Design and Technology Education: An International Journal

24.1

Beyond Programming and Crafts: Towards Computational Thinking in Basic Education

Sinikka Pöllänen, University of Eastern Finland, Finland

Kari Pöllänen, Cloudia Ltd, Finland

Abstract

Continually increasing demands are being placed on the educational system to prepare students with technical skills due to the exponential implementation of information, technology and automation in the workforce. Students should work with design, problem-solving and computational methods and tools early on in their school lives in basic education and across diverse areas of learning. It has been argued that a fundamental understanding of technology requires computational thinking. However, teachers have difficulties integrating technology and programming into students' active learning in crafts. In this systematic literature review, the main aim is to view descriptions of programming through craft science-based concepts of craft labour and, thereafter, to seek examples to enable teaching programming in craft education during basic education. Considering the selection criteria to undertake the analysis, the final data set comprised of 10 articles dealing with programming and craft, and 68 articles describing the possibilities of combining crafting and programming in basic education. According to the results, it seems that contemporary multi-material and design-based holistic craft may encompass different forms of technology and programming such as prototyping, robotics, microcontrollers, 3D modelling, applications for documentation, visualisation, share-out and storytelling via multiple channels. These all help students to learn computational thinking as they start out with design and practical problems and proceed to technology-mediated programming skills. It is hoped that the findings will provide theoretical perspectives for practitioners and policymakers to see the mutual benefit arising from the integration of crafts, technology and computation in basic education.

Key words

basic education, computational thinking, craft, craft education, design and technology education, programming

Introduction

Computational thinking is argued to be a crucial skill for the 21st century due to the exponential implementation of information, technology and automation in the workforce (Wing, 2014). Thereby, increasing demands are being placed also on the educational system to prepare students with technical skills to live in our rapidly changing society (Pellegrino & Hilton, 2012). Students should work with design, problem-solving and computational methods and tools early on in their school lives in basic education and across diverse areas of learning (Papert, 1980; Barr & Stephenson, 2011; Foerster, 2016). Researchers (e.g., Wing, 2008; Webb et al., 2017) have noted that a fundamental understanding of technology requires computational thinking so that students recognise aspects of computation in the world that surrounds them. More widely, according to Wing (2006), everyone who uses computational tools and engages in step-by-step procedures needs computational thinking skills.

Aho (2011) defines computational thinking as thought processes that are involved in formulating problems in such a way that the solutions can be presented as computational steps and algorithms. Concretely, computational thinking means applying tools and techniques to understand, reason and solve problems in relation to both natural and artificial systems and processes (Denning, 2007), developing general-purpose thinking and design skills (Wing, 2006), and mental tools (Tedre & Denning, 2016). Educators have been confused by the multiple definitions of computational thinking (see Tedre & Denning, 2016) and subtle distinctions between computational thinking and programming (Barr & Stephenson, 2011). Viewing computational thinking as planning, learning and scheduling in the presence of uncertainty (Wing, 2006) may disclose the broad relevance of computational thinking for many other areas of the curriculum and for life in general (Webb et al., 2017). In this regard, as Wing (2006) has introduced, computational thinking is a kind of problem solving and a readiness to move between different levels of abstraction and decomposition, transformation and simulation. In other words, computational thinking means thinking recursively and choosing the appropriate representation for a problem or modelling a problem to make it tractable.

Barr and Stephenson (2011) have suggested that programming provides a transparent way of developing computational thinking. Nonetheless, Tedre and Denning (2016) note that coding cannot be accepted as the aim of computational thinking because coding as a part of the program-construction process does not require the highest level of computational thinking. Bellettini et al. (2014) have also warned that if the focus is only on coding or on using computer applications with ready-to-use recipes, there will not be enough space for a deeper understanding and creativity. Coding skills are, according to Tedre and Denning (2016), increasingly less important when what is actually more relevant is the ability to deal with design challenges and handle design tools. In this way, computational thinking provides insights into many areas of everyday life and into a wide range of disciplines (Wing, 2006). In this regard, despite that the terms computational thinking and programming as well as programming and coding are sometimes used interchangeably, in this study we do not see them as synonyms. We see computational thinking as a person’s analytical ability to formulate and solve problems, to design and implement ideas and to address those problems. In turn, programming is seen as the process of designing and building an executable computer program for accomplishing a specific computing task, after which the practical implementation of these solutions is concretized by coding (i.e. actions using some programming language).

To introduce computational thinking into schools, computer science standards have been defined (Computer Science Teachers Association [CSTA], 2011; Computing at School [CAS], 2018) and curriculum guidelines have been prepared by many educational governing bodies around the world (Tedre & Denning, 2016). In practice, however, after having agreed what should be taught in terms of computational thinking (Mannila et al., 2014), the learning objectives are also noted as being difficult to meet, as teachers experience that they do not have enough knowledge and there is an expectation that it always means the use of computer applications (Caspersen & Kölling, 2009; Bellettini et al., 2014; Isayama, Ishiyama, Relator & Yamazaki, 2016). Several studies (e.g., Derus & Ali, 2012; Piteira & Costa, 2013) have reported that students at all levels of education have difficulties with the first steps when learning to program. Students have orientation difficulties and lack the skills to analyse problems. It has been suggested that an effective way to learn hierarchical skills such as programming should be to begin with real-world problems and with learning the lower level skills first, then progressing upwards (Derus & Ali, 2012). Thus, Webb et al. (2017) have recommended using a spiral approach when implementing computational thinking in education: First, by problem solving using concrete meaningful objects to discover concepts, second by practising computational thinking about the objects and concepts to create algorithms and solutions, and third, by programming.

According to Ertmer and Ottenbreit-Leftwich (2013), despite most teachers having shifted away from implementation technology being about learning about technology as a subject, technology is still primarily used as a delivery tool, thus reinforcing old ways of learning. They argue that the teachers' traditional teaching beliefs are the primary reason behind neglecting the use of technology as a form of learning with technology as a cognitive partner (Jonassen, 1996). Several studies have noted that teachers, in general, need direct guidance and advice with concrete examples to expand their technology-based pedagogical knowledge, but also to put it into action in their practices (Bell, Rosamond, & Casey, 2012).

Barr and Stephenson (2011) have described examples in maths, science, social science and language for teachers to help them to identify core computational thinking concepts and capabilities. Several projects with simulation and modelling, robotics, or computer game design have been shared to support teaching computational thinking (e.g., Hutch, 2007; Bell et al., 2012; Foerster, 2016; Merkouris, Chorianopoulos, & Kameas, 2017). Webb et al. (2017) have proposed that computational thinking and abstract symbolic manipulations can be presented in a very concrete way and without starting with computers. However, according to Pellegrino and Hilton (2012), the teacher ought to develop a new understanding of the subjects that they are teaching before implementing computational thinking and programming more widely. Consequently, they should see where and how computational thinking already exists naturally and how it can be implemented to create real and virtual artefacts (Barr & Stephenson, 2011).

According to DeNicola (2016), programming has features that are relevant to crafts. Programming as a digital craft is quite well recognised in computer science (Tedre, 2018) but, in practice, it is not introduced in the same way with the theoretical concepts of crafts. Conversely, there is no comprehensive review that examines the unconscious aspects associated with teaching programming in crafts despite there being an increasing number of studies and more literature dealing with technology education in crafts. Thereby, the aim of this systematic literature review is to provide answers to the following research questions:

1) How is programming described through the craft science-based concepts of craft labour?

2) What kinds of examples are presented to teach programming in crafts during basic education?

We begin with a review of craft education in Finland where programming has been included in craft education in the National Core Curriculum for Basic Education (Finnish National Board of Education [FNBE], 2014) and where craft is based on multi-materiality and design-based holistic craft processes. This positions the paper within the existing literature and helps to deepen the theoretical nature of this review.

Craft education: The Finnish case

In Finland, the new National Core Curriculum for Basic Education (FNBE, 2014) has integrated technology into its national K-9 and K-12 curricula as a cross-curricular topic integrated into all subjects. The terms used in the curriculum are computational and algorithmic thinking, programming and ICT (information and communication technology) competence, which encompass a wide array of ICT-related skills (Kwon & Schroderus, 2017). Programming is integrated into crafts from the third grade onwards. Notably, there is an aim to increase the teaching of computational thinking to develop children's analytical abilities and through processes that also develop algorithmic thinking as a problem-solving strategy. More concretely, the ICT-related skills refer to the problem-solving process (i.e., programming) for accomplishing a specific computing task by using a programming language (i.e. coding, e.g., using Scratch). ICT competence means also responsible and safe use of ICT in communication and networking.

In Finland, craft is a combined single subject for all students in basic education (FNBE, 2014). As a school subject it has similarities with the design and technology education (D&T) and technology education in other countries. Finnish craft classes could also be considered as craft, design and technology education in international comparisons (Porko-Hudd, Pöllänen & Lindfors, 2018). Craft is a compulsory subject of two hours a week for all students from first grade to seventh grade. After seventh grade, it is optional. The curriculum discusses design-based holistic and multi-material craft, which includes technical work and textile work. It has its own objectives, but it is supposed to be implemented as open themes and with a holistic interdisciplinary approach. The self-expressive design and technology-based activities in crafts are required to be investigative, explanatory and experimental, and should be carried out by employing various visual, material, technical and manufacturing solutions. The students are also assumed to be able to utilise new technology and document their craft processes using ICT.

Craft is oriented to integrate design and skilled creative work with new technology (FNBE, 2014). Thereby, in multi-material craftwork, students are supposed to make tangible products through technical machines, devices, materials and systems that people encounter every day. The possibility of informing computational form-finding processes via the physical act of making serves as a point of departure for examining the relationship between material, procedure and form (Pöllänen & Urdziņa-Deruma, 2017). As McCullough (1996) has noted, hands help in acquiring knowledge of the world, of complex systems, concretely.

The craft process itself focuses on design, problem solving, understanding the restrictions, choices and possible defects, data collection and analysis across disciplines, and identifying and evaluating possible solutions (Pöllänen, 2009). This means generating a strategy and a sequence of orders (like an algorithm), implementing designs, and continuously evaluating the solutions by testing and debugging missteps, and modelling and running trial runs or simulations in order to formulate the solution into a prototype or an artefact (Anttila, 1993). In holistic crafts, reflecting and communicating in all phases of the process help in moving between levels of abstraction and help with innovation and exploration in order to design and manufacture artefacts. The holistic craft process refers to the thought processes and embodied thinking required to analyse the prerequisites and effectively solve a specific problem while articulating and formulating it, creating an expression of the solution and analysing the outcomes that are also the core of computational thinking or algorithmic thinking (Pöllänen & Urdziņa-Deruma, 2017; cf. Kwon & Schroderus, 2017).

Methodology

Our methodological option in this qualitative study is developed based on pragmatism (Gutek, 2014) and a systematic literature review (Miles, Huberman, & Saldana, 2013). After defining our research question, we followed the steps of the systematic literature review process: We determined the required characteristics of primary studies, we retrieved the sample of potentially relevant literature and, thereafter, we selected the pertinent literature and synthesised it in order to be able to report the results (Durach, Kembro, & Wieland, 2017).

We started the systematic review by searching the papers that focused on programming and craft. For the first research question concerning programming as a craft, two kinds of publications were chosen: At first, eight discretionary selected publications on craft science (in which the craft process as craft labour was described with scientific concepts) were chosen to act as background literature. Thereafter, to contextualise the study, publications describing the process of programming were screened through Google search: 524,000 results were found in 0.40 seconds. Google Scholar produced 3.67 million results. Consequently, the same concepts were used to find more limited results through the Finnish National Library Service (FINNA), which resulted in 507 hits. For this study, ten publications were chosen as the main data for the first research question based on the following inclusion criteria:

1. The paper focused on programming and coding and its title contained the words ‘craft’, ‘craftsmanship’, ‘craft process’, or ‘programming process’; and
2. The study was written in English.

For the second research question, to seek examples to teach programming in crafts in basic education, the literature review was not initially limited to anything (see Rodgers, 1993). At first, a search for the programming and crafting concepts through Google search produced 11.5 million results in 0.48 seconds. Among the search items there were practical tips, advice and tools. Therefore, these had to be excluded, even though the philosophical viewpoint in this review was pragmatic. Pairing the two concepts – programming and crafting – in Google Scholar still resulted in 43,200 hits. Focusing the concept of programming to ‘computational thinking’ and pairing it to crafts gave 1890 results in FINNA. For this study, 68 research papers, book chapters and conference papers were finally chosen on the strength of their titles – which either referred to craft and programming or to craft and learning technology – and these acted as the main data for the research question. Thereafter, we read the full texts by weighing up their content in terms of the inclusion criteria to verify selection.

To answer to the first research question, we focused on the descriptions of the programming process and on the similarities of the craft process described in the craft science literature. Thus, we reflected the main characteristics of the programming process through the craft science-based concepts of craft labour. Thereafter, to find examples relating to teaching programming in crafts in basic education, we read the full texts, itemising the widest variety of opportunities that combined programming with crafts (both for textiles and for technical work).

To avoid bias, we tried to capture all of the relevant papers to get a representative literature base for this study (see McGowan & Sampson, 2005). Thus, the final search was done through FINNA as it contained a wide range of databases managed by Finnish universities where the search for international scientific reviewed articles could be carried out. To avoid selection bias (Felson, 1992), the inclusion criteria were carefully considered from the perspective of computer science and craft science. Therefore, the related programming concepts (coding, computational thinking, technology) and craft (crafting, crafts, different techniques and materials in crafting e.g., weaving, e-textiles, robotics) were also taken into account while searching the relevant literature. The advantage of this approach is that both disciplines are represented, allowing for the possibility of discussing the related concepts. According to Schlosser, Wendt and Sigafoos (2007), expectancy bias may be reduced by engaging several researchers from a variety of sociocultural and educational backgrounds. It is hoped that the review results will offer more than a mere description of the existing literature, instead offering frameworks for nuanced scientific and subject boundaries at school.

Research results

The purpose of this study was to view descriptions of programming through the craft science-based concepts of craft labour and, thereafter, to seek examples to enable teaching programming in craft education. Thereby, the results initially present how programming is described as a craft. Thereafter, we introduce different examples of how technology and programming and, in particular, computational thinking can be taught in crafts.

Programming as a craft

Sennett (2008) has written about software as a modern craft, and Hansen, NørGård and Halskov (2014) have written about programmes as a digital craft, with Kaijima and Michalatos (2008) focusing on software as a craft. Additionally, Lindell (2014) has described code as design material that allows the metaphor of craft to be used for the activity of programming. Programming has also been called material crafting practice (McBreen, 2002; Martin, 2008) and designing and building (Tedre, 2018). More clearly, DeNicola (2016) has used craft-related concepts while talking about modern-day software, programming processes and programmers as digital artisans. He has linked them to concepts of materiality, aesthetics and embodiment. All these descriptions view programming as involving concepts that are also linked in craft science to craft such as creativeness, the ability to make aesthetic judgments, ethical considerations, iterative design, contingency planning, holisticity, refinement and reflection (Ihatsu, 2002).

Craftsmanship brings with it a metaphor of the skilled practitioner who is intent on mastering programming as a craft with pride and responsibility (McBreen, 2002). The programming process (Caspersen & Kölling, 2009; Webb et al., 2017) involves systematic executable steps to get a solution to a problem. This means that, like a craft process (Anttila, 1993), the process of programming is opposite to that of mechanical production (Lindell, 2014) and industrially produced artefacts (DeNicola, 2016). Craft contains the whole design and making process, where hand-controlled machines are used, and the activity is directed by thinking (Anttila, 1993). According to Ihatsu (2002), craft includes both the idea of the product, the product itself and the know-how for its realisation. The craft process can be understood not only as a way of making things by hand, but also as a way of thinking through the hand by manipulating a material as a means for logically thinking, learning and understanding through one's senses (Gray & Burnett, 2009). This kind of definition means that craft is a way of thinking through practices of all kinds (Adamson, 2007) and is one way of thinking intellectually (Sennett, 2008).

Just as Sennett (2008) described craft, Martin (2008) has also characterised programming with objective standards, striving for quality, having a sense of excellence, but also experiencing competitive pressure and frustration that comes from doing one's best. Hansen et al. (2014) have found in their study that coders' attitudes, when engaged in creative and expressive programming, had similarities to the relationship that craftsman had with their craft. In the same vein, the manufacturing metaphor and the concept of software as a product and as a beautiful articulation of a design (Martin, 2008) indicate that craft-based design and the processes involved are the key aspect, not the static endpoint (Ihatsu, 2002). Hansen et al. (2014) have noticed that the design and use of computer technology was conceptualised as craft engagement, craftsman rhythm and craftsmanship expressivity. Through persistent interaction with the material, the coder worked as a craftsman who immersed herself or himself in the form-giving activity. Implications of the programmers' activities as a craft means understanding materiality more than physical substance (Anttila, 1993), while hardware and software can be seen as manufactured things or objects of care with material aspects (DeNicola, 2016).

Programming is not only a mental exercise, as it is an embodied cognitive process that starts with our hands (Sennett, 2008). Thus, the programmer attains intimate knowledge of the potentials and limitations of the (raw)materials and technologies like a craft maker or designer does for aesthetic expression (Dormer, 1997). Material consciousness requires prolonged engagement, practice and patience with the work in crafts, where the constraints of the technology and materials slowly hone the worker's skills (Nimkulrat, 2012). This means that the programmer must gain the knowledge of the principles, patterns, practices and heuristics involved, just like a craftsman must do, so that the knowledge leads to the fingers and eyes (Martin, 2008). This is justified, as DeNicola (2016) has claimed that programming is shaped by material considerations that demand a sensitivity to aesthetics and entail embodied practice as a craft.

Computational thinking in crafts

The results of this study show how, for example, robotics and automation, microcontrollers, 3D modelling, prototyping, applications for documentation, visualisation, share-out and storytelling via multiple channels may help to build computational thinking-based learning experiences by using technology and programming in crafts. Thereby, crafts may help to teach computation, authentic computer science concepts (Angeli et al., 2016) as well as design and the basics of technology (Tedre & Denning, 2016) before the students are asked to code.

However, it is notable that, for example, the process of knitting itself has inherently procedural and computational aspects through knitting-pattern conventions and basic programming concepts (Craig, Petersen, & Petersen, 2012). In addition, weaving, crocheting and cross-stitching are intentionally models of mathematical designs and simplified programming language with code (see Eisenberg, 2002; Buechley, Eisenberg, Catchen, & Crockett, 2008). Practising in lower classes first with a pen and paper, and editing digital images, and making digital drawings and designs and programming may also help later with learning to visualise algorithms (Glenn & Larsen, 2012).

Programming could be practised in crafts, for example, in robotics and automation, and embedded systems can be used in the design and making the products (Nykänen & Lindh, 2012). Embodied target platforms such as robotics are noted as serving as motivators when learning building, programming and commanding (Blauvelt, Wrensch, & Eisenberg, 2000; Qi, Huang, & Paradiso, 2015; Merkouris et al., 2017), because the algorithms and programs are reified in concrete objects and not just as virtual characters on screen (Armoni, Meerbaum-Salant, & Ben-Ari, 2015). Today, there are several robotic technologies and educational robotics toolkits teaching programming and engineering concepts, for example, from LEGO (Mindstorms EV 3, WeDo) and VEX robotics (VEX EDR, VEX IQ). Thus, children can construct robots, conduct scientific experimentations, measure and log environmental data, create controllers and even build interactive art installations with microcontroller-based devices (e.g., GoGo Board). These constructions may enable the creation and programming of artistic works involving motion, light, sound and music (Rusk, Resnick, Berg, & Pezalla-Granlund, 2008).

The arrival of recent materials such as conductive fibres and threads, combined with accessible embedded computing platforms, have made it possible to combine microcontrollers with craft materials and processes (Mellis, Jacoby, Buechley, Perner-Wilson, & Qi, 2013) to build, for example, computationally enriched paperworks and textiles (Katterfeldt, Dittert, & Schelhowe, 2009; Kafai, Fields, & Searle, 2014). For example, open-source-based fabric-based construction kits with sewable microcontrollers, e-textiles, have enabled even beginners in crafts to design and build their own soft wearables (e.g., scarves, light-up T-shirts) and other artefacts (e.g., soft toys, jewels and other decorative items) with different kinds of materials and techniques, and with light and sound (Buechley et al., 2008; Rusk et al., 2008; Buechley & Hill, 2010; Peppler & Glosson, 2012; Weibert, Aal, von Rekowski, & Wulf, 2015). Sewable electronic components can be attached to fabric and connected to another with conductive thread, after which the completed circuit can be hooked up to a computer and programmed (Searle & Kafai, 2015). Students may, for example, in an open learning task, create an interactive garden with a wide variety of materials (Rusk et al., 2008; Millner & Baafi, 2011), or sew crafted stitches with conductive thread to create a binary pattern above and below the fabric surface, where the strobe light may mimic the created pattern (Tan, Keune, & Peppler, 2017). Microcontroller boards and toolkits (e.g., Micro:bit, The Circuit Playground, Maker Toolkit, Maker App, App Lab) with minimal software installation and wide platform compatibility allow the students to quickly build apps that communicate with external hardware. Virtual reality (VR) environments and applications are constantly improved and thus, students may use the virtual realm while designing, prototyping and simulating prior to their own making and construction process (Kauhanen et al., 2017).

Novel technologies and materials, and digital design tools, for example, with modelling, scanning and prototyping, and 3D printing have also become prevalent in craft and design (Goodman & Rosner, 2011; Weiler & Kuznetsov, 2017). Digital fabrication devices such as laser cutters and computer-controlled embroidery machines (Bucheley & Hill, 2010) enable the crafter to stay in the material realm while using technology in crafting. Students can use applications for design, electronics and programming, and simulate their artefact on a prototyping platform as well as see and present it in interactive animations (e.g., Tinkercad, Sketch.up, Proto.io, Prott). In becoming increasingly common, 3D printing methods create easy ways in which to personalise the fabrication of crafting colour 3D artefacts (Weiler & Kuznetsov, 2017) or to use 3D conveyor cutting and image-transfer techniques together in the same artefact. Crafts may also be integrated with mathematical subjects (Foerster, 2016) such as in computational crafts used in the design and construction of mathematical model sculptures made from paper (Eisenberg, 2002).

Technology may offer creative and facilitating opportunities for design and craft to visualise and manufacture designed objects, for example, a knitting visualiser connects knitting and code (Yang, 2017), and weaving as a traditional technology to make products by hand can be customised with computational tools, rapid prototyping, 3D and computer-aided weaving (Tao et al., 2016). Computer-aided design and computational tools that enable two-dimensional (2D) patterns to be viewed in 3D, or vice versa, 3D designs to be obtained as 2D patterns, will give technology-based examples alongside traditional methods in crafting (Martin, 2015; Tao et al., 2016). Additionally, 3D body-scanner programs and styling simulators may introduce computer-based schematic systems to design education (Pursiainen, 2011). Revitalising craft culture as digital craft even helps to proliferate up-to-date crafting and rare handicraft techniques (Goodman & Rosner, 2011), for example, like the ancient marbling technique that uses flowing patterns of paint directly on the surface of water that are then captured on paper, which may now be made by using digitally controllable electromagnets (Nitsche, Quitmeyer, Farina, Zwaan, & Nam, 2014).

Several applications for documentation, and image and video tools make students' designs, crafted artefacts and crafting processes visible with texts, photos and videos. Students can visualise, share and co-develop their designs in collaboration, and make interactive presentations and portfolios (e.g., Onenote, Book Creator, Microsoft Sway). In addition, QR codes or linking applications (e.g., Thinglink) may be used in a new way in crafts where students can create their own codes with a generator that helps to record the student's craft process in illustrated steps (Jaatinen, Ketamo, & Lindfors, 2017). Mobile applications for smartphones aid craft processes to become multimodal stories with documentation, communication and instruction for a process-based workflow (Wiklund-Engblom, Hartvik, Hiltunen, Johansson, & Porko-Hudd, 2015).

Recently, many museums and archives have allowed their materials to be used, for example, FINNA is a service for the exploitation of the cultural heritage having an application for a programming interface to be used for one's own applications. Social media has helped designers, artists, programmers and educators to collaborate and to show how openly available materials on the Internet can be reused in creative ways (see http://hack4.fi/2015-2/hackathon/projects/). Transparency is promoted by a creative common licence where anyone can be associated with their own output. This role played by open materials, technology, digital resources, and design with computers and mobile software also suggests a transition from purely physical to digital practices in crafts (Rosner, 2010).

Fernaeus, Murer, Tsaknaki and Belenguer (2014) have found that students use materials creatively and productively in interdisciplinary projects that combine craft and natural materials, mechanical parts and programmable devices. Thereby, the students created and learned through hands-on activities about the role of materials in the design process, but they also learned how to make the artefacts and accessories interactive. This inspired both girls and boys and working in teams. Katterfeldt et al. (2009) have noticed that through the computation construction activities, students became more self-confident in dealing with technology and were able to draw relations between their own creations and technologies present in their environment. In the same vein, Blikstein (2013) has noted that students felt proud when working with computer-controlled tools. Eisenberg (2002) has insisted that craft activities have both intellectual and emotional affordances that are usually relatively lacking in computer-based education. Accordingly, creative hands-on experiences in a familiar setting have helped in overcoming misconceptions about computation (Pollock, McCoy, Carberry, Hundigopal, & You, 2004).

Conclusion

It seems that parallels between programming and crafting are convincing. Viewing computational thinking as craft with problem solving and design, and a readiness to move between different levels of abstraction and decomposition, transformation and simulation (see Pöllänen & Urdziņa-Deruma, 2017; cf. Wing, 2006; Kwon & Schroderus, 2017) may help in selecting appropriate representations and achievements for tractable hands-on activities to concretise programming in basic education. Computationally-enhanced craft education can integrate the physical and virtual worlds, generating different artefacts and engaging different skills and the use of them in creative ways (Wood, Rust, & Horne, 2009). Consequently, novel output devices and different kinds of materials may construct new domains of technology with a combination of crafts and computation that can render both activities even more valuable (Blauvelt et al., 2000; Mellis et al., 2013; Kafai et al., 2014).

It seems that ICT offers the opportunity to use different learning materials, learning platforms, drawing programs, editing digital images and making drawings, videos and designs to support students' own innovations and designs in crafts. A media-rich software development environment for novices with an emphasis on an iconic style of programming may serve as a motivating pedagogical instrument that helps foster computational thinking and forms a precursor for programming activities (Resnick et al., 2009; see Armoni et al., 2015). Technology and programming may shift the focus in craft education to the design and use of different materials, tools and technologies to create new knowledge for solving complex problems. The findings show that by connecting craft education to modelling, visualisation, simulation and printing, and to new materials and applications combined with accessible embedded computing platforms, the students may learn craftsmanship-based actions and the basis for computational thinking in different kinds of authentic technology-based projects.

Blikstein (2013) has found that students were delighted when working with computer-controlled tools as they could make products that look good and, compared to hand-crafted objects, the process was easy and fast. Yet, making multiple identical or nearly identical items does not meet the criteria of a design-based holistic craft process, nor does it develop the skills of craftsmanship and computational thinking. When copying and using computer applications with ready-to-use recipes, there will not be enough space for deeper understanding and creativity (Bellettini et al., 2014). Technology ought to be implemented in such a way that it boosts students' problem-solving and thinking skills, as in a design-based craft process (Pöllänen, 2009), avoiding ready-made lessons with step-by-step, recipe-like models, and enabling learning through technology as a cognitive partner (see Jonassen, 1996; Resnick & Rosenbaum, 2013). Thus, it is not enough to programme a robotic toy to move as desired, to focus attention on the laboratory equipment, or to view crafts as routine-based craft making (see Pöllänen & Urdziņa-Deruma, 2017).

According to Martin (2015), the risk of a tool-centric technology-based approach is the neglect of the multifaceted understanding of design-based making and the real mindset of the making by hand. In turn, designing, creating and programming in crafts may allow for the creation of personally meaningful projects (Katterfeldt et al., 2009). Enabling students to design and program artistic creations that integrate, for example, light, sound, music and motion may expand the way in which technology, computational thinking and physical computing can be introduced in future-based educational settings. Programming must relate to the real world and especially to the students' interests and thoughts to overcome the initial difficulties in learning to program (Katterfeldt et al., 2009; Armoni et al., 2015) as well as gender-based attitudes towards technology (Hutch, 2007; Merkouris et al., 2017).

Clearly, design-based holistic and multi-material craft projects with playful hands-on learning experiences and abstractions with modelling, visualisation, simulation, printing, automation, robotics and programming can be used to teach computational thinking in crafts. The projects can be designed to start with an easy level first to familiarise students with the concepts with tips to solve questions, and then progress in difficulty with interesting themes and graphical programming to facilitate the transition to the more abstract representational style of programming languages (see Rusk et al., 2008; Isayama et al., 2016; Merkouris et al., 2017). Generic interactive programming platforms (Quinson & Oster, 2015) enable teachers to create specific programming microworlds that are adaptable to the goals of learning.

The presented findings describe that contemporary multi-material and design-based holistic craft may encompass different forms of technology and may thus help students to learn computational thinking by starting with design and practical problems and proceeding to technology-mediated programming skills (see Tedre, 2018). Thus, it is hoped that teachers see computational thinking more from a holistic perspective as a practice and through working with different kinds of materials. More widely, despite the case being based on Finland’s experience, we hope that the findings will give theoretical perspectives for practitioners and policymakers to see the mutual benefit of the integration of crafts, design, technology and computation in basic education.

Acknowledgments

The project is supported by the Ministry of Education and Culture, Finland (Grant No. 50185-OKM/INNOKOMP 2017-2019).

References

Adamson, G. (2007). *Thinking through craft*. Oxford, UK: Berg.

Aho, A. V. (2011, January). What is Computation? Computation and computational thinking. *Ubiquity,* January 2011. Retrieved from https://ubiquity.acm.org/article.cfm?id=1922682

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology & Society, 19*(3), 47–57. Retrieved from https://www.jstor.org/stable/jeductechsoci.19.3.47

Anttila, P. (1993). *Käsityön ja muotoilun teoreettiset perusteet* [The theoretical basis for craft and craft design]. Helsinki: WSOY.

Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From scratch to “real” programming. *ACM Transactions on Computing Education*, *14*(4), 1–15. doi:10.1145/2677087

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 48–54. Retrieved from https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/BarrStephensonInroadsArticle.pdf

Bell, T., Rosamond, F., & Casey, N. (2012). Computer science unplugged and related projects in math and computer science popularization. In H. L. Bodlaender, R. Downey, F. V. Formin, & D. Marx (Eds.), *The multivariate algorithmic revolution and beyond: Essays dedicated to Michael R. Fellows on the occasion of his 60th birthday* (pp. 398–456). Berlin, Heidelberg: Springer-Verlag.

Bellettini, C., Lonati, V., Malchiodi, D., Monga, M., Morpurgo, A., Torelli, M., & Zecca, L. (2014). Informatics education in Italian secondary school. *ACM Transactions on Computing Education,14*(2), 1–5. Retrieved from https://boa.unimib.it/retrieve/handle/10281/53500/80479/Informatics%20Education%20in%20Italian%20Seconfary%20Schools.pdf

Blauvelt, G., Wrensch, T., & Eisenberg, M. (2000). Integrating craft materials and computation. *Knowledge-Based Systems, 13*(7–8), 471–478. https://doi.org/10.1016/S0950-7051(00)00063-0

Blikstein, P. (2013). Digital fabrication and “making” in education: The democratization of invention. In J. Walter-Herrmann & C. Büching (Eds.), *FabLabs: Of machines, makers and inventors* (pp. 203–222). Bielefeld: Transcript.

Buechley, L., Eisenberg, M., Catchen, J., & Crockett, A. (2008). The LilyPad Arduino: Using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. *Proceedings of the Conference on Human Factors in Computing Systems* (pp. 423–432). New York, NY: Association of Computing Machinery.

Buechley, L., & Hill, B. M. (2010). LilyPad in the wild: How hardware’s long tail is supporting new engineering and design communities. *Proceedings of the 8th ACM Conference on Designing Interactive Systems* (pp. 199–207). New York, NY: Association of Computing Machinery.

Caspersen, M. E., & Kölling, M. (2009). STREAM: A first programming process. *ACM Transactions on Computing Education, 9*(1), 1–29. Retrieved from http://cs.au.dk/~mic/publications/journal/34--toce2009.pdf

Craig, M., Petersen, S., & Petersen, A. (2012). Following a thread: Knitting patterns and program tracing. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, Raleigh, NC*, 233–238. doi:10.1145/2157136.2157204

Computing at School. (CAS). (2018). *Computing at school*. Retrieved from https://www.computingatschool.org.uk/

Computer Science Teachers Association, (CSTA). (2011). *CSTA K-12 computer science standards*. New York, NY. Retrieved from https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/Docs/Standards/CSTA_K-12_CSS.pdf

DeNicola, L. (2016). Forging source: Considering the craft of computer programming. In C. M. Wilkinson-Weber & A. O. DeNicola (Eds.), *Critical craft: Technology, globalization, and capitalism* (pp. 35–55). London: Bloomsbury Academic.

Denning, P. J. (2007). Computing is a natural science. *Communications of the ACM, 50*(7), 375–415. Retrieved from https://www.ic.unicamp.br/~wainer/cursos/2s2006/epistemico/denning-nat.pdf

Derus, S. R., & Ali, A. Z. M. (2012). Difficulties in learning programming: Views of students. *Proceedings of the International Conference on Information and Engineering, Singapore,* 74–78.

Dormer, P. (1997). *Craft and the Turing Test of practical thinking*. Manchester: Manchester University Press.

Durach, C. F., Kembro, J., & Wieland, A. (2017). A new paradigm for systematic literature reviews in supply chain management. *Journal of Supply Chain Management, 53*(4), 67–85. https://doi.org/10.1111/jscm.12145

Eisenberg, M. (2002). Output devices, computation, and the future of mathematical crafts. *International Journal of Computers and Mathematical Learning, 7*(1), 1–44.

Ertmer, P. A., & Ottenbreit-Leftwich, A. (2013). Removing obstacles to the pedagogical changes

required by Jonassen's vision of authentic technology-enabled learning. *Computers & Education, 64*(5*)*, 175–182. https://doi.org/10.1016/j.compedu.2012.10.008

Fernaeus, Y., Murer, M., Tsaknaki, V., & Belenguer, J. S. (2014). Handcrafting electronic accessories using "raw" materials. *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction, Munich, Germany,* 369–372.

Felson, D. T. (1992). Bias in meta-analytic research. *Journal of Clinical Epidemiology*, *45*(8), 885–892.

Foerster, K-T. (2016). Integrating programming into the mathematics curriculum: Combining scratch and geometry in grades 6 and 7. *Proceedings of the 17th Annual Conference on Information Technology Education*, *Boston, MA*, 91–96.

Finnish National Board of Education. (FNBE). (2014). *Perusopetuksen opetussuunnitelman perusteet* [National core curriculum for basic education]. The Finnish National Board of

Education. Retrieved from http://www.oph.fi/download/163777_perusopetuksen_opetussuunnitelman_perusteet_2014.pdf

Glenn, J., & Larsen, E.F. (2012). *Unbored: The essential field guide to serious fun*. London: Bloomsbury.

Goodman, E., & Rosner, D.K. (2011). From garments to gardens: Negotiating material relationships online and "by hand." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vancouver, BC, Canada*, 2257–2266.

Gray, C., & Burnett, G. (2009). Making sense: An exploration of ways of knowing generated through practice and reflection in craft. *Proceedings of the Crafticulation and Education Conference, NordFo, Helsinki, Finland*, 44–51.

Gutek, G. (2014). *Philosophical, ideological, and theoretical perspectives on education*. New Jersey, NJ: Pearson.

Hansen, N.B., NørGård, R.T., & Halskov, K. (2014). Crafting code at the demo-scene. *Proceedings of the Conference on Designing Interactive Systems, Vancouver, BC, Canada*, 35–38.

Hutch, F. (2007). Learning programming with Erlang or learning Erlang with ladybirds. *Proceedings of the ERLANG 2007 SIGPLAN Workshop, Freiburg, Germany*, 93–99.

Ihatsu, A-M. (2002). *Making sense of contemporary American craft.* Publications in Education No. 73. Joensuu, Finland: University of Joensuu.

Isayama, D., Ishiyama, M., Relator, R., & Yamazaki, K. (2016). Computer science education for primary and lower secondary school students. *ACM Transactions on Computing Education*, *17*(1). doi:10.1145/2940331

Jaatinen, J., Ketamo, H., & Lindfors, E. (2017). Pupils' activities in a multimaterial learning environment in craft subject: Pilot study using an experience sampling method based on a mobile application in classroom settings. *Techne Series A, 24*(2), 32–49.

Jonassen, D.H. (1996). *Computers in the classroom: Mindtools for critical thinking*. Columbus, OH: Merrill/Prentice Hall.

Kafai, Y.B., Fields, D.A., & Searle, K. (2014). Electronic textiles as disruptive designs: Supporting and challenging maker activities in schools. *Harvard Educational Review, 84*(4), 532–556. doi:10.17763/haer.84.4.46m7372370214783

Kaijima, S., & Michalatos, P. (2008). Simplexity, the programming craft and architecture production. *Proceedings of the First International Conference on Critical Digital: What Matter, Cambridge, Harvard University Graduate School of Design*, 181–194. Retrieved from https://cumincad.architexturez.net/system/files/pdf/cdc2008_181.content.pdf

Katterfeldt, E., Dittert, N., & Schelhowe, H. (2009). EduWear: Smart textiles as ways of relating computing technology to everyday life. *Proceedings of the International Conference on Interaction Design and Children Conference* (pp. 9–17). Como, Italy: Association of Computing Machinery.

Kauhanen, O., Väätäjä, H., Turunen, M., Keskinen, T., Sirkkunen, E., Uskali, T., & Karhu, J. (2017). Assisting immersive virtual reality development with user experience design approach. *Proceedings of the 21st International Academic Mindtrek Conference*, *Tampere, Finland*, 127–136.

Kwon, S., & Schroderus, K. (2017). *Coding in schools: Comparing integration of programming into basic education curricula of Finland and South Korea.* Finnish Society on Media Education. Retrieved from http://mediakasvatus.fi/wp-content/uploads/2018/06/Coding-in-schools-FINAL-2.pdf

Lindell, R. (2014). Crafting interaction: The epistemology of modern programming. *Personal and Ubiquitous Computing, 18*(3), 613–624. doi:10.1007/s00779-013-0687-6

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in K-9 education. *Proceedings of the ACM Working Group Reports of the 2014 Innovation and Technology in Computer Science Education Conference, New York*, 1–29.

Martin, R.C. (2008). *Clean code: A handbook of agile software craftsmanship.* New Jersey, NJ: Prentice Hall.

Martin, L. (2015). The promise of the maker movement for education. *Journal of Pre-College Engineering Education Research*, *5*(1), 30–39. Retrieved from https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1099&context=jpeer

McBreen, P. (2002). *Software craftsmanship: The new imperative*. Boston, MA: Addison Wesley.

McGowan, J., & Sampson, M. (2005). Systematic reviews need systematic searchers. *Journal of the Medical Library Association, 93*(1), 74–80.

McCullough, M. (1996). *Abstracting craft: The practiced digital hand*. Cambridge, MA: MIT Press.

Mellis, D A., Jacoby, A., Buechley, L., Perner-Wilson, H., & Qi, J. (2013). Microcontrollers as material: Crafting circuits with paper, conductive ink, electronic components, and an “untoolkit.” *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, *Barcelona, Spain*, 83–90. https://dl.acm.org/citation.cfm?id=2460638

Merkouris, A., Chorianopoulos, K., & Kameas, A. (2017). Teaching programming in secondary education through embodied computing platforms: Robotics and wearables. *ACM Transactions on Computing Education, 17*(2), 1–22. doi:10.1145/3025013

Miles, M.B., Huberman, A.M., & Saldana, J. (2013). *Qualitative data analysis: A methods sourcebook.* Los Angeles, CA: Sage.

Millner, A., & Baafi, E. (2011). Modkit: Blending and extending approachable platforms

for creating computer programs and interactive objects. *Proceedings of the International Conference on Interaction Design and Children*. Michigan, MI: Ann Arbor, 250–253. Retrieved from http://web.media.mit.edu/~millner/papers/millner_baafi_IDC_2011.pdf

Nimkulrat, N. (2012). Hands-on intellect: Integrating craft practice into design research. *International Journal of Design, 6*(3), 1–14.

Nitsche, M., Quitmeyer, A., Farina, K., Zwaan, S., & Nam, H.Y. (2014). Teaching digital craft. *Proceedings of the Conference on Extended Abstracts on Human Factors in Computing Systems, Toronto, ON, Canada*, 719–730.

Nykänen, J., & Lindh, M. (2012). Robotics and automation in primary teacher education: Changing practices in the faculty of education at the University of Oulu, Finland. In Marc J. de Vries (Ed.), *Pupils Attitudes Toward Technology* (pp. 19–42). Retrieved from https://www.iteea.org/File.aspx?id=39522&v=531c4707

Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. New York, NY: Basic Books.

Pellegrino, J., & Hilton, M.L. (Eds.). (2012). *Education for life and work: Developing transferable knowledge and skills in the 21st century*. Washington, DC: The National Academic Press.

Peppler, K., & Glosson, D. (2012). Stitching circuits: Learning about circuitry through E-textile materials. *Journal of Science and Educational Technology, 22*(5), 751–763.

Piteira, M., & Costa, C. (2013). Learning computer programming: Study of difficulties in learning programming. *Proceedings of the International Conference on Information Systems and Design of Communication*, *Lisbon, Portugal*, 75–80.

Pollock, L., McCoy, K., Carberry, S., Hundigopal, N., & You, X. (2004). Increasing high school girls' self confidence and awareness of CS through a positive summer experience. *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*, *Norfolk, VA*, 185–189.

Porko-Hudd, M., Pöllänen, S., & Lindfors E. (2018). Common and holistic crafts education in Finland. Techne Series. *Research in Sloyd Education and Craft Science, 25*(3), 26-38.

Pursiainen, M. (2011). *Vaatemuotoilu kehoskannauksen valossa* [Garment design in the light of body scanning]. Lapin yliopiston taiteiden tiedekunta, Sarja C, Katsauksia ja puheenvuoroja No. 37.

Pöllänen, S. (2009). Contextualising Craft: Pedagogical Models for Craft Education. International *Journal of Art and Design Education, 28*(3), 249-260.

Pöllänen, S., & Urdziņa-Deruma, M. (2017). Future-Oriented Reform of Craft Education: The Cases of Finland and Latvia. In E. Kimonen & R. Nevalainen (Eds.)*, Reforming Teaching and Teacher Education: Bright Prospects for Active Schools* (pp. 117-144). Rotterdam, NL: Sense Publishers.

Qi, J., Huang, A., & Paradiso, J. (2015, June). *Crafting technology with circuit stickers*. Paper presented at the 14th International Conference on Interaction Design and Children (pp. 438–441). Association for Computing Machinery, New York.

Quinson, M., & Oster, G. (2015). A teaching system to learn programming: The programmer's learning machine. *Proceedings of the Conference on Innovation and Technology in Computer Science Education* (pp. 260–265). Vilnius, Lithuania: Association for Computing Machinery.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., … Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), 60–67. doi:10.1145/1592761.1592779

Resnick, M., & Rosenbaum, E. (2013). Designing for Tinkerability. In M. Honey & D.E. Kanter (Eds.), *Design, make, play: Growing the next generation of STEM innovators* (pp. 163–181). New York, NY: Routledge.

Rodgers, B.L. (1993). Concept analysis: An evolutionary view. In B. L. Rodgers & K. A. Knafl (Eds.), *Concept development in nursing* (pp. 73–92). Philadelphia: W. B. Saunders.

Rosner, D.K. (2010). Mediated crafts: Digital practices around creative handwork. *Proceedings of the Extended Abstracts on Human Factors in Computing Systems, Atlanta, GA*, 2955–2958.

Rusk, N., Resnick, M., Berg, R., & Pezalla-Granlund, M. (2008). New pathways into robotics: Strategies for broadening participation. *Journal of Science Education and Technology*,

17(1), 59–69. Retrieved from https://web.media.mit.edu/~mres/papers/NewPathwaysRoboticsLLK.pdf

Schlosser, R.W., Wendt, O., & Sigafoos, J. (2007). Not all systematic reviews are created equal: Considerations for appraisal. *Evidence-Based Communication Assessment and Intervention*, *1*(3), 138–150.

Searle, K.A., & Kafai, Y.B. (2015). Boy's needlework: Understanding gendered and indigenous perspectives on computing and crafting with electronic textiles. *Proceedings of the Eleventh Annual International Conference on International Computing Education Research* (pp. 31–39). Retrieved from https://www.researchgate.net/publication/300117018_Boys%27_Needlework_Understanding_Gendered_and_Indigenous_Perspectives_on_Computing_and_Crafting_with_Electronic_Textiles

Sennett, R. (2008). *The craftsman*. New Haven, CT & London: Yale University Press.

Tao, Y., Lu, N., Zhang, C., Wang, G., Yao, C., & Ying, F. (2016). CompuWoven: A computer-aided fabrication approach to hand-woven craft. Late-breaking work: Interaction in specific domains. *Proceedings of the Conference on Extended Abstracts, San Jose, CA,* 2328–2333.

Tan, V., Keune, A., & Peppler, K. (2017). The materiality of design in e-textiles. In S. Goldman & Z. Kabayadondo (Eds.), *Taking design thinking to school: How technology and design can transform teachers, learners, and classrooms* (pp. 180–194). New York, NY: Routledge.

Tedre, M. (2018). The nature of computing as a discipline. In S. Sentace, E. Barendsen, & C. Schulte (Eds.), *Computer science education: Perspectives on teaching and learning in school* (pp. 5–18*)*. London: Bloomsbury Academic.

Tedre, M., & Denning, P. J. (2016). The long quest for computational thinking. *Proceedings of the 16th Koli Conference on Computing Education Research*, *Finland*, 120–129.

Webb, M., Davis, N., Bell, T., Katz, Y.J., Reynolds, M.N., Chambers, D.P., & Syslo, M.M. (2017). Computer science in K-12 school curricula of the 2lst century: Why, what and when? *Education and Information Technologies, 22*(2), 445–468. Retrieved from https://link.springer.com/article/10.1007/s10639-016-9493-x

Weibert, A., Aal, K., von Rekowski, T., & Wulf, V. (2015). ”Hey, can we make that, please?”: On craft as a means to cross-cultural community-building. *The Journal of Community Informatics, 11*(2), 1–9. Retrieved from http://ci-journal.org/index.php/ciej/article/view/1188

Weiler, J., & Kuznetsov, S. (2017). Crafting colorful objects: A DIY method for adding surface detail to 3D prints. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, Denver, CO*, 2217–2223.

Wiklund-Engblom, A., Hartvik, J., Hiltunen, K., Johansson, M., & Porko-Hudd, M. (2015). Process documentation in sloyd: Pilot study of the ”talking tools” application. *International Journal of Interactive Mobile Technologies*, *9*(3), 11–17.

Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. Retrieved from https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf

Wing, J.M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A, 36*(1881), 3717–3725. Retrieved from https://www.cs.cmu.edu/~CompThink/papers/Wing08a.pdf

Wing, J.M. (2014). Computational thinking benefits society. Social Issues in Computing, January 10. Retrieved from http://socialissues.cs.toronto.edu/index.html%3Fp=279.html

Wood, N., Rust, C., & Horne, G. (2009). A tacit understanding: The designer's role in capturing and passing on the skilled knowledge of master craftsmen. *International Journal of Design, 3*(3), 65–78.

Yang, S. (2017). Knitting visualizer: Connecting craft and code. *Proceedings of the Conference on Interaction Design and Children*, *Stanford, CA*, 705–708.