ISSN:0975-9646

Computational Thinking: The Skill Set of the 21st Century

Mahsa Mohaghegh[#1], Michael McCauley[*2]

[#]*Department of Computing, Unitec Institute of Technology, Auckland, New Zealand*
[*]*Faculty of Engineering, The University of Auckland, Auckland, New Zealand*

Abstract- **Computational thinking is a relatively new term, and is the topic of much discussion in the educational realm today. Research shows that computational thinking is a highly valuable skill that is becoming a topic of increasing interest among computational education researchers, as well as computer scientists. The reason for this is due to the significant benefits associated with it in terms of problem solving.**

This review begins with an introduction to computational thinking as a term, and gives some characteristics surrounding the skill set. We then examine the benefits and advantages of computational thinking in general, and areas in which it may be applied. We also explore the importance of computational thinking in education and teaching practice, specifically at years 1 - 13, and note areas in which this is currently being implemented. We also search for the use of computational thinking in Kaupapa Māori, again noting current work in this area. Finally, we make recommendations for the inclusion of computational thinking as a core topic in primary and secondary education.

1. Introduction to Computational Thinking

Like many scientific terms and phrases, computational thinking as a term is reasonably well known, however a clear definition, what it entails, and areas in which it is applied are generally not so clear. However, according to Curzon, Black *et al.,* (2009), computational thinking is *the* skill of the 21st Century. As such, a growing number of people in academia are beginning to realise the importance of bringing computational thinking to the core of many areas of study such as business and commerce, biology, and biomedical engineering.

At such an early stage in its observed history, it is difficult to settle on a concise definition of computational thinking that is widely agreed on. However, it can be generically stated to the agreement of most that computational thinking is a collection of multiple problem-solving skills based on fundamental principles of computer science (Curzon, Black *et al.,* 2009).

Computer science itself is a somewhat misunderstood term – most probably because of the word "computer". Rather than being "the study of computers", computer science can be briefly described as using computers and computational technology to solve problems; the main focus is in problem solving. Jaokar (2013) draws some important points from such a description of computer science. Firstly, computer science shares many characteristics with mathematics, and therefore it is implied that it will also share problems and problem-solving techniques with other scientific domains. Secondly, computer science often deals with creating tools to solve problems, rather than just using the tools. It should follow naturally that other problems external to computer science as a discipline can be addressed using the same or similar techniques. These techniques can be defined or abstracted in an algorithm, a step-by-step instruction set.

Computer science – and more specifically computational thinking – is bringing about a fundamental change in every field of science. Computing technology is no longer just a tool to aid scientific research, but is becoming woven "into the very fabric of science". These fundamental developments in problem solving strategy pose significant implications and opportunities in a wide range of fields, and, as we argue further in this review, for primary and secondary school curriculum (Grover, 2013).

Computational thinking is a very broad term in its definition, with numerous and sometimes disagreeing definitions and descriptions of what the term entails. However most agree that Jeannette M. Wing, Head of Department of Computer Science at Carnegie Mellon University, is still one of the greatest pioneers of computational thinking. Wing's energies in this area in the mid-2000s attracted the attention of Microsoft, who in 2007 granted Carnegie Mellon University 1.5 million dollars to establish a research and study centre dedicated to this area (Curzon, Black *et al.,* 2009).

According to Wing (2006), computational thinking can be defined as a method or approach of "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (Wing, 2006, pp 33-35). Wing also describes computational thinking as a type of analytical thinking, or computing concept-driven approach to solving problems, modelling situations, or designing and implementing systems. Computational thinking can be envisioned as a thought process by which problems are represented in such a way that their solutions may be evaluated using information processing techniques. Solving a computational problem involves logical and algorithmic thinking approaches. The key skill is in logically breaking down a problem and systematically devising an algorithm suitable for solving it (Grover, 2013). Computational thinking and problem solving strategy enables those who implement it to model problems and situations that may yield a computational solution. Instead of separating problems and their solutions, computational thinking promotes problem decomposition, and the use of logic, algorithms and often innovation to solve them. It is a combination of logical, arithmetic, efficiency, scientific and innovative thinking, together with qualities such as creativity and intuition (Curzon, Black *et al.,* 2009). Computational thinking involves skills or techniques which often include decomposition of a task or problem, pattern recognition and abstraction, and formulating algorithms to solve this and similar problems or situations (Exploring Computational Thinking, 2015)..

Phillips (2007) takes an interesting perspective while addressing secondary school teachers regarding computational thinking, and defines what computational thinking does *not* entail. Phillips' points help eliminate misconceptions mistaking computational thinking for "thinking like a computer", involving only programming, or thinking that a physical computer is also central to the term.

Despite the numerous differing opinions on what is encompassed by the term, most agree on the significance of computational thinking as a skill set that has an important place in the 21st century, and must be explored in greater depth across a range of levels.

2.1 COMPUTATIONAL THINKING IN DETAIL

Computational thinking can be broken down into different facets of thought, each with its own particular strengths and applications. The following sections describe computational thinking aspects in more detail.

2.1.1 Logical thinking

Logical thinking is perhaps the most important part of computational thinking. Logic in this sense may be confused with a computer's logical calculation, however in terms of computational thinking it refers to a deduction or extrapolation of new information or data based on existing information. According to Curzon, Black *et al.,* (2009) **,** the logical aspect is in forming realistic conclusions, not reaching correct assumptions by chance. One prime example of logical thinking in action is the Sudoku puzzle game: answers to each cell in the grid must be correctly deduced based on "existing information" in the completed cells. A process of elimination can be utilised to determine correct choices.

2.1.2 Algorithmic Thinking

Algorithms play a major part in problem solving in computer science, especially in repetitive problems. This aspect of computational thinking is perhaps the most closely aligned to computer science itself. Algorithmic thinking can also be thought of as strategic thinking, or step-by-step processing. Algorithmic thinking in general problem solving can greatly improve efficiency, especially when dealing with problems of similar nature (Curzon, Black *et al.,* 2009)**.**

2.1.3 Efficiency

In computer science, and in particular in algorithm design, efficiency deals with the minimization of resources required by an algorithm to solve a problem. Although many computational resources may be defined, two are of significant importance: the *time* it takes for an algorithm to solve a problem, and the *memory space* required while solving. Of these two however, time required is usually the most important. This means that specific thought must go into designing an algorithm to best handle a specific type of problem: it is not possible to simply "speed up" an algorithm at runtime to improve its time complexity (Goodrich & Tamassia, 2002). In terms of algorithm design, an "efficient algorithm" is one which takes the least number of steps to solve a problem. A good example of efficient problem solving is the Rubik's cube (Curzon, Black *et al.,* 2009) **.** While it is true that making faster moves and spending less time calculating the best next move will result in a solution being reached in less time, it is more beneficial to determine the fewest possible steps needed from any given starting point. It is this kind of efficiency which plays a major part in computational thinking.

2.1.4 Innovative Thinking

Innovation is a key characteristic of computational thinking, and is best evidenced in the fact that computer science lies at the forefront of modern innovation. According to the Cable News Network (CNN, 2014), the top ten inventions are currently all results of amazing innovations in computer science. Innovative thinking trains the mind to question things which already exist, to challenge assumptions, and ultimately to think "outside the box" (Curzon, Black *et al.,* 2009). This aspect gives computational thinkers a significant advantage in problem solving.

2.2 OBJECTIVES

We seek to address computational thinking in an educational context, and explore the benefits that exist in this space, highlighting areas in which they could or should be utilized. While many of these areas encompass all educational levels in general, we focus specifically at years 1-13. We also search for evidence of computational thinking within Kaupapa Maori.

3 BENEFITS OF COMPUTATIONAL THINKING

Computational science has taken a stand with theory and experimentation as a pillar of science Phillips (2007). This is due to the power with which simulations and models of various phenomena may be created using computational thinking and application, enabling computer science to drive huge advances to multiple fields of science. Physicists have been able to computationally simulate supernovas, and cell shapes and accelerator cavity to learn more about the big bang (Southern Methodist University, 2010); computational biologists, have used computational resources to model the functions of sub-cell molecular motors. Geologists and environmental engineers use computers to model potential flow of groundwater-contained contaminants. Geotechnical and structural engineers computationally model earthquakes to test simulated structures and ensure their designs are structurally sound. Upon deeper inspection, it is actually the computational thought processes, the skills and mindsets that thinking computationally brings about, that are the real cause to these advances.

Although computational thinking is a core concept behind most areas of computer science, it has attracted significant attention in recent years as a skill that should be discovered and exercised in a much greater way than it is. Wing (2006) published an article introducing computational thinking as a term, and began to discuss the importance of this approach as an attitude and a skill set that deserves much more attention**.** Wing states that computational thinking is increasingly influencing many other disciplines since it possesses many characteristics which are beneficial and even necessary in these fields.

Computational thinking has been boldly labelled as *the* 21st Century literacy, since it allows non-computer scientists to benefit from a computational approach to problem-solving (Cuny *et al.,* 2010). Computational thinking helps us to understand problems and sub-problems that are computable, helps thinkers to determine the correct tools and methods for solving certain problems, as well as helping the exploration of method limitations. Almost all disciplines have now been influenced by computational thinking in some way, in both the sciences and humanities. There are numerous examples of this influence in many fields – machine learning has influenced the implementation of probabilistic graphical models in statistics, greatly improving pattern recognition for extended data sets (Machine Learning Department, 2008); in biology, computational thinking has advanced human genome sequencing (Fisher & Henzinger, 2007); even in fields such as economics, computational thinking has had influence: online auctioning, ad placement, and banking.

In 2011, the Institute for the Future (Institute for the Future, 2011) published a document overlooking future work skills that will be necessary by 2020. The report highlights six “drivers of change”: longevity, smart machines and systems, a computational world, new media ecology, superstructure organisations, and a globally connected world. The report also highlights ten key skills that are of paramount importance **(Figure 1)**. On this list is computational thinking, which IFTF describe as the skill to conceptualise and draw abstractions from data sets, and model problems computationally.

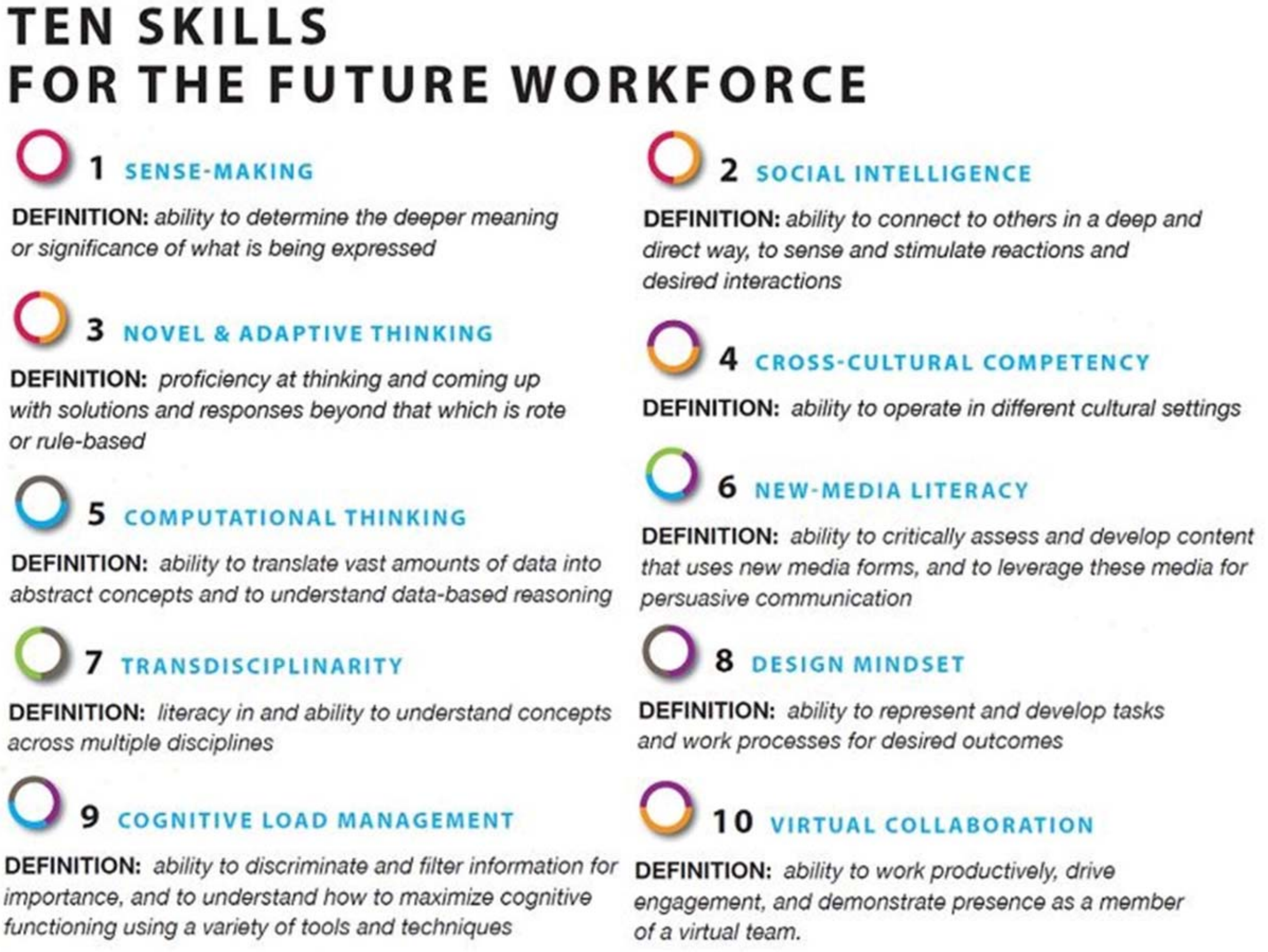

Figure 1 - Ten Skills for the future

Jaokar (2013) speculates even further than the immediate future, and suggests that computational thinking may evolve to the extent of being able to be used to address even more complex problems and situations, to ultimately drive our level of innovation higher than ever before.

4 Computational Thinking in Year 1-13 Education

The importance of computational thinking as a skill to be taught in schools has been recognised be a number of high profile individuals. Wing (2006) goes beyond the tertiary sector to state that computational thinking should be learned by everyone and used not just by those at university-level academic fields. She promotes computational thinking as a vital skill for today and the future, equating its importance to that of reading, writing, and basic arithmetic.

The drive for more attention on computational thinking is gaining momentum, with many universities' computer science departments modifying their curriculum to focus more on fundamental computer science concepts and principles rather than having programming their main area of focus**.** Dr. Wing's efforts have also carried even further, and there is now significant work being done to bring these fundamentals to primary and secondary schools (albeit at a suitable level of complexity). The importance of this move is evident to many, including international ICT giants such as Google and Microsoft. In 2007, Microsoft Research awarded Carnegie Mellon University a grant of several million dollars to establish a research centre for computational thinking, and continues with its support. In 2006, Google initiated its CS4HS (Computer Science for High Schools) workshop at Carnegie Mellon University, a two-day workshop open to high school teachers to show them emerging technology, and new and energetic methods to introduce computer science and computational thinking at primary and secondary level. Several years later in 2010, CS4HS had spread to a total of 34 tertiary institutes worldwide, and is currently one of the largest single workshops aimed at bringing computational principles into schools (Blum & Cortina, 2007). Google also launched a website as an extension of its Google for Education programme in 2010, aimed at promoting computational thinking and providing numerous links to web-based resources at both tertiary and secondary levels (Google, 2015).

One programme originating in New Zealand is the work by Bell, Fellows and Witten in creating Computer Science Unplugged, a website and programme aimed at teaching K-12 computer science principles without computer (Bell *et al.,* 2009)**.**

Computational terms may be more effectively understood if students are able to see them effectively demonstrated in areas they are already familiar with. Teachers need to be constantly evaluating areas in which they could demonstrate the use of computational terminology and analogy. Students need to see computer science as more than just programming, but instead an immensely broad field and the initiation of a branch of thinking that may be used to solve many problems in numerous areas. In this instance, mathematics can be seen as a tool to be used in computational representation and problem solving.

There are a number of benefits of computational thinking that can be seen from a secondary school classroom perspective. Computational thinking has the potential to equip students with more than just "technology literacy", or a working knowledge of how to use computers for everyday tasks. It enables students to be more effective problem solvers for situations beyond the computer science realm, and encourages them to create tools to solve problems, rather than use existing tools (Phillips, 2007). Computational thinking is a skill that needs to be developed in the next generation. According to Hunt (2012), a clear understanding of what computational thinking is, together with how and where it can be implemented is of paramount importance in preparing our next generation for a world filled with technology and constant technological advance.

Computational thinking involves a number of core principles from computer science, such as abstraction and algorithm design, decomposition, pattern matching, generalization, and inference. However computational thinking is a skill of significant benefit to multiple disciplines, and is not just limited to computer science and technological fields (Hunt, 2012). The reason for this is that it helps students define what can and can't be solved, and prompts them to research computational models for situations that are traditionally unrelated to computer science. Even if a student chooses a career other than computing, the skills learned and developed through computational thinking will benefit them in whatever field they eventuate.

Grover (2013) identifies a significant advance in elementary school computer science learning with programming languages such as *Scratch*, MIT's *App Inventor*, *Kodu* and *Alice*, which enable students to construct working programmes and apps in most cases very quickly. While this is advantageous in learning programming skills, the degree of conceptual knowledge obtained through using these tools can be questioned, specifically to what extent students learn computational thinking skills. Grover (2013) draws from first-hand experience in teaching high school students various computational principles and exercises in Scratch, robotics and App Inventor, and identifies the need to concentrate on *how* to construct solutions, and *why* some solutions are more suitable and effective than others, rather than simply learning the coding syntax of a particular language. Grover encourages going beyond "the tools": programming languages, syntax 'quick solution' methods. Instead she promotes the need to use young students' learning ability and creativity to develop key computational thinking skills. A deeper understanding of computational problem solving is more valuable than exploring the surface of tools in this area without realising their full potential.

Computer science professionals and teaching professionals at all levels have a responsibility to begin driving computational thinking to various degrees across all disciplines. However, incorporating computational thinking into non-computer science curriculum requires careful planning in its approach, and despite the fact that it is in education that the drive for computational thinking is most needed, both computer science researchers and educators need to work together towards this goal. Computational thinking springs from the principles of computer science, but the methods of incorporating computational thinking into general education will be different than those used in computer science. Guzdial (2008) proposes several questions that must be answered to determine the best approach. It is important to correctly determine the level of understanding non-computer science students have of computing principles, and what areas pose significant challenges to students.

To help pave the way for computational thinking in the classroom, it is necessary to revisit and improve the way computing is taught. This fact is well supported by Guzdial (2008), and Barr & Stephenson (2011). Two areas of research are of particular importance to this goal: human-computer interaction, and computing education. Human-computer interaction involves exploring new methods with which humans interact with computers, while computing education research involves investigating how humans reach an understanding of computer science principles. Of these two however, computing education is arguably of the greatest importance. According to Guzdial, "research in computing education will pave the way to make computational thinking a 21st century literacy that we can share across the campus" (Guzdial, 2008, p27). It is also necessary to make changes to current curriculum and pedagogy in order to construct an atmosphere that is more conducive to exposing classrooms to emerging technology. Tertiary-level education is no longer the right place to introduce computational concepts, even though it is at this level that most students make choices for a future career. The reason for this is the simple fact that the next generation of students, regardless of what career path they choose, are already in an ever-increasing technology-driven world, filled with computers and computer-led solutions. Logical problem solving, an understanding of how to manipulate algorithmic problem solving and abstraction, together with other computational thinking skills are now required at both primary and secondary school levels.

There is significant complexity in introducing computational thinking to primary and secondary school curriculum. The vision and belief in the necessity for the drive must be shared by a several groups. Educators must be made aware of the necessity, and computing education researchers must communicate with computer science researchers to explore optimum ways to incorporate it into curriculum. The whole change requires a paradigm shift in education, enhanced teacher engagement, together with research-led resource development and collaborative work with educators in computer science. This shift and advance in education necessitates two main directions of effort: educational policy, and resources (Barr and Stephenson, 2011). The biggest challenge in educational policy is perhaps the fact that computational thinking as a term in still relatively new, and there is still some discussion on what the concept actually entails. This can present difficulties in infrastructure change, and in presenting a comprehensive strategy for change, which in turn challenges the most appropriate direction to take in preparing suitable resources.

The importance of computational thinking is becoming more apparent to tertiary institutes, and as a result they are beginning to incorporate it into their curriculum. Wing (2008) extrapolates from this the necessity of enhancing the focus of computational thinking at primary and secondary school levels, and challenges computer science and education communities worldwide to investigate effective methods of bringing sound computational thinking to school students.

Wing (2008) proposes several fundamental characteristics and principles of computational thinking which as she puts, are either extremely helpful or of particular necessity in the classroom. At undergraduate tertiary level, courses are continually being developed which bring focus to core principles of computing, and Wing suggests attention of this nature should be given to primary and secondary level education. One principle she suggests is parallel and sequential processing, which can be explained very effectively to younger students by analogy and real-world examples. Another principle is that of the parity bit, where an extra 'check' bit is added to the end of a binary string, indicating whether the number of bits with a value of one is even or odd (simple error-checking code).

Wing (2008) also warns against letting the "tool" (i.e., the computer, or the programming language, etc) hinder a solid understanding of the concept. Just as teaching a child arithmetic operations only by calculator will hamper their understanding of basic mathematic skills, so teaching a student to code, create programs and form solutions without teaching them the correct steps in breaking down a problem, forming abstractions and designing algorithms will hinder their understanding of the core principles. This seems extreme, but the situation of students learning to code and assuming (or worse yet, being told) that they understand the concepts of computer science is frighteningly prevalent. What is necessary is an effective integration of the "tool" *with* the concepts. Cuny *et al.*. (2010) also impress the importance of the algorithm (and solution) design process, rather than the solution itself, and while coding and programming language syntax is important, conceptualisation is more essential, as it can be transferred to other areas of study.

This generation's familiarity with technology helps form a strong background for understanding key underlying computational concepts. Technology's fast-paced advance can often be overwhelming, and can cause concerns particularly for parents. American musician Bill Laswell once stated *"People are afraid of things they don't understand. They don't know how to relate. It threatens their security, their existence, their career, their image"*. However, computer science is becoming such an underlying part of our culture, and an understanding and knowledge of not just how technology works, but the fundamental computational steps to problem solving can help dispel these fears.

United States educator and technology advisor Brian Puerling suggests that the fear of technology felt by parents and teachers must be overcome, since our current generation of children can be viewed as 'technology natives'. Children as young as 18 months are already manipulating tablets and smartphones with ease. He advises that technology must be used in conjunction with traditional learning, rather than replacing it altogether. Puerling states that the fact that today's children are immersed in technology is prime evidence that computational concepts are a necessary aspect of early education (Jones, 2013). This generation's familiarity with technology helps form a strong background for understanding key underlying computational concepts.

A drive towards the inclusion of computational thinking in standard education curriculum is particularly important for people groups who are not traditionally heavily represented in computer science and technology, since these groups can benefit vastly from being exposed to the methods, thought processes and problem solving strategies that are employed in computational thinking. Regardless of their representation, these are technology consumers all the same, and to the same degree as others. On top of this, a reworking of curriculum content to include focused attention on computational thinking can be coupled with a drive for enhanced focus on development of these groups also. A prime example of this is in Te Ao Māori in New Zealand.

5 Computational Thinking in Kaupapa Māori

Traditionally, computer science has not been as popular an area for Māori students as others such as literature, design and arts (Hook, 2008). In terms of computational thinking, there is currently no major work involving, or of significant influence to Māori. However, the need to promote technology and computer science to Māori has been identified by a number of groups.

TangataWhenua.com, a Māori ICT company founded in 2002, is a current pioneer of digital technology originally formed to counter a negative impression from mainstream Māori representations, and is now a leading independent Māori news and information portal. This company was one of the main facilitators of the Google Māori Project (Tahana, 2008), an initiative aimed at enabling Māori users to use a Māori-translated Google Search Interface, completed in 2008. TangataWhenua.com recognises the necessity for Māori to be trained as developers and creators of technology and digital solutions, rather than just users and consumers of existing technology. In a bid to initiate this effort, the company is currently developing Digital Natives Academy (DNA), a "real life space" to provide tamariki, rangatahi, pakeke and kaumatua support and resources to inspire them to experiment with and create and develop their own digital tools (Digital Natives Academy, 2015). DNA's chief objective is to give children the facility to equip themselves in an age driven by technological advance. To accomplish this, DNA is planning to hold after-school classes and workshops, holiday programmes and courses aimed at teaching computer programming fundamentals to whanau with the use of real-world teaching aids. DNA hopes to promote creative and computational thinking, systematic and logical reasoning, and collaborative work with the use of online and mobile technologies, and ultimately contribute to preparing the next generation for a digitally-rich future.

One significant learning programme that has been developed and implemented on trial is the Te Ika Unahi Nui in Okato, Taranaki (Ministry of Education, 2015). One main goal of the programme is to introduce and promote digital emerging technologies in the classroom in order to improve and strengthen learning and literacy. The programmed was trialled with Coastal Taranaki School students, with reports of significant success. A great improvement in students' connection to their language, culture and heritage was noted, and the development of leadership skills, and ability to approach and solve problems, but students also found a connection to the relevance of technology, and how principles in technology could be used to enhance their learning (Ruakere, 2015).

Another group aimed at improving digital learning among Māori is the *Learning with Digital Technologies for Māori and Pasifika Learners* group (White, 2015). This is essentially a forum of approximately 400 members, and provides a networking platform for teachers and educators to connect with each other to share and discuss strategies, approaches and resources in order to improve the engagement of Māori and Pasifika learners.

Clubhouse 274 is a community-formed initiative sponsored by the Tindall Foundation (Tindall Foundation, 2013). It was the first clubhouse under the Intel Computer Clubhouse Network, an international network of groups started by MIT Media Labs and the Museum of Science in Boston Massachusetts (Intel Computer Clubhouse Network, 2015). Clubhouse 274's goal is to open digital opportunities to young students in Otara, and provides tutorials and workshops in areas such as digital media production, photography, animation and 3D modelling.

While there is evidence of early steps being taken to introduce computational thinking and problem solving skills in Kaupapa Māori, there is still much room for improvement. Certain groups are taking great steps in the development in this area – such as White (2015) – however this needs to be scaled to a greater degree, with a greater awareness of the value of computational thinking and problem solving.

6 Conclusions

We are currently standing on the brink of a new era of learning. As technology advances, and computers and computational solutions are involved more and more in our everyday lives, all levels of education must at some point take a turn to frame young minds to prepare for an increasingly digital world. Computational thinking has been named by a number of computer science and educational authorities as *the* literacy of the 21st century, and something which is being addressed at tertiary level with high degrees of success. As shown in this review however, there is still some work to be done at the primary and secondary school level to reap the enormous benefits of students' development of computational thinking skills. This transformation is not an easy one, as it involves significant changes to pedagogy and practice, and necessitates the collaboration and co-operation from every party involved, from researchers to educators.

While there is still discussion over the exact bounds of computational thinking as a term, there is significant agreement on the benefits of computational thinking as a key skill in all aspects of our society, from doctors to engineers, managers to researchers – a workforce trained in computational problem solving spells efficiency, economic benefit, and even further advances to technology. However, these reasons only highlight the need to provide students from a young age with the core principles of this valuable skill.

In terms of computational thinking in Kaupapa Māori, initiatives such as TangataWhenua.com's Digital Natives Academy, and Learning with Digital Technologies for Māori and Pasifika Learners group are taking practical, active steps to facilitate and increase in focus on digital technologies within these spaces. While this is excellent progress forward, there could still be more attention given to computational thinking and problem solving as stand-alone skills (with or without a digital-focused environment). Development in this space could well be coupled with the drive for the inclusion of computational thinking skills in New Zealand primary and secondary curriculum.

REFERENCES

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.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology, 13*(1), 20-29.

Blum, L., & Cortina, T. J. (2007). *CS4HS: an outreach program for high school CS teachers.* Paper presented at the ACM SIGCSE Bulletin.

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking.* Paper presented at the Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada.

Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing, 1*(2), 67-69.

The CNN 10: Inventions. (2014). from http://edition.cnn.com/interactive/2013/11/tech/cnn10-inventions/

Computer science: A curriculum for schools. (2012). Retrieved from Computing at School Working Group website: http://www.computingatschool.org.uk/

Cuny, J., Snyder, L., & Wing, J. M. (2010). Demystifying computational thinking for non-computer scientists. *Unpublished manuscript in progress, referenced in* http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Curzon, P., Black, J., Meagher, L. R., & McOwan, P. (2009). cs4fn. org: Enthusing students about Computer Science. *Proceedings of Informatics Education Europe IV*, 73-80.

Digital Maori Forum (2015). from http://www.dmf.co.nz/

Digital Natives Academy. (2015). from http://digitalmaori.com/about/

Exploring Computational Thinking. (2015). from http://www.google.com/edu/programs/exploring-computational-thinking/

Fisher, J., & Henzinger, T. A. (2007). Executable cell biology. *Nature biotechnology, 25*(11), 1239-1249.

Foundation, T. T. (2013). Computer Clubhouse. Retrieved from The Tindall Foundation website: http://tindall.org.nz/computer-clubhouse/

Goodrich, M. T., & Tamassia, R. (2002). Algorithm Design, Foundations, Analysis and Internet Examples: John Wiley and Sons, Inc, USA. ISBN: 0-471-38365-1.

Grover, S. (2013). OPINION: Learning to Code Isn't Enough, from https://www.edsurge.com/n/2013-05-28-opinion-learning-to-code-isn-t-enough

Guzdial, M. (2008). Education Paving the way for computational thinking. *Communications of the ACM, 51*(8), 25-27.

Hook, G. R. (2008). Māori students in science: Hope for the future. MAI Review, 1(11).

Hunt, S. (2012). Computational Thinking. Retrieved from Computing At School website: http://www.computingatschool.org.uk/index.php?id=computational-thinking

Intel Computer Clubhouse Network. (2015). from http://www.computerclubhouse.org/

Jaokar, A. (2013). Evolving the definition of Computational thinking, from http://www.opengardensblog.futuretext.com/archives/2013/07/evolving-the-definition-of-computational-thinking.html

Jones, N. (2013). Hi-tech lessons for 'digital natives. Retrieved from The New Zealand Herald website: http://www.nzherald.co.nz/nz/news/article.cfm?c_id=1&objectid=10886368

Machine Learning Department 2008 Research projects. Retrieved from Carnegie Mellon University website: http://www.ml.cmu.edu/research/index.html

Phillips, P. (2007). Computational thinking: A problem-solving tool for every classroom. *Center For Computational Thinking*.

The Re-working of "Work". (2011). Retrieved from Institute for the Future website: http://www.iftf.org/our-work/global-landscape/work/future-work-skills-2020/

Ruakere, J. (2015). Te Ika Unahi Nui. Retrieved from http://teikaunahinui.blogspot.co.nz/p/blog-page.html

Southern Methodist University. (2010). *New mathematical model aids Big Bang supercomputer research.* Retrieved from http://www.sciencedaily.com/releases/2010/01/100105170938.htm

Tahana, Y. (2008). Google Maori helps te reo go places. Retrieved from The New Zealand Herald website: http://www.nzherald.co.nz/nz/news/article.cfm?c_id=1&objectid=10521805

Te ika unahi nui wānanga – A marae-based learning programme. (2015). Retrieved from Ministry of Education website: http://elearning.tki.org.nz/Beyond-the-classroom/Learning-beyond-the-classroom/A-marae-based-learning-programme

Togyer, J. (2011). Research Notebook: Computational Thinking--What and Why? *Magazine: The Carnegie Mellon University School Of Computer Science*.

White, A. (2015). Learning with Digital Technologies for Māori and Pasifika Learners, from http://www.vln.school.nz/groups/profile/721864/learning-with-digital-technologies-for-maori-and-pasifika-learners

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717-3725.