Teaching and Learning Music through the Lens of Computational Thinking

Eddy K. M. Chong
National Institute of Education
Nanyang Technological University
Singapore, Singapore
eddy.chong@nie.edu.sg

Abstract—In recent years, many countries around the world have intensified their efforts to introduce computational thinking in schools. This paper supports this educational move not by proposing the teaching of computational thinking as a subject per se but by taking inspiration from the nature of such systematic and logical thinking to reframe the teaching and learning of music. After clarifying what computational thinking involves, it presents some parallels between computational thinking and musical thinking, principally, decomposition, pattern recognition, abstraction, and algorithmic thinking. From a teaching perspective, this comparative juxtaposition elucidates how computational thinking can provide a fresh lens on musical thinking, and in turn inform and transform the teaching and learning of music. In fact, this proposed pedagogical move can potentially broaden the disciplinary purview when design thinking is brought into the picture, yet without sacrificing disciplinary grounding in the subject itself.

Keywords—computational thinking, musical thinking, music teaching, music learning, music harmony.

I. INTRODUCTION

Modern-day conception of computational thinking (CT), as it had developed in relation to the discipline of Computer Science in the mid-twentieth century, has had some seven decades of development now. Remarkably, Seymour Papert, who introduced the term in 1980 [14], had foreseen the day when children would “grow up surrounded by computers and a computational culture”. Anticipating this diffusion of computational concepts, which he saw as likely to accelerate even more in common between CT and thinking in the arts than meets the eye. In fact, I will further argue that explicitly adopting a CT frame of mind can benefit both the teaching and learning of the arts. The broader underlying position here is the recognition that CT has already been changing the way we think as well as the way research is done in both science and humanities disciplines [1]. Neither the arts as a discipline nor the world of the arts that our students are entering are immune from the broader societal transformations in this regard. From a subject-specific standpoint, I would contend that there is much to be gained from the adoption of CT by arts and arts education educators. Let us consider this using music teaching and learning as illustrations.

The rest of this paper is organized as follow: Section II define what is computational thinking. Section III describes CT and thinking in music. Section IV presents the music teaching through the lens of CT. Finally Section V concludes this work.

II. WHAT IS COMPUTATIONAL THINKING?

We begin with a clarification of what computational thinking is and how it relates to a number of related terms such as coding and programming. Carnegie Mellon University (CMU) has often been quoted for their description:

Computational thinking is a way of solving problems, designing systems, and understanding human behavior that draws on concepts fundamental to computer science. [21]

Fundamentally, CT is a way of thinking. It undergirds computer programming—which is the scripting of instructions for the computer to perform a particular task—but it is more than that because it entails other processes such
as those mentioned in the definition quoted above. Coding is a form of programming but the two terms have often been equated even though, for some, there are technical and even connotative differences [26]. For the present purpose, we shall put aside the subtle nuances and henceforth use the term coding, not least because in a number of recent advocacy literature in educational contexts, this presumably less intimidating term was used. With these terms of reference, we will underscore the understanding that "Computational thinking focuses more on the thinking process rather than just coding" [22].

To further clarify, CT also does not apply to all areas in the broader academic discipline of Computer Science. As alluded to in the CMU description above, the latter discipline has both the technical aspects dealing with the hardware and software of computing as well as, on the application front, broader human, societal considerations. In an education-focused taskforce report for the Association for Computing Machinery (ACM), Computer Science was defined as "the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society" [31]. In this regard, Digital Promise has helpfully clarified the relation between computational thinking, computer science, and coding thus (see in Figure 1):

![Fig. 1. The relationship between computational thinking, computer science and coding](image)

One notes that there are aspects of CT that lie outside the disciplinary domain of Computer Science. At the same time, while coding is definitely a sub-domain of Computer Science, CT also only intersects with coding rather than entirely subsumes it.

What then does CT entail? There have been various articulations in recent years for different purposes or contexts. One recent construal relates it to two core practices in Computer Science [29]:

- recognizing and defining computational problems
- developing and using abstraction

For the purpose of incorporating CT into K-12 curriculum, the Computer Science Teachers Association (CSTA) in America and the International Society for Technology in Education (ISTE) identified nine core computational thinking concepts and capabilities in 2011: data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation parallelization and simulation [7].

Specifically as a problem-solving thought process, CT has been distilled into four most fundamental elements [18, 24]:

- decomposition – simplifying a complex problem into smaller, more manageable problems, each to be solved separately
- pattern recognition – identifying commonalities among the smaller, decomposed problems and also within individual problems to help solve the more complex problem more efficiently
- abstraction – identifying the salient information and filtering out details that are not needed, then using the different levels of abstraction to understand and frame the problem
- algorithmic thinking – crafting a series of rules or steps to solve the problem

These have been referred to as "the four cornerstones of CT" [18].

Admittedly, there have been other configurations of CT too. The Computing At School (CAS) community of teachers, academics and industry supporters, for example, identifies six CT processes for solving problems, adding two to the above four [20]:

- logical reasoning – predicting and analysing
- evaluation – making judgements

We may, however, subsume them under one or more of the above four fundamental processes.

III. CT AND THINKING IN MUSIC

A. Abstraction in Music

Like Computer Science, which encompasses a number of sub- or associated disciplinary domains, Music as a subject also has sub-disciplinary divisions. Broadly speaking, music studies can be classified underperforming, composing, analysis, or other academic music studies. The disciplinary thinking involved in each of these sub-domains necessarily differs from one another to varying extent. Nonetheless, there are aspects which parallel those operative in Computer Science. We shall use music analysis to illustrate some parallels with CT, not least because many of the other sub-domains involve music analysis—performers analyze music to make performative decisions, composers analyze other composers’ music to learn from them even as they analyze their own while composing, musicological and ethnomusicological studies often entail some degree of music analysis too.

Perhaps a simple way to help one begin to see the parallel between CT and thinking in music is by taking a leaf from Henderson and his associates’ presentation of CT [8]. In their curricular mind map, they position counting as a first step towards CT. Arithmetic follows next, leading to more abstract levels of symbol-based thinking like algebra. In music, music note reading may be considered the counterpart to counting in that discerning the scale degree of the notes—whether mentally reading the notes or singing them aloud in solfège—is like counting. The different ways of putting
together musical notes is then likened to the manipulation of numbers in arithmetic. When musical symbols are introduced to succinctly reference particular types of musical configurations with their attendant musical concepts, the musical thinking operates on a more abstract level not dissimilar to that in the world of mathematical symbols.

Thinking about musical harmony in terms of chord symbols, as opposed to the aural experience of the harmony itself or the physicality of playing the harmony on a piano, is a good case in point. This immediately involves abstraction where each chord symbol represents particular sets of notes. When one construes chunks of harmonic progressions in terms of harmonic expansion units using symbols like T, PD and D [12], the musical abstraction is taken to a higher level. Finally, when musicians speak of large-scale harmonic structure at the level of a whole piece of music, the music chord symbols used may be similar to those at the chord level, but the musical conception involved is at a far more complex and higher abstract level. At each of these levels, there are underpinnings that govern the use of the symbols. Figure 2 presents these levels with a particular example provided for each.

For a trained musician, the above abstraction and the attendant reasoning involving the various sets of symbols can be part and parcel of their musical thinking, whether as a performer, composer or even just a music listener. In other words, abstraction and symbol-based abstract thinking or reasoning are not foreign to musicians.

B. Problem Solving in Music

Since problem solving is also central to CT, let us illustrate more extensively the affinity between CT and musical thinking with a problem solving scenario. Take for example the question (read: “problem” in music) “what are the features of a particular harmonic style.” To tackle this “problem,” we would need to first identify the relevant pieces of music to examine (“data collection”) and determine how the music will be represented for this purpose (“data representation”). Given the complexity of music harmony, the task will need to be broken down into smaller tasks (“problem decomposition”) whereby particular aspects of the harmony can be examined, such as types and range of chords, common chord successions, typical harmonic elaborations, harmonic rhythms, and so forth. For each of these processes, the mode of representation (“Abstraction”) of what is heard or what is conveyed by the musical score will need to be decided. Conventionally, if this were conventional classical harmony, we may use chord symbols like Roman Numerals or possibly Guitar Chord Symbols or, still using music stave notation, harmonic reduction or the likes of Schenkerian voice-leading graphs. If, however, the tonal approach is not suitable, alternative representations will need to be identified, if not created. Whatever the case, each of these representations embodies different kinds of musical abstractions.

In converting the aural or visual raw data into a particular music-symbolic representation, some form of music analysis is involved. At this first stage, “pattern recognition” in the CT sense of “data analysis” is called for. We shall use classical tonal harmony to illustrate. First, we need to recognize, whether aurally or visually, the various chord renditions—different “patterns” of notes—that belong to the same family represented by a particular chord symbol; this pattern recognition is already involving some abstraction. Then the second level of “data analysis” work with this first-level “data”, now represented by the symbols used. We may, for instance, collate common instances of “patterns”, represented in those symbols, to reveal certain typical harmonic features on the basis of frequency of occurrences. In the process, observations of variants or interesting details—whether during the groundwork analysis stage or the subsequent pattern recognition stage—can prompt additional abstractions. For example, once the common occurrence of a particular suspension has been noted, the different types of suspensions can be symbolically represented and subject to further pattern recognition. In terms of outcomes, instead of a finding like “the dominant chord is one of the most commonly used chords”, we may have a more exacting one like “the dominant chord is one of the most commonly used chords, and it is often used with a 4-3 suspension”.

Where then is the algorithmic thinking? There are actually more than a few sets of algorithmic procedures embedded in the various processes described above. This becomes immediately clear when we think of getting a computer to do the above work. From the conversion of the raw musical input into some form of representation to the collation of salient features, very specific sets of instructions (“algorithm”) must be crafted, and these can be very complex indeed. Just to illustrate, imagine getting the computer to identify a simple dominant chord by parsing these two musical scores:
advances in social science, education and humanities research, volume 327

the computer will need to recognize and interpret all the notes that are sounding simultaneously in the first chord in each case. at the individual chord level, it needs to first recognize the root of the chord and the bass note, amongst other things. to do this, a set of algorithms is needed so that the computer “knows” how to convert the first vertical set of notes into some form of abstraction and generate an output statement like “a major chord with g as the root” for figure 3a and “a major chord with the chordal fifth in the bass” for figure 3b. next, on a more global level, the computer needs to know that g is the dominant of the key in both cases. determining the key will entail creating another set of algorithms, before yet another set of algorithms enables the computer to recognize that g in both cases is the dominant.

admittedly, even after completing the above series of algorithmic procedures, we have only achieved a rather rudimentary level of harmonic analysis, far from the more sophisticated kinds of answers desired for our simple question above. nonetheless, the above suffices to show the step-by-step process involved, and the underlying musical thinking that we as music teachers sometimes take for granted as being very simple and obvious to our students.

C. CT in Music Evaluation Task

Let us now turn to a musical “problem” of a different kind. imagine teaching a computer to evaluate the grammatical correctness of harmonic progressions. consider, for example, the progression I-Vb-I-ivb-V-Ib-V-bVI. to evaluate, the computer must again be given a set of rules or steps to follow. again, the complex nature of this evaluation task calls for “decomposition”. as a first step, we may get the computer to evaluate pairs of chords systematically by examining overlapping pairs of chords, that is, proceeding with the first two chords, then the second and third, and continuing in this manner. for each chord pair, we then break down the “problem” further into a series of mini-problems or questions:

- is the chordal-root movement appropriate?
- is the bass movement melodically appropriate?
- are there chordal dissonances in the first chord?
- if there are dissonances, are they resolved appropriately?

each of these will have its set of possible scenarios, each to be handled accordingly. for example, root-movement considerations will need to further factor in the specific chord inversion for the two chords in question, which may entail various “if…then” programming statements. if the algorithm is properly formulated, I-Vb-I-ivb-V will be evaluated as grammatically sound after the four overlapping pairs of chords in I-Vb-I-ivb-V have been successively evaluated.

the appearance of chord ivb, however, calls for an additional set of algorithms because of it being a mixture chord. a larger chord-progression context needs to be taken into account insofar as the computer must be “instructed” to check for and register any occurrence of a mixture chord as it proceeds. to do this, the mode of the key must first be noted. once a mixture chord has been detected, the computer needs to identify any subsequent diatonic chord which has a mixture counterpart but the latter is not used. it is only with such systematic checking that chord Ib will be singled out as inappropriately appearing after an earlier mixture chord ivb.

in the present example, we have foregrounded “decomposition” and “algorithmic thinking”. this is of course not to deny that the illustrative mini-questions above do stem from various “abstractions” and “pattern recognition” for the purpose of evaluating the harmonic progression. nor are we overlooking the role of human understanding and deliberation in deciding on the use of such notions as chordal-root movement, bass movement and chordal dissonances to frame or formulate the music-computational problem for the computer to handle. to be sure, the human thinking here in part necessarily stems from music-theoretical understanding, which embodies music-aesthetic considerations. we can thus see the pertinence of and the role such technical understanding and artistic sense play in shaping the machine CT. in sum, what we see is a “partnership” between human and machine CT whereby the latter cannot entirely replace the former.

IV. MUSIC TEACHING THROUGH THE LENS OF CT

A. Benefits

The above re-sketching of the various music analytical processes to illustrate some parallels between the musical thinking and CT has allowed us to see afresh some of the music thinking musicians are familiar with when making sense of music. to a certain extent, it demystifies musical thinking: behind “musical sense” are various logical steps. invaluably, from an educational standpoint, it has made the musical thinking visible. in fact, it goes beyond thinking aloud, as it were. in having to “instruct” a computer to perform those tasks, not only do we need to articulate the thinking steps involved, we are also forced to be very logical. the computer can only proceed in the way it has been
programmed to. If the algorithm is illogical in certain parts or faulty in any way, the programming outcome will show that up. Programmers are critically aware of this.

Granted that there are areas in musical thinking where computational work has its limitations, a CT lens can nonetheless enable a music teacher to better prepare the teaching. The “decomposition” phase alerts the teacher to the often multiple cognitive (sub-)processes involved, so that the learning task may be broken down into smaller ones if necessary. Such an understanding can in turn inform the assessment and feedback phases in teaching whereby the teacher can diagnose students’ errors and learning difficulties in relation to particular sub-processes, and follow up with better targeted remediation. The algorithmic steps themselves can be used to better instruct students, without of course forgetting any necessary complementing music-aesthetic considerations. In sum, the process of formulating and articulating the algorithms or heuristics give teachers insights into the particular musical thinking activity. If music teaching were approached in such a CT manner, it could share the benefits of AI education insofar as disciplinary knowledge is made “computationally precise and explicit” [2]. It is as if the “black box” of musical thinking is opened up, and once put on the table it can be more easily used to design teaching and learning.

On a more radical front, CT can fundamentally change the way we think. In adopting a CT mindset, the computational metaphors involved can raise new kinds of disciplinary questions, produce new hypotheses and theories, and even expand disciplinary thinking [1]. Just as with any disciplinary topic where there are often alternative and sometimes competing theories or perspectives and understandings, CT and its metaphors can potentially give rise to new theories and disciplinary thinking paradigms. In an age where the right mindset and capacity to deal with multiple perspectives and to think creatively are prized as two important twenty-first century competencies, introducing CT into music teaching and learning can contribute to the fostering of these valued educational outcomes. Moreover, the CT approach here can also go a long way in bridging between the arts and the sciences.

B. Concerns?

What then is the difference between teaching a computer and teaching a student to engage in musical thinking, at least for the musical tasks discussed? We have thus far focused much on the computational—or the “computation-able”—aspects of musical thinking. What is clear is that disciplinary knowledge and reasoning is important. For the computer, that knowledge comes in the form of symbols and instructions and the reasoning is embedded in the programming. For the human learner, the teacher in a similar way needs to ensure the learning of the knowledge and reasoning involved. But, as pointed out above, music learning also involves aesthetic judgement and other relevant considerations; this we have noted is one limitation of CT. Therefore, in injecting CT into music teaching, we do not necessarily limit or skew the learning as long as we are mindful of the limitations or even pitfalls of CT and knowingly fill in the gaps. What we have been interested in here is harnessing the positive affordances of CT for music teaching and learning. The next section will not only assure us that adopting CT will not compromise music teaching and learning, it will in fact suggest that CT can offer us a basis to deal with the multi-facet nature of musical thinking.

Beyond CT

It is remarkable that in defending Computer Science as a science, Peter Denning, a past president of the ACM, also accords a place for the arts within the discipline insofar as he sees design as part of the subject [2]. Most recently, and more pertinent to our cause here, he has argued that ultimately it is not CT that we should be interested in but computational design [3]. He shared how a team of computational fluid dynamics scientists and PhD computer scientists essentially failed to achieve deep-level collaboration because of the mutual lack of understanding of the other’s expertise. The computer scientists were not able to express the problem and solution methods appropriately and the physicists ended up treated them as no more than just programmers. One may conclude that what Denning witnessed therefore highlights the importance of CT incorporating the relevant disciplinary thinking; or better yet, that it points to the ideal of the physicists being able to think like computational scientist and the latter thinking like physicists.

But this does not suffice. He goes one step further to propose the idea of computational design, which involves not just CT for the purpose of programming, but is in fact—and necessarily—a broader skillset that “includes the dimensions of listening to the community of users, testing prototypes to see how users react, and making technology offers that take care of user concern” [3] Figure 4 illustrates how computational design lies at the intersection between CT, relevant disciplinary thinking and design thinking.

While Denning does not refer to particular models of design thinking, we know that the prime characteristic of design thinking is its human-centredness as seen in its emphasis on empathy in the problem-solving process. Once again, the limitation of CT is clear: even when infused with disciplinary thinking. Denning alerts us to the limitation of CT in solving real world problems by citing “wicked problems” that, for example, call for social actions as a
critical part of the solution. In such cases, it is empathy on a larger social level and it may in fact involve socio-psychology, amongst other things.

What then is the implication for music educators vis-à-vis Denning’s proposition. If the earlier comparison between CT and music analytical thinking has suggested ways for us to give music teaching and learning a twenty-first century "makeover", then Denning has broadened our educational agenda to go beyond our traditional disciplinary boundaries. At the same time, given his affirmation of the critical role disciplinary thinking plays in CT, we need not feel apologetic when giving due emphasis on the musical aspects of CT or even when modifying the CT in our context. Next, taking our subject inquiry into the real world, if the CT applied involves testing some kind of prototype, whether we are solving real-world problems involving music or more primarily musical “problems”, the “user considerations” should include musical ones as appropriate.

Now, if, Denning’s story has underscored the importance of disciplinary knowledge, the same applies equally to the domain of teacher education when CT is embraced. Speaking of preparing teachers, Yadav and his colleagues have made clear that deep understanding of the subject discipline as well as teacher knowledge of learners and curriculum goals are essential when attempting to incorporate CT into their teaching. Or, as they put differently later in the article, teachers’ computational thinking knowledge must be supported by pedagogical content knowledge (PCK) and technological pedagogical content knowledge (TPACK) [7]. That is to say, the nexus of knowledge represented in Figure 3 must be further augmented with the intersecting teacher knowledge bases represented in Mishra and Koehler’s TPACK model [5].

V. CONCLUDING REMARKS

We close by returning to the astute observations of Papert. Lest we are still hesitant or even skeptical of the need to embrace the computational culture that has now become part-and-parcel of our daily lives, we should pay heed to the implications he has spelt out for us in no less direct terms more than three decades ago:

The computer is happening; whether educators accept it or not. Their choice is not one of deciding that it is good and should happen or bad and should not happen. Their real choice is either to recognize the trend and try to influence it or to look the other way until it has happened without their input. [10]

Apropos the present paper, I would propose that we jump on the CT bandwagon, not so much because we have no choice, but because it offers us exciting opportunities to breathe new life into the teaching and learning of our subjects. Furthermore, more than being just in-sync with the times ourselves as educators, we are also preparing our students for a world where computational thinking is fast permeating every discipline and profession. As Grover and Pea rightly recognized, “those in possession of computational competencies will be better positioned to take advantage of a world with ubiquitous computing” [4].

REFERENCES

Journal Papers


Conference Proceedings Papers


Books


Book Chapters


Online


