Are we missing something in our courses?

A STACS perspective on CS BGE, NQ and NPAs

1. Summary

STACS' work with the SQA course reports and upskilling suggests that there are pieces missing from the computing science course specifications, to the detriment of our learners. These missing pieces are hard to identify because of the lack of a big picture of the subject. The NQs, BGE and NPAs are analysed to help understand how this has come about. Suggestions towards a big picture are given – once defined, the missing pieces can then be identified.

2. Missing pieces and a big picture for computing science

Over the years, CS teachers and examiners repeatedly find that learners fail to learn specific parts of our courses. An example is the writing of evaluation statements about solutions. SQA NQ course reports repeatedly highlight these failings, and STACS has used those reports as a starting point for targeted upskilling designed to address the issue and raise attainment.

Whilst the STACS approach has been well-received, we are realising that the issue does not lie *only* with the troublesome parts of the course specifications identified in the course reports. Instead, we suspect there are parts of the subject that are *not in* the course specifications at all, but that constitute essential connective tissue for effective learning. For example, it may be hard for pupils to write effective user and functional requirements because the scenarios are contrived and over-specified already, and there is no experience of having a real client nor any experience of working in teams. We may look for how to better teach what *is* specified, as with STACS upskilling, but at times, we should be looking for what is *missing* from the specifications that would help us achieve the larger goals of a well-rounded computing education – a *big picture* of the subject. These *missing pieces* should then be added into the course specifications. Note that this need not require significant new material to be assessed, but rather, for example, an emphasis on progressive development of skills and making connections across the topics, from BGE right through to AH.

This pilot review of Computing Science is an excellent opportunity to remind ourselves what the big picture should be and to search for and incorporate the missing pieces.

3. An analogy – a jig-saw puzzle with missing pieces and no big picture

Our challenge is akin to a puzzler with a jig-saw puzzle that has missing pieces, and worse, the puzzler doesn't have the big picture on the front of the box. *Without* the big picture, they just have a pile of pieces which are very difficult to connect up. W*ith* the big picture on the front of the box, the puzzler can work out the right positioning of the jig-saw pieces they have. This leaves them with clear gaps into which the missing pieces would fit. And, again with the big picture, the puzzler can then identify the detail on the missing pieces, and craft replacement pieces.

Note the central importance of the big picture and how it enables us to repair the jig-saw. As any puzzler knows, of course, the big picture and the pieces are connected. Every piece has its place and contributes to the whole. That sounds like an excellent way to view all the pieces we teach our pupils – all contributing to a whole that we both support and understand.

As we work on through our situation with the Scottish BGE and NQs, we will refer back to the jig-saw with its missing pieces.

4. The BGE and NQs as a pile of jig-saw pieces with no box picture

That is a contentious section heading, but bear with us. Many of us have, at different times, been involved in contributing to the computing science offering, and we have all acted in good faith. But we assert that the end result is still, remarkably and somewhat disappointingly, a pile of jig-saw pieces with no big picture! How did this happen?

4.1 The lack of a shared understanding, or a Body of Knowledge, for computing science

As noted, we assert that we don't have a big picture, or body of knowledge (BoK), for computing science. To clarify this point, we first suggest that long-established subjects have an implicit body of knowledge known to all practitioners. For such subjects, the practitioners most likely first experienced the content and teaching methods when they themselves studied them at school. Any gaps were filled in during higher level study of the subject later in school or in university, and then there was a sound subject-specific teacher education also. There are also most likely plentiful textbooks. Without needing to spell it out explicitly, teachers know what the subject entails. This implicit knowledge is common to the whole teacher community, thanks to their backgrounds, and is fundamental to effective collaboration and developing good practice.

For computing science, none of the above hold. For some current CS teachers, there was no computing in their own schooling; others would have had a patchy secondary education, not matching to what is now covered since the 2014 refresh of the NQs and the 2017 refresh of the BGE. CS teachers have a wide range of prior qualifications in computing-related areas, sometimes not at degree level, and their prior experience may or may not be a good fit for the particular languages and systems used across the curriculum. In-service teachers have noted limited computing science specific preparation during their teacher education for teaching the subject; this is hardly surprising given that sound progression frameworks and pedagogical content knowledge for primary and secondary school CS education are still being determined. Computing textbooks often focus on particular languages or systems, or on the individual aspects of a course, rather than painting a more holistic picture that focusses on underlying principles and concepts, developed over time. All of this means there is no implicit body of knowledge shared across the computing science teacher body, damaging effective collaboration and the development of good practice.

4.2 The National Qualifications content specifications have become a kind of BoK

Given this general lack of a *shared and overarching* understanding of the subject, the content grids in NQ course specifications for computing science end up being a *de facto* definition of the subject – that is, they are taken as the body of knowledge that is to be taught. But an NQ course specification is really an assessment specification, and these are not the same as a body of knowledge. An assessment specification is a relatively brief summary, using concise language, the meaning of which is assumed to be shared across all practitioners; a body of knowledge is a full definition and explanation of language, concepts, skills as well as an overarching, holistic, understanding that ties these all together. As noted above, an assessment specification is ok for other subjects where the subject's body of knowledge is shared by practitioners, but that is not the case for computing science. We do not have that shared body of knowledge across all teachers. Furthermore, computing science teachers are often in one-person or small departments, and therefore have no opportunity to discuss content with colleagues, nor have a specialist PT to guide or direct the teaching.

The NQ specifications do include higher level goals – a big picture – but these are not explicitly linked to the various topics in the content list. This is *not* then like the jig-saw puzzle, since it is not clear how the big picture and the pieces are related. Moreover, a computing teacher, typically time-poor, will usually jump past the higher-level stuff to the content list to determine what they must cover. The focus becomes the individual topics rather than the development of a bigger picture understanding, because this is how things are laid out in the specification. The subject becomes siloed, and the opportunity is lost to deepen learning of concepts common across topics. There *are* shared concepts and skills that would unify knowledge, but these are not appreciated.

The appendices for the NQ course specifications do attempt to capture aspects of a BoK – but they are more geared towards how ideas and material will be presented in an exam setting. They are not there to exemplify a learning approach, and they are not necessarily complete – for example, they may have a focus on aspects that were known to be new to many practitioners at the time of the introduction of the course.

An additional issue with the absence of a body of knowledge concerns past papers. These become a ground truth for what should be taught. They provide an exemplification of the brief content lists – implicit or explicit in a question is a broader definition of what these items mean, or how they will be interpreted by markers, which is the same thing for a teacher who wants their pupils to do well. In this way, the subject being taught is being defined by the process of setting questions, rather than via a larger discussion across the subject. This is an unfair pressure on setters, but also unfair to the subject itself.

4.3 Articulation between the BGE and the NQs

The BGE and its relationship with the NQs is also an issue. The BGE structure was an attempt to capture the concept of computational thinking in a practical way befitting the Technologies area of CfE within which it sits. While small scale studies at the primary level have shown that the structure is easy to understand for primary teachers, there has not been a concerted effort to explain the structuring to all teachers, primary and secondary. Once again, as with the NQs, we have a collection of pieces that are described in such a concise fashion that they are open to interpretation, given the lack of professional development on the big picture of the curriculum design.

Furthermore, the variability in levels of provision during primary and secondary BGE, in addition to a perceived necessity to make secondary BGE appealing to draw pupils into the NQs, adds to very differing levels of attainment among pupils in different schools on the way into the senior phase

The NQs are therefore constrained in some ways because of this variability in the learner's knowledge levels developed during the BGE. An expected level of attainment is neither tightly specified, as noted above, nor implicitly appreciated and acted on by the members of the teaching profession, as it would be in other subjects. Can the NQs truly build on the BGE, or are they effectively designed to start from scratch?

A more holistic developmental understanding of the subject, a necessary part of a big picture, is required so that knowledge developed in primary can be assumed at the start of secondary, and following on from that, the secondary BGE knowledge can be assumed in the NQs. This would be a huge accelerator to deeper understanding and more success in the subject.

Towards addressing this issue, we welcome an aspect of the CIC review process that involves taking the subject or course specification right through to the end of the NQs, rather than stopping at the end of the BGE phase, as they do now. As such, the NQ documents can become a true "assessment specification" rather than a "course specification", as they are currently referred to.

4.4 NPAs alongside NQ

While National Progression Awards (NPA) are developed as vocational awards for delivery in the college sector, many schools now include CS-related NPA courses at levels 4 to 6. With a catalogue of almost 20 titles, learners might study diverse areas such as Games Development, Cyber Security, Data Science or Web Design. On one hand there is wider choice and vocational pathways leading to college and beyond; on the other, these courses are also described from an assessment perspective and by their nature are specialised in their content. Whether schools position NPAs as an alternative or a complement to NQ in CS, the course description in terms of outcomes and performance criteria are elements of assessment, again like jig-saw pieces without a clear big picture. Freedom and flexibility in presentation of these courses may be appealing, however, as with NQ courses, CS teachers don't have a pre-existing shared BoK or big picture, we simply have the NPA specification.

4.5 Summarising

In summary, we propose that the NQ course specifications have become a kind of "body of knowledge" (BoK) for computing science teachers in Scotland, when in fact they are an assessment specification designed to give brief summaries of examinable content whose fuller definition is assumed. As an assessment specification, the focus is more on detailing items than on overarching concepts and principles, more on a pile of pieces than on the big picture. For a different reason, the BGE is somewhat similar, lacking a commonly understood big picture, and leaving the Es and Os of the curriculum definition open to interpretation. And the NPAs are primarily focussed on particular application areas with little recognition of how they might contribute to broader and more long-lasting skills. Again, one can view them as collections of pieces. This pile of pieces approach, alongside other logistical issues in CS provision, leads to a less-than-ideal developmental progression for learners as they progress from early BGE through to the senior phase.

In writing this, and as noted above, this situation is nobody's fault. Computing science is a peculiar subject compared to other school subjects, stemming primarily from its (lack of) longevity in school curricula and all the consequences that brings. What works acceptably and even effectively for other subjects hasn't worked well for computing science.

And a final, but important, point. The vast majority of the pieces in our various piles are good pieces. Perhaps the odd piece from another puzzle has crept in – but overall, our pieces do all concern the kind of computing science knowledge that we want to foster in our learners, particular in the BGE and NQs. But the lack of the picture on the box means that we are not tying all that learning together as effectively as we could be, and we are not appreciating the few missing pieces. Adding those missing pieces and tying it all together more effectively could potentially create a really formidable offering

5. Matching big pictures & missing pieces to the pilot review structure

For the CIC (Curriculum Improvement Cycle) process, of which this pilot review is a part, Education Scotland is following the "Double Diamond" approach to design. This separates the identification and specification of the problem (discover, define) from the solution to the problem (design, deliver). And the first day of the pilot review is very much the problem discovery/definition phase.

The discussion in this document has proposed two issues / problems. The first is that we need to develop an agreed big picture for computing science; the second is that we need to identify the missing pieces in our curriculum. We have argued that the first is needed in order to accomplish the second. We're not exactly sure what these two stages mean for the double diamond, but we include below some thoughts on the big picture for our subject.

6. Pointers towards a big picture for computing science

We include below three aspects that contribute towards a possible big picture for the subject: comments made in the Scottish Tech Ecosystem Review; the contribution of computing science to the four capacities of CfE; and the underpinning framework of the current computing science BGE.

6.1The Scottish Tech Ecosystem Review (STER)

The STER, commissioned by Scottish Government, and the recommendations of which have now been adopted, gives an indication of a big picture. The STER notes that the larger the pool of engaged young people coming out of school with "raw engineering skills", the better for the ecosystem. Currently, however, many of those who are interested in starting a business involving tech struggle because of their lack of prototyping skills. That is, they cannot take their own idea and bring it to life computationally, even in a mocked-up form. One of the reasons for this lack of development, reported anecdotally, is that many children are put off CS by the National 5 course. "Put simply, the curriculum is boring", notes the STER, continuing "For a subject that is inherently magical (the ability to build almost anything) this is disappointing, and something is wrong."

This idea of "building almost anything" fits well with computing science's place in the Technologies area of Curriculum for Excellence. Just as woodwork and needlework are concerned with developing skills necessary for making things ultimately of our own imagination and design, so should CS be also.

We know of course that in both National 5 and Higher, the originating ideas for development tasks are decided upon by others, such as the coursework designers and exam setters. Hence, learners are not required to go through the messy process of taking their idea and matching it to what they know computationally – the process of model-building that is at the heart of developing computational artefacts. As such the processes of analysis, design and evaluation, although present in name, are not effectively developed in learners.

Hence, the STER contributes to a possible big picture – the ability to take personal ideas arising from any context and turn them into computational artefacts.

6.2 The CfE four capacities

At the start of the document, we referred to the idea of a well-rounded CS person. The Curriculum for Excellence captures the more general idea of a well-rounded person via the four capacities. To justify its inclusion in the curriculum, it is important for a big picture of CS to connect to the four capacities. As a starter to considering this:

- **Successful learners**. A successful computing science learner will develop skills to apply their computing knowledge to new contexts and problems. Learning about a problem and its context is an aspect of computing at any level – asking the right questions, investigating and so on. While underlying principles stay largely the same, technologies change, and a successful computing person will need to be able to adapt and learn new technologies repeatedly.
- **Confident individuals**. Being able to generate one's own problem / task, and solve it, is a huge confidence boost, developing agency in the learner.
- **Responsible citizens**. Successful developers need to think about their clients, ensuring that their requirements are properly understood and then satisfied. The Apps For Good movement is an example of an approach that encourages learners to think responsibly, and this opens to door to considering issues such as climate and sustainability.
- **Effective contributors**. Software developers nearly always work in teams, as do most in the world of work, at some level. There are ways this could be brought effectively into the computing science curriculum, and this would be a huge benefit both to subject knowledge and skills and to the learners' wider education.

6.3 The structure of the BGE

The three organisers of the BGE for computing science are designed to capture key aspects involved in the process of taking a problem or task through to a solution, and what must be learned to enable that to happen – to some extent, a big picture and how to get there educationally.

- The first organiser introduces computational concepts through observation and analysis of real-world contexts.
- The second organiser introduces computational contexts, such as programming languages or database systems, showing how the computational concepts are represented there.
- The third organiser addresses the task of solving problems emerging in real-world contexts by using the tools of the computational contexts.

The third organiser captures the skill of computational thinking – but this depends on the other two organisers. Solving problems using computers, the focus of the third organiser, is in essence a modelling process, and can only be achieved if the real-world context and problem can be understood in computational terms (the first organiser) and the computational contexts are well-understood too (the second organiser). Considering the issues raised in the STER, pupils are likely not encountering enough real-world contexts and problems in their studies around the first and third organisers.