosing a background, creating and/or positioning a camera, lighting the scene/model, adding materials to the model/scene, and setting the format and attributes of the rendered image. The process can be time-consuming, and requires of a number of trials before the desired rendered image can be achieved. Most CAD packages include this functionality, but there is also a wide array of standalone 3D rendering programs available that can import 3D models from a variety CAD packages as well.

Reverse Engineering: Term used to refer to the process of extracting/recovering design information from an existing product, and from which a product design specification could be produced. Design information extracted can include: dimensions, materials, methods of assembly, etc. In the context of Computer Aided Design, it often implies the creation of a 3D computer model from an existing physical product. This 'digitization' process is often aided with the use of equipment like 3D scanners.

Solid Models: In the context of Computer Aided Design, Solid models are defined by mathematical expressions that represent volumetric bodies such as a sphere, a cylinder, a cube, etc. Solid models can never be 'open,' thus, there is no way to unwrap a solid model like a polygon can be unwrapped. Unlike NURBs-based models, the surface of Solid models has no control points, therefore it cannot be 'sculpted' like NURBs models can. On the other hand, this also means that the surface definition is very stable and can be scaled down or up without showing issues. For this reason, Solid models are commonly used in CAD packages geared towards mechanical engineering. Most parametric modelling environments are based on Solid models due their characteristics.

Surface Model: In the context of Computer Aided Design, a Suface Model, is defined by mathematical expressions that define surfaces, and which can be represented graphically using computer graphics. Surface Models are infinitely thin, surfaces can be totally flat, they can be curved in one direction, or both (double curvature).

Surface models are often based on NURBS (non-uniform rational b-spline), thus the terms are often used interchangeably. NURBS are a special type of curve (which when arranged continuously form surfaces) defined by a mathematical equation. Common CAD software based on NURBS include: Autodesk's Alias (formerly Alias Wavefront) and McNeel's Rhinoceros 3D.

Surface Continuity: An important aspect of surface models, is surface continuity. In order to achieve smooth surfaces on products, it is important to be able to control the way that surfaces are joined and flow into other surfaces. This surface continuity is divided in four types: Positional (often expressed as G0), Tangential (G1), Curvature (G2) and Acceleration (G3), being this last the smoother one.

Teaching Machines: Teaching machines are devices used to present educational material to a person. The popularity of teaching machines reached its peak during the late 50s and 60s with the advent of programmed education as a result of the introduction of behaviourist theories of education. Teaching machines were suitable for programmed education because they could 'easily' present the student with educational materials according to the sequence of the program; this was useful particularly in the training of very specific duties like using Morse code. Teaching machines have taken different forms through time depending on technologies available; some of the very first were electro-mechanical and then electronic, but due to its interactive nature, soon teaching machines based on the computer became predominant. In a way a computer running education software like leapfrog's different educational programs could be considered to be a teaching machine, but nowadays the term is not widely used and may have negative connotations due to association with behaviourist education.

Turnkey CAD System(s)/Solution(s): As opposed to Generic CAD systems, turnkey systems/solutions, are Computer Aided Design Packages developed for a particular industry. Examples are Cadense's OrCAD for the electronics industry, Autodesk's Revit for the Architecture/Construction industries, Visi Shoes for the shoe industry, Gerber Accumark for the fashion industry, etc. In some cases, turnkey CAD solutions have been developed by the interested industry itself, such as ICEM surf, developed by Volkswagen in order to support its concept development operations.

Vector Graphic: A vector graphic in the context of Computer Aided Design and Computer Graphics, refers to images created using geometric entities, such as curves based on vectors. A vector can be described as a mathematical expression that indicates position and direction. Vectors have a starting point and an ending point, each with mathematical coordinates (X and Y). Vectors are 'resolution-proof,' that is, the quality of the image displayed/printed is always the same regardless of how big or small it is. Vectors can be thought of as arrows (and are graphically represented that way), the longer the arrow, the larger the value of the vector. The direction, is indicated by the direction the arrow points.

Wearable Smart Device: In general, wearable smart devices are clothes or accessories such as bracelets, belts, and necklaces which incorporate some kind of electronic technology. These devices represent a form of ubiquitous computing and communication aiming at integrating technology into everyday life in a seamless manner. While wearable smart devices are a form or wearable technology, not all wearable devices are 'smart,' however most often they come together with or can be connected to one, like in the case of a Bluetooth adapter and a smartphone. One of the biggest applications of these wearable smart devices is the monitoring of vital signs such as heartrate and temperature. Popular examples of smart wearable devices these days are Apple's iWatch and the Google Glass.

References:

Alcaide-Marzal, J., Diego-Más, J. A., Asensio-Cuesta, S., & Piqueras-Fiszman, B. (2013). An exploratory study on the use of digital sculpting in conceptual product design. *Design Studies*, *34*(2), 264-284.

Andrews, R. (Ed.). (2004). *The impact of ICT on literacy education*. Psychology Press.

Ashby, M. F., & Johnson, K. (2013). *Materials and design: the art and science of material selection in product design*. Butterworth-Heinemann.

Amit, G. (2010, Apr 11). *American Design Schools Are a Mess, and Produce Weak Graduates*. Retrieved from https://www.fastcompany.com/1662634/american-design-schools-are-a-mess-and-produce-weak-graduates

Association for Computational Creativity ACC (2014). *Computational Creativity.* Retrieved from: http://computationalcreativity.net/about/computational-creativity/

Bennett, E. (2012). A four-part model of informal learning: Extending Schugurensky's conceptual model. *Adult Education Research Conference* (pp. 24-31). Saratoga Springs, NY: Adult Education Research Conference: AERC.

Berry, M. *(2015). Computational thinking*. Retrieved from: https://www.youtube.com/watch?v=NDWy1SpuYNM&t=19s

BIDA. (2017, July 11). *BIDA Industrial Design graduate employability survey 2017* results. Retrieved June 13, 2018, from http://www.britishindustrialdesign.org.uk/news/2017/7/11/your-verdict-strong-liks-to-industry-fundamental-to-employability-to-graduate-industrial-designers

Bilalis, N. (2000). Computer Aided Design CAD. *INNOREGIO: dissemination of innovation and knowledge management techniques,* 1-27.

Bingimlas, K. A. (2009). Barriers to the successful integration of ICT in teaching and learning environments: A review of the literature. *Eurasia Journal of Mathematics, Science & Technology Education*, *5*(3).

Boden, M. A. (1998). Creativity and artificial intelligence. *Artificial Intelligence, 103*(1), 347-356.

Boden, M. A., & Edmonds, E. A. (2009). What is generative art? *Digital Creativity, 20*(1-2), 21-46.

Booth, W., Colomb, G., & Williams, J. (1995). *The Craft of Research.* Chicago: The University of Chicago Press.

Bryden, D. (2014). *CAD and rapid prototyping for product design.* Laurence King Publishers

Budd, J. (2011). Can Industrial Design Education Turn the Corner: Setting a New Trajectory for the Future of Industrial Design Education. In *Proceedings of the 2011 IDSA National Design Educator's Conference*.

Buckingham, D. (2007). Media education goes digital: an introduction. *Learning, Media and Technology*, *32*(2), 111-119.

Bürdek, B. E. (2005). *Design: History, theory and practice of product design*. Walter de Gruyter.

Bureau of Labor Statistics, U.S. Department of Labor. (2018, May 11). *How to Become an Industrial Designer.* Retrieved June 13, 2018, from https://www.bls.gov/ooh/arts-and-design/industrial-designers.htm#tab-4

Burleson, W. (2005). Developing creativity, motivation, and self-actualization with learning systems. *International Journal of Human-Computer Studies*, *63*(4), 436-451.

Burton, N., Brundrett, M., & Jones, M. (2008). *Doing Your Education Research Project.* London: SAGE.

Campbell, T., Williams, C., Ivanova, O., & Garrett, B. (2011). Could 3D printing change the world. *Technologies, Potential, and Implications of Additive Manufacturing, Atlantic Council, Washington, DC*.

Candy, L. (1997). Computers and creativity support: knowledge, visualisation and collaboration. *Knowledge-Based Systems, 10*(1), 3-13.

Cao, R., Bieng Chuah, K., Chung Chau, Y., Fai Kwong, K., & Yin Law, M. (2012). The role of facilitators in project action learning implementation. *The Learning Organization*, *19*(5), 414-427.

Chapman, G. (1995). *Computer Aided Design in Industrial Design Education. Journal of Art & Design Education, 14(1), 65-76.*

Chapman, G. (1999). *Intergraph MicroStation for 3D Industrial Design.* Documents of the AGOCG. Retrieved from: http://www.agocg.ac.uk/reports/graphics/26/node2.htm on October 2013.

Chase, S. C. (2005). Generative design tools for novice designers: Issues for selection. *Automation in Construction*, *14*(6), 689-698.

Chester, I. (2008). 3D-CAD: modern technology--outdated pedagogy? *Design and Technology Education: an International Journal*, 12(1).

Chip Taylor Communications., & VEA Productions. (2006). *CAD/CAM in the classroom*. Derry, N.H: Chip Taylor Communications.

Chomsky, N., Gardner, H., & Della Chiesa, B [Noam Chomsky Videos]. (2013, May 29). *The pedagoggy of the opressed* [Video file]. Retrieved from: http://www.youtube.com/watch?v=-S0w55BU7yg

Cohen, M. (2004). *Socratic Definitions.* Retrieved from: http://faculty.washington.edu/smcohen/320/socdef.htm on October 2013

Cohen, L., Manion, L., & Morrison, K. (2000). *Research Methods in Education.* London: Routledge Falmer.

Cole, T. (1994). *A New Approach to Teaching Computer-based Design Skills*. Documents of the AGOCG. Retrieved from: http://www.agocg.ac.uk/reports/graphics/26/node4.htm on October 2013.

Colton, S. (2009). Seven catchy phrases for computational creativity research. *Proceedings of the Dagstuhl Seminar: Computational Creativity: An Interdisciplinary Approach.*

Colton, S., Lopez de Mantaras, R., Stock, O., & others. (2009). Computational creativity: Coming of age. *AI Magazine, 30*(3), 11-14.

Colton, S., Wiggins, G. A., & others. (2012). Computational creativity: The final frontier? *ECAI*, (pp. 21-26).

Conner, B. P., Manogharan, G. P., Martof, A. N., Rodomsky, L. M., Rodomsky, C. M., Jordan, D. C., & Limperos, J. W. (2014). Making sense of 3-D printing: Creating a map of additive manufacturing products and services. *Additive Manufacturing*, *1*, 64-76.

Contreras, G. J., & Siu, K. W. M. (2015). Computer programming for all: A case-study in product design education. *Procedia-Social and Behavioral Sciences*, *182*, 388-394.

Cornock, S., & Edmonds, E. (1973). The creative process where the artist is amplified or superseded by the computer. *Leonardo*, 11-16.

Coughlan, M., Cronin, P., & Ryan, F. (2007). Step by Step Guide to Critiquing Research. Part 1: quantitative research. *British Journal of Nursing*, 658-663.

Creswell, J. (1994). *Research Design: Qualitative & Quantitative Approaches.* Thousand Oaks CA: SAGE.

Crosby, M. (2001). Working with people as an informal educator. In *Principles and Practice of Informal Education: Learning Through Life*, 54. London: Routledge

Cross, J. (2007). *Informal Learning: Rediscovering the natural pathways that inspire innovation and performance.* San Francisco: Wiley.

Cross, N. (2007). Designerly Ways of Knowing. Board of International Research in Design. *Basel: Birkhauser*, *41*.

Crouch, C., & Pearce, J. (2012). *Doing Research in Design.* London: Berg.

Crow, D. (2014). Why every child should learn to code. *The Guardian*, *7*.

Cuffaro, D. (2006). Process, Materials, and Measurements.

Dankwort, C. W., Weidlich, R., Guenther, B., & Blaurock, J. E. (2004). *Engineers' CAx education it's not only CAD. Computer-Aided Design, 36(14), 1439-1450.*

De Biswas, K. (2012). Sunglass: A Conversational Framework. *Dosya*, 30(29).

de Bono, E. [love0taxi]. (2012, Feb 4). *Edward de Bono Six Thinking Hats*. [Video file]. Retrieved from: https://www.youtube.com/watch?v=_5DK8XPUEaU

DeLoura, M., & Paris, R. (2013). *Don't Just Play on Your Phone, Program It*. Retrieved from: https://obamawhitehouse.archives.gov/blog/2013/12/09/don-t-just-play-your-phone-program-it

Denzin, N. (1978). *The Research Act: A Theoretical Introduction to Sociological Methods.* New York: Mcgraw-Hill.

Desbarats, G. (2010). *Effective 3D Design Collaboration.* Retrieved from: https://www.plm.automation.siemens.com/zh_cn/Images/100430-Siemens-NX_Whitepaper_tcm78-96048.pdf

DiTullo, M. (2014, 01 7). *Thoughts on the Term Industrial Design.* Retrieved from: http://www.core77.com/blog/essay/thoughts_on_the_term_industrial_design_26199.asp

Doin, G. (Director). (2012). *The Forbidden Education* [Motion Picture]. Argentina

Dönmez, S. (2013). Computer aided industrial design software selection in industrial product design education at Turkey using expert choice program. *Procedia-Social and Behavioral Sciences*, *106*, 682-689.

Dresselhaus, B. (2000). *ROI: Return on innovation*. Portland: Bill Dresselhaus Design Group.

Duggan, F. (2013, 04 23). *Why universities should acquire – and teach – digital literacy.* Retrieved from: http://www.theguardian.com/education/2013/apr/23/universities-should-teach-digital-literacy

Duncan-Howell, J. (2012). Digital mismatch: Expectations and realities of digital competency amongst pre-service education students. *Australasian Journal of Educational Technology*, 827-840.

Earle, J. H. (1994). *Engineering Design Graphics: AutoCAD Release 12*. Addison-Wesley Longman Publishing Co., Inc.

Efer, O. O. F. (2017). 8. Industrial Design: The Roles and Factors of Aesthetics, Modeling, Styling, Product Brand and Branding in Design/Design Education. *Review of Artistic Education*, *14*(1), 186-199.

Elliott, R., & Timulak, L. (2005). Descriptive and interpretive approaches to qualitative research. *A handbook of research methods for clinical and health psychology*, *1*, 147-159. Retrieved from: http://nideffer.net/classes/GCT_RPI_S14/readings/interpretive.pdf

Erkal, E. (2012). Thereminspace: The next challenge for digital design tools. *Dosya, 29(29), 56.*

Feenberg, A., & others. *(2012). Questioning technology.* London: Routledge.

Findeli, A. (2001). Rethinking design education for the 21st century: Theoretical, methodological, and ethical discussion. *Design issues*, *17*(1), 5-17.

Fraenkel, J., & Wallen, N. (1996). *How to Design and Evaluate Research in Education.* New York: McGraw Hill.

Friedman, K. (2014). *Writing for the PhD in Art and Design. Issues for Research Supervisors and Research Students.* Retrieved from: https://s3.amazonaws.com/academia.edu.documents/37407916/Friedman_RSW_2015_Writing_the_PhD_150423_.pdf

Garfield, L. (2016, Jul 1). *Under Armour 3D-printed soles for its new $300 sneakers with powder and lasers* Retrieved from http://uk.businessinsider.com/under-armour-3d-prints-its-new-architech-shoes-2016-6

Gesher, A. (2010, 03 08). *Friction in Human-Computer Symbiosis: Kasparov on Chess.* Retrieved from: https://www.palantir.com/2010/03/friction-in-human-computer-symbiosis-kasparov-on-chess/

Gibbs, G. [Graham R Gibbs]. (2014, Mar 11). *Crosstabulations and their Interpretation*. [Video file]. Retrieved from: https://www.youtube.com/watch?v=B6bqHNVd-Kw

Gilchrist, A. (2001) Principles and practice of informal education: Learning through life. In: Richardson, L. D. and Wolfe, M., eds. (2001) *Principles and practice of informal education: Learning through life.* London: Routledge. ISBN 9780415216906 Available from: http://eprints.uwe.ac.uk/770

Gonzales, R. (2012). The Forbidden Education [Film]. (G. Doin, Interviewer)

Gornick, N., & Grout, I. (2008, July). A dialogue on the future of design education. In *Changing the Change Conference Proceedings*, Turin (pp. 10-12).

Grix, J. (2010). *The foundations of research*. London: Palgrave Macmillan.

Gun, O. Y. (2012). Theorists of Design and Computation. *Dosya, 29(29),* 1-3.

Guzdial, M. (2008). Education Paving the way for computational thinking. *Communications of the ACM*, *51*(8), 25-27.

Hannah, B. (2004). *Becoming a Product Designer*. John Wiley & Sons.

Haughton, E. (2001). *A Piece of Cake*. Retrieved from: https://www.theguardian.com/education/2001/jan/09/itforschools.schools18

Henn, S. S. (2014). *Computers Are The Future, But Does Everyone Need to Code?* Retrieved from: http://www.npr.org/blogs/alltechconsidered/2014/01/25/266162832/computers-are-the-future-but-does-everyone-need-to-code

Husser, J. [Elon University Poll]. (2014, Sep 26). *How to read a cross tab*. [Video file]. Retrieved from: https://www.youtube.com/watch?v=Z711Kzb4nh0

IDSA. (2017). *What is ID?* Retrieved from: http://www.idsa.org/events/what-id

IDSA. (2018, Jan 15). *How they do it*. Retrieved from: http://www.idsa.org/education/how-they-do-it

Irvin, R. (2007). Information and communication technology (ICT) literacy: Integration and assessment in higher education. *Journal of Systemics, Cybernetics and informatics*, *5*(4), 50-55.

Jeffs, T. (2001). First Lessons: Historical perspectives on informal education. In L. Richardson, & M. Wolfe, *Principles and Practice of Informal Education* (pp. 34-50). London: RouletdgeFalmer.

JISC (2012) *JISC Developing Digital Literacies Programme: Summary of Project Baseline Reports.* Retrieved from: http://jiscdesignstudio.pbworks.com/w/file/fetch/53609142/Digital%20Literacies%20-%20summary%20of%20baseline%20reports%20i1.pdf

Kalpakjian, S., Vijai Sekar, K. S., & Schmid, S. R. (2014). *Manufacturing engineering and technology*. Pearson.

Katz, I. R. (2005). Beyond technical competence: Literacy in information and communication technology. *Educational Technology 45*(6), 44.

Keirl, S. (2004). Creativity, Innnovation and Life in the Lily-Pond: nurturing the design and technology family while keeping the alligators fed. *Journal of Design & Technology Education*, *9*(3), 145-157.

Kiernan, L., & Ledwith, A. (2014). Is design education preparing product designers for the real world? A study of product design graduates in Ireland. *The Design Journal*, *17*(2), 218-237.

Koller, D. [TED]. (2012, Aug 2). *What we're learning from online education*. [Video file]. Retrieved from: https://www.youtube.com/watch?v=U6FvJ6jMGHU

Kolko, J. (2005). New techniques in industrial design education. In *Design-System-Evolution, Proceedings of the 6th International Conference of The European Academy of Design, EAD06*.

Krish (2013). *Introduction to Computational Design.* Retrieved from: http://www.academia.edu/1488733/1._Introduction_to_Compuational_Design

López de Mántaras Badia, R. (2013). Computational creativity. *ARBOR Ciencia, Pensamiento y Cultura*, *189*(764), 10-3989.

Lostritto, C. (2012). The Definition, Necessity and Potential of Drawig Computation. *Dosya, 29(29)*, 61.

Loukides, M. (2014). *The Reson Everyone Should Learn to Code*. Retrieved from: http://radar.oreilly.com/2014/01/the-reason-everyone-should-learn-to-code.html

Lubart, T. (2005). How can computers be partners in the creative process: classification and commentary on the special issue. *International Journal of Human-Computer Studies, 63*(4), 365-369.

Luscombe, P. (2018). Rulers and Dividers: A Technology of Design. *Design Issues*, *34*(2), 5-19.

MacDonald, S., & Headlam, N. (2008). *Research Methods Handbook: Introductory guide to research methods for social research*. Centre for Local Economic Strategies. Retrieved from: http://www.cles.org.uk/wp-content/uploads/2011/01/Research-Methods-Handbook.pdf

Machado, P., & Cardoso, A. (2003). NEvAr--system overview. *Proceedings of Generative Art 2003*.

Mahoney, J. (2001). What is Informal Education. In L. Richardson, & M. Wolfe, *Principles and Practice of Informal Education* (pp. 17-33). London: Rouletdge Falmer.

Maldonado, T., Cantarell, F. S., & i Ramió, J. R. (1977). *El diseño industrial reconsiderado: definición, historia, bibliografía*. Barcelona: Gustavo Gili.

McCullough, M. (1997). Abstracting craft: the practiced digital hand. *The MIT Press, Cambridge, Massachusetts*.

Mateli, J. (2012). The Forbidden Education Documentary Film. (G. Doin, Interviewer)

Marcus, G., & Davis, E. (2014). *Do We Really Need to Learn to Code*. Retrieved from: http://www.newyorker.com/tech/elements/do-we-really-need-to-learn-to-code

Margolin, V. (2010). Doctoral education in design: problems and prospects. *Design Issues*, *26*(3), 70-78. doi:10.1162/DESI_a_00031

McMahon, C., & Browne, J. (1998). CAD/CAM Principles. *Practice and Manufacturing Management. Addison-Wesley*.

Meacham, J. (2013, 9 26). What Colleges Will Teach in 2025. *Time Magazine*. Retrieved from: http://nation.time.com/2013/09/26/the-class-of-2025/

Murray, M., & Perez, J. (2014). Unraveling the Digital Literacy Paradox: How Higher Education Fails at the Fourth Literacy. *Issues in Informing Science and Information Technology*, 85-100.

Musta'amal, A. H., Norman, E., Jabor, M. K., & Buntat, Y. (2012). Does CAD really encourage creative behaviors among its users: a case study. *Procedia-Social and Behavioral Sciences*, *56*, 602-608.

National Governors Association Center for Best Practices. (2010). *Read the Standards*. Retrieved from: http://www.corestandards.org/the-standards

Noonoo, S. (2012). The Third Revolution: "Disruptive Technology" is Reshaping How Students Learn. But How Can Schools Prepare for What They Can't Predict?. *THE Journal (Technological Horizons In Education)*, *39*(10), 4-7.

Norman, D. (2010). Why design education must change. *core77, 11*, 26. Retrieved from: https://www.researchgate.net/profile/Donald_Norman/publication/235700801_Wir_brauchen_neue_Designer_Why_Design_Education_Must_Change/links/54a2b47e0cf256bf8bb0d448.pdf

Norman, D. (2015). *Computer Science and Design: a Perfect Match*. Lecture, The Hong Kong Polytechnic University, School of Design.

Oppenheimer, T. (2003). *The flickering mind: The false promise of technology in the classroom, and how learning can be saved.* New York: Random House New York.

Oxman, R. (2006). Digital design thinking: in the new design is the new pedagogy. *Proceedings of the 11th International Conference on Computer Aided Architectural Design Research in Asia*. Kumamoto Japan, March 30th - April 2nd 2006, 37-46

Pearce, K. (2013, 12). *Why You Should Learn to Code*. Retrieved from: http://www.diygenius.com/learn-to-code-online/

Pease, A., & Colton, S. (2011). On impact and evaluation in computational creativity: A discussion of the Turing test and an alternative proposal. *Proceedings of the AISB symposium on AI and Philosophy.*

Pei, E., Campbell, I., & Evans, M. (2011). A taxonomic classification of visual design representations used by industrial designers and engineering designers. *The Design Journal*, *14*(1), 64-91

Peng, W. (2006, November). Brief discussion about the CAI class application in industrial design teaching. In *Computer-Aided Industrial Design and Conceptual Design, 2006. CAIDCD'06. 7th International Conference on* (pp. 1-4). IEEE.

Platt, T. (2011, 02 24). *Demand, supply, cost-curve elasticity and the impact of quality.* Retrieved from: http://plattperspective.wordpress.com/2011/02/24/demand-supply-cost-curve-elasticity-and-the-impact-of-quality/

Porter, A., McMaken, J., Hwang, J., & Yang, R. (2011). Common core standards: The new US intended curriculum. *Educational Researcher*, *40*(3), 103-116.

Porter, S., & James, C. (1999). *The use of ergonomics software in industrial design education.* Documents of the AGOCG. Retrieved from: http://www.agocg.ac.uk/reports/graphics/26/node15.htm

Prior, S. D., shen, S.-T., & Karamanoglu, M. (2007). The Problems With Design Education in the UK. *Proceedings of the 2nd Conference of the*

International Association of Societies of Design Research IASDR 2007 (p. 11). Hong Kong.

Qi, B., He, W., Yu, S., & Sun, X. (2010, September). Application of computer aided industrial design in model making course teaching. In *Educational and Information Technology (ICEIT), 2010 International Conference on* (Vol. 2, pp. V2-165). IEEE.

Reas, C., & Fry, B. (2007). *Processing: a programming handbook for visual designers and artists*. Cambridge. Mit Press.

Ritz, J. (2011). A Focus on Technological Literacy in Higher Education. *Journal of Technology Studies*, 1. Retrieved from: https://doi.org/10.21061/jots.v37i1.a.4

Rivas, M., & Contreras, G. J. (2016). The Neurolecturer as Model for Design Education: Fostering Creativity and. *Design Education for Fostering Creativity and Innovation in China* (pp. 212-226). IGI Global.

Robert, S. Root-Bernstein and Michele M. Root-Bernstein (2001). Sparks of Genius: The Thirteen Thinking Tools of the World's Most Creative People.

Rodriguez, W. E. (1992). Modeling of Design Ideas: Graphics and Visualization Techniques for Engineers. McGraw-Hill, Inc.

Rodgers, P., & Milton, A. (2011). *Product design*. Laurence King Publishing.

Rogers, H. [Harry Rogers]. (2017, Feb 12). *How Easy is It To Learn To Use CAD Software - Shed Talk 017* [Video file]. Retrieved from: https://www.youtube.com/watch?v=B6bqHNVd-Kw

Runberg, D. (2015). *The SparkFun Guide to Processing: Create Interactive Art with Code*. No Starch Press.

Rusten, G., & Bryson, J. (Eds.). (2009). *Industrial design, competition and globalization*. Springer.

Schoonmaker, S. J. (2002). *The CAD guidebook: A basic manual for understanding and improving computer-aided design* (Vol. 150): CRC Press.

Schugurensky, D. (2000). *The forms of informal learning: Towards a conceptualization of the field.* Retrieved from: https://tspace.library.utoronto.ca/bitstream/1807/2733/2/19formsofinformal.pdf

Self, J. (2012). *CAD versus Sketching, Why ask?* Retrieved from: https://static1.squarespace.com/static/514ba64be4b09fef8c0a47f1/t/5743f37b1bbee07d9fb996e6/1464071038473/Core77_02.pdf

Shapiro, J., & Hughes, S. (1996). Information Literacy as a Liberal Art. *Educom Revoew Sequence, 31*(2), pp. 31-35.

Shein, E. (2014). Should everybody learn to code? *Communications of the ACM*, *57*(2), 16-18.

Shneiderman, B., Fischer, G., Czerwinski, M., Resnick, M., Myers, B., Candy, L., & Jennings, P. (2006). Creativity support tools: Report from a US National Science Foundation sponsored workshop. *International Journal of Human-Computer Interaction*, *20*(2), 61-77.

Sioli, Lucilla, European Comission. (2010, 05). *Pillar VI: Enhancing digital literacy, skills and inclusion*. Retrieved from: http://ec.europa.eu/digital-agenda/en/our-goals/pillar-vi-enhancing-digital-literacy-skills-and-inclusion

Siu, K. W. M., & Contreras, G. J. (Eds.). (2016). Design Education for Fostering Creativity and Innovation in China. IGI Global.

Siu, K. W. M., & Contreras, G. J. (2016). 20 Years of Doctoral Research Studies in Design in Hong Kong. *Design Education for Fostering Creativity and Innovation in China* (pp. 227-244). IGI Global.

Siu, K. W. M., & Contreras, G. J. (2017). Disruptive Technologies and Education: Is There Any Disruption After All?. In *Educational Leadership and Administration: Concepts, Methodologies, Tools, and Applications* (pp. 757-778). IGI Global.

Siu, K. W. M., & Contreras, G. J. (2015). Personal Interaction and Informal Learning: The Case of China. In *Measuring and Analyzing Informal Learning in the Digital Age* (pp. 140-152). IGI Global.

Stallard, C. K., & Cocker, J. S. (2001). *The promise of technology in schools: The next 20 years.* US Rowman & Littlefield Education.

Starling, M. (1999). The use of I.T. in Art and Design. Documents of the AGOCG. Retrieved from: http://www.agocg.ac.uk/reports/graphics/26/node1.htm

Stiny, G., & Onur, G. (2012). An Open Conversation With George Stiny About Calculating and Design. *Dosya, 29(29)*, 61.

Study.com. (n.d.). *Industrial Designer: Education Requirements and Career Info*. Retrieved June 13, 2018, from: https://study.com/articles/Industrial_Designer_Education_Requirements_and_Career_Info.html

Tepper, F. (2017, Apr 7). *Adidas' latest 3D-printed shoe puts mass production within sight.* Retrieved from https://techcrunch.com/2017/04/07/adidas-latest-3d-printed-shoe-puts-mass-production-within-sight/

Thornburgh D. (2013). The Third Revolution: "Disruptive Technology" is Reshaping How Students Learn. (Stephen Noonoo, Interviewer)

Tiene, D., & Ingram, A. (2001). *Exploring Current Issues in Educational Technology with Free PowerWeb.* New York: McGraw-Hill.

Tierney, W. G. (2013). 2013 AERA Presidential Address: Beyond the Ivory Tower: The Role of the Intellectual in Eliminating Poverty. *Educational researcher*, *42*(6), 295-303.

Tierney, W. G., & Sablan, J. R. (2014). Examining college readiness. *American Behavioral Scientist 2014. 58*(8) 943–946.

Togay, A., Coşkun, M., Güneş, S., & Güneş, Ç. (2016). Computer aided design in education and its' interpretation through design thinking. *New Trends and Issues Proceedings on Humanities and Social Sciences, 2*(1), 328-337. doi:10.18844/gjhss.v2i1.315

Tovey, M., & Owen, J. (2000). Sketching and direct CAD modelling in automotive design. *Design Studies*, *21*(6), 569-588.

Tovey, M. (2002). Concept design CAD for the automotive industry. *Journal of Engineering design*, *13*(1), 5-18.

Trovalds, L. (2014). *Linux creator Linus Trovalds: I do not believe everyone should learn to code*. Retrieved from: http://venturebeat.com/2014/06/09/linux-creator-linus-torvalds-i-do-not-believe-everybody-should-learn-to-code/

Tsamis, A. (2012). Have You Seen Outside Barbarella's Window?: A software approach to CAD. *Dosya, 29(29), 47.*

Ulrich, K. T., & Pearson, S. (1998). Assessing the importance of design through product archaeology. *Management Science, 44(3), 352-369.*

Unver, E. (2006). Strategies for the transition to CAD based 3D design education*. Computer-Aided Design and Applications, 3(1-4), 323-330.*

Vaidyanathan, S. (2012). *Should Kids Learn to Code in Grade School?* Retrieved from: http://blog.acm.org/archives/csta/2012/12/should_kids_lea.html

Valtonen, A. (2007). *Redefining industrial design changes in the design practice in Finland.* University of Art and Design Helsinki.

Vanden Heuvel, A. [Google Glass]. (2013, May 3). *Explorer Story: Andrew Vanden Heuvel through Google Glass*. [Video file]. Retrieved from: https://www.youtube.com/watch?v=yRrdeFh5-io

Varinlioglu, Guzden, Halici, S. M., & Alaccam, Sema. (2015). Computational Approaches for Basic Design Education: Pedagogical Notes Based on an Intense Student Workshop*. Blucher Design Proceedings, 2(3), 576-580*.

Verene, D. P. (2013*)*. Does online education rest on a mistake*? Academic Questions, 26(3), 296-307.*

Vishnevsky, T., & Beanlands, H. (2004). Qualitative research. *Nephrology Nursing Journal*, *31*(2), 234.

Sullinger, B., Meridian Education Corporation., & Films Media Group. (2003). *What is CAD?*. Lawrenceville, NJ: Meridian Education Corp.

Wagstaff, K. (2012). Can we fix computer science education in America. *Time Magazine*. Retrieved from: http://techland.time.com/2012/07/16/can-we-fix-computer-science-education-in-america/

Wallas, G. (1976). Stages in the creative process. *The creativity question*, 69-73.

Willig, C. (2013). *Introducing qualitative research in psychology*. London: McGraw-Hill Education.

Winn, D., & Banks, F. (2012, June). CAD and Creativity-A New Pedagogy. In *PATT 26 Conference; Technology Education in the 21st Century; Stockholm; Sweden; 26-30 June; 2012* (No. 073, pp. 488-495). Linköping University Electronic Press.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences*, *366*(1881), 3717-3725.

Wing, J. [Columbia Journalism School]. (2014, Feb 10). *Computational Thinking*. [Video file]. Retrieved from: h*ttps://www.youtube.com/watch?v=U67utvZai8s&t=149s*

Whitley, K. N. (1997). Visual programming languages and the empirical evidence for and against. *Journal of Visual Languages & Computing*, *8*(1), 109-142.

Wolfe, M. (2001). Conversation. In L. Richardson, & M. Wolfe, *Principles and Practice of Informal Education* (pp. 124-137). London: RouletdgeFalmer.

Wong, K. V., & Hernandez, A. (2012). A review of additive manufacturing. *ISRN Mechanical Engineering, 2012*.

Ye, X., Peng, W., Chen, Z., & Cai, Y.-Y. (2004). Today's students, tomorrow's engineers: an industrial perspective on CAD education. *Computer-Aided Design, 36*(14), 1451-1460.

Yue, L. (2011). Seek for Distinction and Create Excellence: Promote the Art Design Education Quality of Higher Education with the Scientific Spirit. *Proceedings of the Information Management, Innovation Management and Industrial Engineering (ICIII), 2011 International Conference* (Vol. 3, pp. 149-152). IEEE.

Zammit, R., & Munoz, J. A. I. (2014). Has Digital Clay Finally Arrived?. *Computer-Aided Design and Applications*, *11*(sup1), S20-S26.