USITAA Submission to ACARA

Executive Summary

- We strongly support every Australian student being taught a general purpose programming language in Years 7-8
- We strongly support every Australian student being taught a programming language in years 5-6 (including visual programming languages)
- We strongly support the teaching of algorithmic and computational thinking in early primary years
- We strongly recommend removing project management from the syllabus completely, and instead including syllabus points on entrepreneurial thinking.

We believe that the draft National Curriculum: Digital Technologies is an excellent step in the right direction for the future of Australian education. NCSS and USITAA believe that world class technology education for all Australian students should be a national imperative.

Aligning a National Curriculum to the needs of an increasingly competitive global digital economy will empower our children to with skills needed to contribute towards Australia's future growth. Teaching the next generation fundamental programming/coding and entrepreneurial skills is vital for Australia's future economy post mining-boom.

NCSS and USITAA believe it is essential that every Australian student be introduced to computer programming and computational thinking in primary and lower secondary school. We recognise that the implementation of this curriculum will require substantial support from industry and tertiary education providers, and resolve to collaborate with educational bodies to support the implementation of the curriculum, teacher training and student engagement in digital technologies.

Discussion focused around three separations of the draft curriculum:

1. the Rationale, Aims, Organisation and Key Concepts
2. the Strands; each row in the Scope and Sequence table
3. the Year Groups; each column in the Scope and Sequence table

More detailed notes from discussions follow.

Discussion of Rationale, Aims, Organisation & Key Concepts etc.

Overall, we found these sections adequately expressed the aims of the curriculum, though no rationale as to why Digital Technologies is grouped with 'Design and Technology' 'Technologies'. We feel that there is little in common between the two subjects, and that each deserves to stand on its own. While we do see many opportunities for cross-curricular activities, this is no different between Digital Technologies and literally any other subject studied at school, including maths, science, English, foreign languages, etc.

Key Concepts

Like:

- Emphasis on "appreciating the transformative power of digital technologies" in the section on Interactions and impact. Understanding this transformative power is critical to ensure students realise the potential of the skills they are learning.
- We like the focus on ethical and social responsibilities of technology, and feel that this will allow for many interesting discussions, that will serve to further encourage student engagement.

Change:

- We consider that the concept of **controlling** real-world interactions should be expanded on, including looking at aspects of embedded systems, algorithms "in the wild" (e.g. "HFT stock prices, Amazon sellers) and feedback.
- Projecting digital systems into the real world of physical things and people.
- Data collection first sentence needs reworking
- We strongly recommend dropping project management from the curriculum. It's not a key concept and it is not generally an interesting aspect of IT. It certainly is not an appropriate topic to engage and stimulate interest in technology. Technical project management skills are best learnt on the job, and are of little to no use for students writing small, independent projects, not large projects of hundreds of thousands of lines of code, with many contributors. If Project Management must be discussed, perhaps it would be best to investigate the concept of how Open Source works, including learning about Source Control systems.

Add:

- More focus on innovation and using ICT to improve the world should be included. It is this that will inspire students.
- Include Real world questions about ethics, e.g. privacy with unmanned drones. Who is in control: humans or algorithms? We strongly believe that there are incredibly engaging discussions to be had that students can relate to the real world.
- In the section on Abstraction, the concept of universality should be added. Specifically, computer science is not simply a special case of electrical engineering.
- In the section on Interactions and impact, we recommend including concepts of safety and privacy in a digital world, as well as a different topic covering security, privacy and integrity. Note the distinction between identifying a 'safe' online environment with which to interact and identifying potential security issues.
- Further discussion could occur on the topic of what is true on the internet, and Open Access to information.

- In the section on Interactions and impact, we recommend explicitly including historical impacts of technology, and modern entrepreneurship.
- We strongly recommend adding in a section of investigation of history, specifically how the world operated without computers. This reflection will be increasingly important in the future.
- Some introduction to algorithmic complexity should be included, for instance looking for a name in a phone book, looking up books in a library etc.
- Concept of universality of computation and the plasticity of programming to solve all sorts of problems / domains should be developed.

Regarding the section on Data Collection:

We recommend that the following be added:

- Data storage that remains for the very long term when people may not realise it has been kept at all or its implications
- The notion of control systems and the real world

Regarding the section on Specification, Algorithms and Implementation

We recommend that the following be added:

- We note that there is no discussion of the concept of Algorithms with unpredictable behaviour, for instance due to intelligence / randomness etc. and believe that this could be beneficial to add in.

Strands

The following are notes from the discussion of each Strand.

Representation of Data

- The elaborations were found to be very helpful in trying to understand what is covered by this strand, but some elaborations remained unclear.
- We recommend adding in the concept of the fact that a bitstream can be interpreted in different ways. Perhaps an interesting discussion would be illegal numbers and illegal primes: http://en.wikipedia.org/wiki/Illegal_numbers, http://en.wikipedia.org/wiki/Illegal_primes)

Digital Systems

- The content on services and layers of abstraction was found to be good.
- Some of the language of the elaborations was problematic and unclear.
- We found that the concept of abstraction should be developed. Understanding the concept of layering is more important than understanding FTP, for instance.

Interactions and Impact

Things that we particularly like:

- We like the explicit mention of 'sharing' in 2.2 elaboration 1
- We also particularly like the idea of encouraging students to envision new applications

We recommend that the following be changed:

- Historical aspect for F-2 2.2(5)
- Concepts may date quickly. E.g. in passwords may soon be replaced by biometric data
- In 4.2, "imagining" is nebulous. Would "describing" fit better?
- Is 4.2 too hard for the age group?
- Less ergonomics. This seems to be a general capability, rather than Digital Technology.

We recommend that the following be added:

- More playing, exploring and interaction. Less 'noticing'.
- "Recreation" should include games
- Notion of composition of systems has to be introduced, perhaps through something as simple as Lego.

Managing and Analysing Data

Things that we particularly like:

- We particularly like the focus on understand why data is collected.
- We like the phrase 'Play with' in junior years as being an example of data manipulation

We recommend that the following be changed:

- We have concerns that this content is simply unachievable in the time allocated, and recommend that Digital Technologies be allocated more time.

- We feel that there needs to be clearer distinction between an abstract data model and a concrete data model.

We recommend that the following be added:

- The topic of security as an aspect of data
- Discussion of the concept that deletion of data is not always true deletion, but often 'deactivation', and what that means for privacy etc.
- We believe that there is scope here for discussion on the topic of jurisdiction of storage, and the legal and social implications this has.
- Local vs Cloud storage (Facebook, MegaUpload etc)

Using Digital Systems

Things that we particularly like:

- The distinction of "Identify" in 2.4, and the fact that this is not as trivial as it may seem. Identifying that digital systems are embedded in potentially unexpected places, from cars to watches.
- Constructing a model in 2.4's expansion

We recommend that the following be changed:

- 4.4's perhaps too operational focus. Less 'how' and 'why'.

We recommend that the following be added:

- Perhaps something else that extends beyond years 3-4

Specification, Algorithms and implementation

Things that we particularly like:

- algorithms progression, measured and logical
- Scope and sequence generally board; not reliant on a current technology.
- Problem solving as a general focus
- Starting with English then going to digital

We recommend that the following be changed:

- The specific reference to "Agile" will date. Whilst we appreciate that this is certainly better than staying with the already dated 'Waterfall method', and the potential problems with simply referring to 'current practices', it's not clear that specifically calling out the somewhat faddish term 'agile' is better.
- More scope for creativity, for instance fractals, designing the 'AI' for an animal.
- We would like to see a bigger focus on learning through experience.
- "Digital Solutions" sounds dry and boring.

We recommend that the following be added:

- Creative exploration.
- Need for "recursion" (in form of reduction to a simpler case?)

Creating and Interacting online

Things that we particularly like:

- Focus on safety, especially in the early stages
- Spread of applications covered, i.e. images, audio etc
- Continuation of ideas through the grades

- Has enough freedom and scope to allow teachers to be creative
- Broad language allowing for new innovations / technologies
- The later years have been covered well

We recommend that the following be changed:

- We strongly recommend removing project management from the curriculum.
- It isn't quite clear how teachers would be able to implement the provision of a 'safe online environment'. Perhaps this is something the NSW Board of Studies or other similar organisations could help provide?
- Some of these points could have some specific examples to teach to ensure a minimum set of practical knowledge.
- Hard to know whether project management fits in here.
- 2.6 - unrealistic to assume that kids will have safe access to sites with known people. Realistically will be unknown from the beginning.
- Teen years - ethical considerations not addressed.

We recommend that the following be added:

- We would like more detail about how teachers should implement this (or at least some guidelines), since we are quite unsure ourselves. The elaboration suggests intranets etc. Do the teachers have to set these up themselves? Could they?
- Need to add context of what the internet is, which doesn't actually seem to be covered anywhere in the curriculum. This should be added in early.
- Online 'stranger danger' training could be added from foundation on.

Year Groups

F-2

Like:

- We like introducing fundamental concepts to kids in fun, creative and playful ways
- 2.3: Types of data; Searching and organisation
- 2.5: Algorithms
- The "Game" and "play" focus, especially in the second half of 2.4, with the creative/experimental play
- Cool stuff to play with

Change:

- Clarity needed about "safety" (i.e. what it means)
- In 2.1: "Learning about how changing a colour photo to black and white changes the file size"? We aren't certain that this should be included here at all.
- In 2.1, the "generalisations about the data sets" seems a little abstract
- 2.2 and 2.6 are quite similar
- 2.3 and 2.4 also overlap

Add:

- Emphasise how computers follow instructions, even blindly. Lots of scope for kids to have fun learning that.

3-4

Like:

- Opportunities to practice
- 4.4 experience many different things, choose different things for different purposes
- 4.5 like the idea of following descriptions
- Especially like 4.4
- Especially like the points on privacy in social media (4.2)

Change:

- 4.3 don't like spreadsheets and databases as specific examples
- 4.4 memory cards? Seems muddled
- Would like 4.4 to be merged into others.
- 4.6 flowcharts? Looks good until the details/elaborations are inspected
- The examples date quickly
- Not sure what the "Asian language" point means

Add:

- 4.3 - connection between family tree and data structure
- 4.4 - good opportunity to explore special needs, e.g. vision impaired
- 4.5 explain how hard it is to formalise and explain an algorithm unambiguously
- 4.7 what are "agreed social protocols"?

5-6

Like:

- Emphasis in 6.7 on visual programs
- 6.8's second elaboration, especially ethics

Change:

- 6.2: last two bullet points seem 'off'. How does augmented reality?
- 6.3 / 6.4 Perhaps too abstract

Add:

- Decent tools for guidance for visualisation
- Some sort of toolkit
- Add in discussion about how people did things before computers and digital networks?
- Add in 'how do you do task X without a computer?'

7-8

Like:

- If students fully understand all of the points given here, they will be on a very good footing.
- The guidelines really begin to look at internal representations and explain *how* computers work
- Takes away the magic, in a good way.
- Good sample of real things → amazing base for higher education and improved skills

Change:

- The fact that 8.9 (teaching programming) is a very big point, which would require more than 1/11th of teaching time
- 8.10 just feels vague and unnecessary.
- Not clear it shows relevance to everyday life

Add:

- More focus on abstraction and conceptualisation skills. I.e. teaching about image and audio representations is good, but ensuring that students would be able to generate new representations when faced with new problems is important.
- "This is the big picture" → Know that whilst many problems are solvable with computers, some problems are very hard, and are not solvable with computers.
- Repetitive tasks are inefficient; students should learn to identify tasks that are repetitive, and realise that they can be automated for improved efficiency.

9-10

Like:

- Algorithms section of 10.8
- 10.2
- 10.8 - testing testing testing!
- 10.11 - legal, ethical sections, but not the management part

Change:

- We don't like all this "client" rubbish / project management.
- 10.7 "stakeholders". Dislike this. What relevance does "with clients" have to high school students?
- 10.3 In elaborations: power surges? Planned obsolescence? Capcha? This all seems off, and not of interest to students at all.
- 10.5 elaboration: spreadsheets are not a programming langauge. Remove references to specific technologies

- 10.6 and 10.7 Concrete Vs abstract data models. The model of the problem is not suitable as a model of the solution.
- 10.10 "Agile" is a loaded term, and won't be relevant soon.
- 10.10 in elaboration: "Feedback from client". We shouldn't be teaching students to become consultants. We should be equipping them with useful skills, and demonstrating the interesting, important impact that computer science has on the world.
- 10.1 Elaboration includes XOR, which seems too specific for security.

Add:

- Software is NOT about consulting for "a client". This is simply the wrong mindset.
- Where is 'building great products and services for the world?
- Needs more of a focus on entrepreneurship
- The language used needs to change totally to reflect a more relevant focus on innovation, entrepreneurship and solving important problems.
- Students should be able to develop and write an algorithm
- 10.4 you don't get it right the first time, intuition is not a good guide for data structures of security.
- The concept of efficiency.
- More historical aspects.