Check for updates

Informatics in Schools: Why, What, Who – and How to Initiate Change?

Neil C. C. Brown
King's College London
London, UK
neil.c.c.brown@kcl.ac.uk

Michael E. Caspersen
It-vest & Aarhus University
Aarhus, Denmark
mec@it-vest.dk

Michael Kölling
King's College London
London, UK
michael.kolling@kcl.ac.uk

Abstract

Informatics in schools is in a different situation in each country, whether barely present or fully integrated into a national curriculum. In this position paper we discuss the reasons ("why") behind teaching informatics to all students ("who"), the shape that this should take ("what"), and offer case studies of two countries that have implemented it in differing ways (and to differing extents): England and Denmark, including "how" they initiated this change. We find that although England has been successful at introducing informatics at all ages, it is very programming focused. Denmark offers a more rounded curriculum suitable for all students, but progress has been slow at implementing it nationwide. Important issues of equity remain unsolved and overlooked, especially around special educational needs and disabilities.

CCS Concepts

• **Social and professional topics → K-12 education**.

Keywords

Informatics, School, Curriculum, England, Denmark

ACM Reference Format:

Neil C. C. Brown, Michael E. Caspersen, and Michael Kölling. 2025. Informatics in Schools: Why, What, Who – and How to Initiate Change?. In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2025), June 27-July 2, 2025, Nijmegen, Netherlands.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3724363.3729076

1 Introduction

Over the last ten to fifteen years the importance of "coding" has been increasingly emphasised, by politicians and in the general press. Headlines have proliferated urging schools to teach "coding" to all pupils, and world leaders have weighed in to emphasise its importance. This trend signals a shift in the public perception of the importance of informatics, and in particular programming, and raises many interesting questions, including: *What* do non-experts and policy makers mean when they talk about "coding"? Are they right in asserting that *who* it is for includes all pupils? And if they are, *why* is it important? If the answers are convincing, then the next question is *how* to introduce informatics into schools. It is simple to pronounce that informatics should be taught in schools, but the actual process of carrying out such a change is difficult, very context-dependent and relies on public support, political will, and timing. This position paper discusses these questions using examples from the school informatics curricula (and the process of its introduction) from two nations: England and Denmark. We present considerations in the design of these curricula, formulate principles and guidelines, and discuss lessons learned.

First, we offer a brief note on terminology. There are many possible names for the school subject: computing, informatics, digital technology, etc – all with their own subtle connotations. In this paper we adopt the term *informatics*, in line with a recent multinational ACM report [10]. We will use individual countries' own terminology where appropriate (e.g. England calls its subject "Computing"). As described by Stephens et al. [38], "teacher training" often refers to a more operational English model of inducting teacher trainees into a specific subject, rather than a European model of "teacher education" which is more reflective and academic, but we will use "teacher training" to encompass both.

2 Case studies

The state of informatics in schools naturally varies by education system, differing between and sometimes within countries (such as in Germany [22] or the USA [18, 21]). A report from the European Commission presents the status quo of 39 European education systems for the school year 2020-2021 [14]. In this paper we will give case studies of two systems we are most familiar with – England and Denmark – to examine some general principles.

2.1 England

England has a slightly complicated educational system[1]. There is a National Curriculum for England, which is determined by the government and is mandatory for government-controlled schools. However, many English schools have now (somewhat controversially [36]) become "academies", which are allowed to deviate from the National Curriculum. In practice, many academies continue to follow it fairly closely [29]. We provide a few excerpts here from The National Curriculum for Computing in England[2], with England's "key stage" terminology translated to age groups for clarity:

Ages 5–7: "Pupils should be taught to: understand what algorithms are..., create and debug simple programs..., use technology safely and respectfully ..." There is a deliberate focus on programming from an early age. Typically this uses block-based languages, or unplugged activities. There is also mention of technology usage.

[1]England is part of the United Kingdom of Great Britain and Northern Ireland, but its education system is separated from those of Wales, Scotland and Northern Ireland.
[2]The full curriculum is public: https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study, retrieved December 2024.

ITiCSE 2025, Nijmegen, Netherlands

ACM ISBN 979-8-4007-1567-9/2025/06
https://doi.org/10.1145/3724363.3729076

Ages 7–11: "Pupils should be taught to: design, write and debug programs that accomplish specific goals..., use sequence, selection, and repetition in programs..., understand computer networks including the internet..., use search technologies effectively..., use technology safely, respectfully and responsibly." The curriculum is half programming, half technology usage at this age group.

Ages 11–14: "Pupils should be taught to: use two or more programming languages, at least one of which is textual..., understand simple Boolean logic [for example, AND, OR and NOT]..., understand how data of various types... can be represented and manipulated digitally..., undertake creative projects that involve selecting, using, and combining multiple applications..." The focus is on programming and computing principles more than technology. There is an explicit requirement that students must move to text-based programming (not just continue block-based).

Age 14+: England moves to elective subjects, so the national curriculum becomes largely irrelevant.

In summary, England's curriculum is primarily structured around programming. This was a deliberate decision at the time to distinguish it from the prior Information and Communication Technology (ICT) curriculum, which had an Information Technology (IT) application focus and was deeply unpopular [5, 6]. There still remains a substantial component (about a third to half) about technology usage. It does contain several mentions of underlying principles of informatics, such as data storage and networks. It is also a very short document: the complete curriculum fits on two A4 pages.

2.2 Denmark

The EU have launched a Digital Education Action Plan for 2021–2027 [15]. The EU has no direct role in setting education policy in its Member States – this will act as a recommendation. Different Member States are at different stages of implementing informatics education. Denmark's Ministry of Education established a task force in 2008 to revamp the informatics curriculum – this was made permanent (but not mandatory) for upper secondary education in 2015 [9]. A trial for primary and lower secondary education ran 2018–2021. The subsequent evaluation (in Danish [7]) found that: informatics was considered by all stakeholders to be an important and relevant subject; informatics is motivating and instructive, and the pupils (of all genders) were motivated and learned successfully; and formal teacher education is urgently needed – the trial managed without but this was not considered a sustainable model.

The proposed curriculum document for primary and lower secondary arising from this trial is over 50 pages long and is available online[3]. The four competence areas underlying this curriculum (as relayed by Caspersen [9] in English) are:

Digital empowerment refers to the critical and constructive exploration and analysis of how technology is imbued with values and intentions and how it shapes our lives individually, and as a society. It concerns the ethics of digital artifacts and promotes an analytical and critical approach to digital transformation.

Digital design and design processes refers to the ability to frame problems within a complex problem area and, through iterative processes, generate new ideas that can be transformed into form and content in interactive prototypes. It focuses on the processes through which digital artifacts are created and the choices that designers have to make in these processes, highlighting students' ability to work reflectively with complex problems.

Computational thinking and modeling concerns the ability to translate a framed problem into a possible computational solution. It focuses on students' ability to analyze, model, and structure data and data processes in terms of abstract models (e.g., algorithms, data models, and interaction models).

Technological knowledge and skills concerns knowledge of computer systems, digital tools and languages, and programming. It focuses on students' ability to express computational ideas and models in digital artifacts. This includes the ability to use computer systems and the language associated with these and to express ideas through programming. Working within this area aims at providing students with the experience and abilities needed to make informed choices about the use of digital tools and technologies.

In summary, Denmark's curriculum focuses on relatively high-level abstract principles, with more focus on design as a general process rather than programming as a specific implementation. It also includes considerations of technology usage, and ethical issues surrounding digital technology.

2.3 Contrasting Curricula

Some content is shared between the English and Danish curricula. The Danish "computational thinking and modeling" mirrors the understanding of algorithms evident in the English curriculum. Similarly, the Danish "technological knowledge and skills" maps to the English use of computer programs to achieve objectives, as well as overlapping with some of the English programming content.

The Danish curriculum has a more distinct focus on design, which Caspersen [9] explains is partly inspired by the Scandinavian school of Participatory Design [13, 20]. The Danish curriculum also places greater emphasis on the ethics and design consequences of technology in society, which has a more limited presence in the English curriculum. The curriculum documents have very different approaches to their structure: the Danish curriculum is longer and more detailed: over 50 pages compared to the 2 pages of England's curriculum. This is not to say that either approach is necessarily better, but it shows a large variation in approaches to conveying the key items of a national education. Overall, the Danish curriculum has some wider, higher-level concepts than the English curriculum, which is relatively operational, with its list of specific (often programming-related) topics and learning objectives to be covered.

3 What should school informatics look like?

In order to design and implement curricula for informatics in schools, one must first have an overall vision for what that should entail: What are the foundational principles of the discipline, what are its practices, what are the relevant skills? In most countries, informatics is well established at university and at the end of high school (in the 15–18 age range), but is new for younger age groups. There is no apparent limit to how young informatics could be introduced in some form; England has introduced it from age 5 upwards. But what should this look like?

[3]See https://emu.dk/sites/default/files/2019-02/GSK.%20L%C3%A6seplan.Tilg%C3%A6ngelig.%20Teknologiforst%C3%A5else.%20pdf.pdf (in Danish) retrieved Dec 2024.

Most early initiatives (including those in England and Denmark) to introduce informatics at school level started at the upper end of the school curriculum: the subject was offered in the last two or three years of secondary school, usually as an elective. The content of such courses was initially heavily modelled on university content. Essentially, first-year university level informatics teaching was transferred directly to school level.

Later, when informatics teaching was expanded to younger age groups, the material was extended and adapted for younger learners, but the method was similar: existing curricula were extended backwards into younger years, to lead directly into existing courses. The English curriculum is a typical example: it focuses strongly on programming as a central practice, from discussing algorithms to building and implementing software systems. The flavour of the topics is decidedly aimed at software development as a practice and "computational thinking" as a principle.

While this approach works well as a path into an informatics-related career, it is in danger of under-serving a wider purpose: educating a general population of citizens, most of whom will not become software engineers. In extending the university curriculum downwards throughout school education, it has not extended from its original purpose: to educate computer scientists. Not every pupil will (or should) become a computer scientist, so how should the curriculum be framed? Elements of a good education are two-fold: they prepare learners for possible further study of a specific subject later, but they also convey a general body of knowledge to all pupils – whether they intend to become specialists or not – because this is seen as necessary to make sense of the world around us, and to function as an engaged, educated citizen in our society. In informatics education, we must ask: What is it that *every* educated citizen should know about computing and computers to be able to be meaningfully engaged in a modern world and a technology-infused society? We must accept that educating *all* pupils at school level is different from educating the few who select informatics as their chosen professional path.

3.1 Curriculum content

Every country has its own goals and constraints for school education, and its own approach to defining curricula. It is sensible to suggest an overarching framework and set of principles, that can then be referred to and incorporated by each individual country or region. This is the approach taken by a recent report from the "Informatics for all" coalition [10]. Aimed at Europe, but applicable everywhere, the report gives five aims for students at the end of secondary education, as summarised in a companion paper [8]:

(1) Use digital tools in a conscious, responsible, confident, competent, and creative way.
(2) Understand the principles and practices of informatics and their multifaceted applications.
(3) Analyse, design, frame and solve problems "informatically."
(4) Creatively develop computational models to investigate and communicate about phenomena and systems.
(5) Identify and discuss ethical and social issues of computational systems and their use, potential benefits and risks.

This is then translated in the report into 11 topic areas: Data and information; Algorithms; Programming; Computing systems; Networks and communication; Human-computer interaction; Design and development; Digital creativity; Modelling and simulation; Privacy, safety and security; and Responsibility and empowerment.

Mapping these aims and topic areas to the two national curricula previously summarised, we can see where they overlap, where they differ, and where they may have gaps that could be filled.

The English curriculum defines expectations for the first half of the topic areas well, and gives clear guidance and expectations for aims and implementation. Compared to the reference framework, it is more technology focused, and more influenced by preparation for further, professional study. The second half of topic areas, aimed to address more general issues such as social implications, ethics, creativity, responsibility and consequences, are less well defined or omitted from curriculum goals.

The Danish curriculum, on the other hand, has a broader coverage across the topic areas, and has stronger emphasis on informatics topics from the perspective of a general, non-specialist population. It has a more conscious "informatics for all" flavour; whether or not this comes at the expense of specific technical content useful as a preparation for further study as a software specialist will likely depend on details of implementation.

3.2 Computational thinking

One pervasive idea in school informatics teaching is that of computational thinking. First proposed by Wing [43], the idea states that the key part of informatics is a way of thinking. This follows notions such as the famous quote "Computer science is no more about computers than astronomy is about telescopes"[4], i.e. that computers are just an implementation device for a fundamental set of underlying principles. Computational thinking has led to two possible misconceptions about teaching informatics.

The first misconception is that computational thinking should be taught directly. Education research suggests [42] that a more promising approach to teach abstract concepts is to generalise from concrete examples, which should be studied first. That is, in order to teach the principles of what algorithms can do, we must first teach algorithms, which in turn are best taught with programming. So even if programming is not the end goal in school informatics, it is the fundamental building block of many other informatics concepts, and its mastery is a path to more general understanding.

The second misconception with computational thinking is that many of its proponents assume transferability: that the generic thinking skills learned through engaging with computation topics will automatically improve problem solving elsewhere. However, research on transfer across domains shows that it is a relatively limited effect that generally occurs late [1, 42]. Thus transfer is unlikely to be a major consequence of limited teaching of informatics in schools. These kinds of transferability argument have a long history, in relation to subjects such as Latin, chess [30] or music [31], but there is no consistent evidence to support them. The idea that computational thinking will suddenly change students' approach to other subjects does not hold up to scrutiny, and should be avoided as a selling point for informatics.

[4] Often attributed to Dijkstra but the first published mention seems to be by Fellows [16].

4 informatics for all: considerations

We believe that Informatics should be taught to all pupils. There are several important aspects of a change from Informatics for some (an elective optional subject) to Informatics for all (where all students must be taught). First, many more teachers must be trained. Second, if informatics is introduced for everyone, then that mandatorily includes those with disabilities and special educational needs[5]. There also remain other equity issues such as gender balance.

4.1 Teachers

If informatics is to be taught in all schools then naturally there must be teachers capable of doing so in all schools. For most countries, this will not yet be the case, either because the secondary (middle/high) school has not offered informatics as an elective at all, or because the existing informatics teachers are at capacity, or because in primary (elementary) schools there has not previously been any informatics teaching at that level.

This problem can only be solved by training more teachers: either training existing teachers of other subjects (or generalist teachers, in the case of primary schools) to teach informatics, or by training new informatics teachers. Given that the latter has a longer lead time and involves extra recruitment, most countries focused on training existing teachers (e.g. in the USA [46] and England [34, 35]).

Re-training teachers is a serious endeavour and can be a major difficulty in introducing informatics into schools. England changed its curriculum without initially providing funding for training teachers, leaving voluntary associations scrambling to help [6, 34, 35]. Funding was later provided, but not time out of teachers' schedules to re-train, so it often had to be done in the evenings, weekends and holidays at the teacher's own discretion.

If informatics is introduced without support to (re-)train teachers then the subject is being set up to fail. It is a testament to the voluntary organisations in England, first and foremost "Computing At School" (CAS), that this period has now been navigated with reasonable success. The trial of informatics at primary and lower secondary school in Denmark has warned about exactly this issue [7]. Some countries have stricter requirements on subject teaching and may require specialist degrees for teaching a particular subject, which will prevent or complicate this re-training, and instead require new teachers or more extensive training – see Hubwieser et al. [23] for a global survey of this issue.

4.2 Special educational needs

The definitions (and terminology) of special educational needs vary by country and culture, which makes it difficult to give a global average of how many people it affects. To give a concrete example here, from one of our case study countries, 18% of students in England either have a personal "education, health and care plan" or require special educational needs support[6]. It is apparent that even as an elective subject this should have been an important consideration for equity, but as a compulsory subject it is critical.

There has been work with special education teachers to help them teach computer science [3], and work on adapting teaching approaches and materials [41]. One trial of teaching informatics to primary school students with Down's syndrome was successful [39]. However, more work clearly needs to be done to support teachers to teach informatics to those with special educational needs.

4.3 Disabilities

There are some disabilities that – with some general consideration in teaching – do not directly prevent students taking part: for example, deaf students require some terminology and language considerations [25] but are not inherently prohibited from programming. However, there are some disabilities that are more difficult to adjust for, such as vision impairment [37].

Not being able to read text on screen directly is potentially a critical impediment to being able to program. This can be mitigated through the use of assistive technologies such as screen readers. Unfortunately, screen readers are generally designed for prose, and they can be difficult to use with programming code because, for example, by default they pay no attention to white-space or to reading some punctuation. For natural text this is unnecessary information but for some programming languages it is critical. Furthermore, many block-based programming languages lack support for screen reading, making them unusable for blind learners. There have been efforts to fix this problem, such as the Quorum [37] programming language, which has also recently added a blocks-based mode. Nevertheless, this shows the kind of hurdles that must be overcome for students with disabilities if informatics is truly to be taught to all.

4.4 Gender

The issue of gender balance in informatics is well-known and longstanding [26]. The particular issue of interest here is gender balance in *school* informatics. Some research in England since the subject change to Computing suggests that the new subject is less gender-balanced than the ICT subject which preceded it [12, 24]. The hope had been that since informatics is mandatory in England from age 5–14 (with caveats described earlier) that when the subject becomes optional at age 14, this would help redress the gender balance because everyone would have had a chance to experience the subject fully, and thus hopefully be more likely to continue with it after age 14. Instead, it appears that perhaps the societal perception of informatics has persisted and thus informatics has brought the gender imbalance with it into schools despite the curriculum changes.

5 How to initiate change

It is not trivial to put informatics into schools. This change must typically be instigated by the government, based on lobbying, and is fraught with potential pitfalls. Each country will have a different starting point and history from which to commence the change. Below, we examine the process for each of our case study countries.

5.1 England

The path to getting informatics into the English national curriculum was multi-faceted [5, 6]. An economic argument initially gained the attention of politicians, but a pedagogical argument convinced the Department for Education that computing was viable at all age

[5] Those with disabilities and special educational needs should *always* have had the opportunity to study informatics, even when it was an elective. Sadly, due to the extra effort required to accommodate them, the informal solution has generally been to dissuade them from studying informatics rather than supporting them [32].
[6] See https://explore-education-statistics.service.gov.uk/find-statistics/special-educational-needs-in-england, retrieved December 2024.

groups, and that a stable curriculum would be possible. Computing replaced ICT, with an argument that Computing was a more rigorous subject (where academic rigour was a government focus at the time). To quote one person involved in the lobbying: "we went knocking on the [Department for Education's] door and expected a lot of resistance, but the door was open and we fell through it."

The curriculum change turned out to be one of the easier parts of the effort. At the time, in the early 2010s, the UK was recovering from the global economic downturn of 2008 and was in the full swing of austerity politics, with reduced government spending across the board. Informatics was introduced into the national curriculum, but there was initially no money for training the teachers (and a later grant that was modest, considering it was nationwide). Thus informatics in English schools was in danger of being set up to fail: introduced but without sufficient necessary support.

Ultimately, this period has been navigated reasonably successfully. Various voluntary organisations, such as CAS, stepped up to provide training for teachers to help bridge the gap [34, 35].

5.2 Denmark

In Denmark, informatics had nearly vanished as a subject by the mid 2000s [11] due to lack of interest and changes in the structure of secondary education. In late 2008, a government task force recommended developing a more holistic approach to computing, which became the basis for a new upper secondary subject relevant for all. It was trialled for four years and made permanent in 2015.

Until 2016, successive Danish governments had shown little interest in introducing informatics into primary and lower secondary school. When asked about introducing IT education into schools in early 2016, the Danish Prime Minister discussed issues of digital whiteboards and IT provision, rather than informatics education.

In 2016, the informatics zeitgeist began to influence Denmark. That year, the World Economic Forum published a report [44] which included ideas parallel to computational thinking, and named computational thinking in the follow-up a year later [45]. In 2018, Denmark produced the Technology Pact[7] (as relayed by Olofsson [27]), with a focus on upskilling the population (via formal education and industry) to match the growth in digital jobs.

Progress on the issue was stalled by a change in government; the pilot of primary and lower secondary (described earlier) was commissioned directly by the ministry. A positive evaluation was reported to the following government, which did not follow it up. Now the government has changed again and progress may be made. In 2021 a "National Alliance for Digital Technology Comprehension" (English translation) was formed[8] between schools, teachers' unions, universities and industry to promote technology understanding in schools (to include informatics); this may yet bear fruit.

5.3 Public perception of informatics

Schools have a fixed amount of time over the year in which to teach pupils. Therefore time for teaching anything new must inherently be taken away from existing teaching. This will invariably result in some controversy. In England, for example, the teaching of Computing was added in place of the existing ICT curriculum. Some teachers were happy with the replacement, and others were not.

There were questions in England over the motivation of adding informatics to the curriculum. Some of the most-amplified voices for change were those from industry[9], such as Google's Eric Schmidt who gave a much-reported speech in Edinburgh on the issue[10]. This led to a perception that it was an industry-driven change. Casting around for a motivation, some believed that industry just wanted a cheaper way to train up employees [4, 28] and had unduly influenced government for their own direct benefit. This would have been a long-term strategy in such a fast-moving industry as tech then was: informatics was introduced at age 5 in 2014, meaning it would be another 16 years (in 2030) before those students who began at five would graduate from university – at the time Schmidt made his comments in 2011, Google itself was only 13 years old.

Although there was outcry from some teachers and some researchers, there was no noticeable pushback in England from parents. We suggest that in the public mind it was clear that there was a growth in employment opportunities in the technology sector, and an increasing dominance of computers in society, and thus it was sensible to include informatics in the curriculum. Despite a prolonged search, we are not aware of any research on parents' (or the wider public's) opinions of this change in the curriculum.

5.4 Political change

The process of political change is complex. It is a fragile process, vulnerable to changes in governments (and thus policy directions) or even just changes in ministers of education. For example, in the ten years December 2014–2024 England had ten successive ministers of education. Some of that relates to political instability (2022 saw five different education ministers under three different prime ministers), but the 20 years before that saw 11 different ministers. In Denmark, there have been six ministers of education in the last ten years. It is clear that an approach relying solely on an individual minister is unlikely to prevail in such a two year "life span", and will need wider, persistent government support.

In England the process happened to fall at the beginning of a new right-leaning government which wanted to increase rigour in the curriculum [2]. In Denmark a similar situation occurred, but was delayed by changes in government to parties that were less receptive. Overall, government support is vital (including wide enough parliamentary support, not just individual ministers of education) but its fragility can be counteracted through continued wider public support for informatics, to sway successive governments.

6 Discussion

We end by discussing our questions: why, what, how and who?

6.1 Why: the value of informatics in schools

Informatics is a stable, academic and rigorous subject that has increasing importance in our modern digital society. Although many

[7] For an announcement in English, see https://www.eng.em.dk/news/2018/apr/the-danish-government-launches-the-technology-pact retrieved December 2024.

[8] See the announcement (in Danish) https://www.folkeskolen.dk/arbejdsliv-dlf-it/dlf-med-i-ny-alliance-teknologiforstaelse-skal-ind-i-hele-uddannelsessystemet/2178952, retrieved December 2024.

[9] One could argue this was because industry was most listened to by the media and government, in contrast to universities or teachers – a longstanding global problem [19].

[10] See for example https://www.theguardian.com/technology/2011/aug/26/eric-schmidt-chairman-google-education retrieved December 2024.

arguments to promote it are economic (which we discuss below in the "how"), these should not obscure the civic and academic arguments behind informatics. There are many misconceptions about informatics: that it consists of mere technology usage; that its content changes too quickly to be an academic subject; or that it is just programming with no wider principles or applicability. We believe that informatics is important enough that it should be included in mandatory schooling in secondary education, and even primary education. It has applicability for all students, and understanding the power and limits of technology is important for a modern society which is increasingly based on digital technology.

6.2 What: the shape of informatics in schools

Informatics is a sufficiently rich subject that there is room for a lot of variation about what should be covered at school level. There could be a curriculum designed around programming, or a curriculum featuring almost no programming. Ethics and society could be a key focus or completely absent. This is both a positive feature – showing the width and depth of the subject – but also a potential headache for those who have to decide what should be taught. In this regard, we defer to the Informatics for All report [10] which has considered these issues and produced a framework and suggestions for informatics curricula. Although programming is a central part of informatics, and should be included, there are many aspects more suited to our aim of creating an informed society: ethical issues, societal impact and principles of digital technology. The more technical and scientific aspects can be reserved for follow-up specialist instruction when the subject is elective for older age groups. However, a basic knowledge of programming may be needed in order to teach topics such as AI, data representation, algorithms, etc.

6.3 How: introducing informatics into schools

Ultimately, introducing informatics into primary and secondary schools before the elective stage requires convincing governments or similar institutions to change the curriculum. Although educators might prefer otherwise, the dominant force for curriculum change in many countries is economic [19, 33]. In this regard, informatics has the prevailing wind: despite natural ups and downs, programming remains in demand and salaries are high. There remain other challenges, however. Governments may need convincing that informatics can have a stable curriculum that is not rapidly outdated by technology changes. Any curriculum change will require displacement of another subject; sometimes this is a related subject such as IT, but sometimes it may require arguing to displace (parts of) other subjects. A large-scale change will require buy-in from teachers who must be (re-)trained to satisfy the new demand for informatics teachers. To ensure success, this should be government-funded.

There are other potential barriers to introducing school informatics which we have not covered here. The CAPE framework [17] describes several levels of potential issues in introducing informatics education: capacity, access, participation and experience. Utilising this framework, Tshukudu et al. [40] found that some African countries struggle with issues of capacity – some of which appear in our case studies, such as curriculum and teacher training issues, while others were not an issue in England and Denmark, such as access to ICT infrastructure, and funding for equipment.

6.4 Who: issues of equity

Equity is complex and multi-faceted. If informatics is offered at all, then it is important that it is made available and accessible to all students. This includes students with disabilities, and students with special educational needs (such as learning disorders). Some technology used for teaching informatics can exclude students with disabilities: for example, block-based programming systems are often unusable by blind students [37]. This is a key equity issue, and one which there is a moral imperative to solve. It is, however, often out of the control of individual teachers and falls on the designers of these tools. There are also longstanding issues around the balance of who takes informatics as an elective. This is obviated (but not necessarily solved!) if informatics is mandatory. Educators hypothesised that making informatics mandatory early on would fix gender imbalances when it was later elective, but early evidence suggests this has not panned out [24]. Thus far, informatics seems to retain its equity issues regardless of changes to the curriculum.

7 Conclusion

In this paper, we have discussed *what* informatics for all in schools should focus on, *why* it should be taught, *how* to introduce it into the curriculum, and *who* it should be available to: everyone. We have presented case studies from England and Denmark, discussing how they introduced informatics into schools, and the shape of their curricula.

We believe that the right arguments for informatics surround its usefulness for all students to understand and live in the modern digital world, rather than being directed only at those who wish to continue studying informatics at university. School teaching should not be constrained to technology use, but instead should involve important principles of computing: how data is stored, the power and limitations of algorithms and AI, the brute force but unintelligent power of computers. Programming is the ideal concrete vehicle with which to teach some of these abstract concepts, but it should it not be the focus for its own sake.

However, this civic argument is not always the strongest one with which to convince governments. The economic argument is often received more favourably. Informatics does have an advantage in this area in that the jobs are well-paid and have long-term growth potential. Work may also be needed to convince governments that there is a stable curriculum with unchanging underlying principles.

Finally, any large change in the teaching of informatics will require teacher (re-)training, and this is very difficult without government funding, so the introduction of informatics would need to come with such funding.

Informatics teaching continues to be in differing states throughout the world. We have detailed case studies of England, where it is widespread from age 5–14, and Denmark, where it is currently only widespread in upper secondary school. Introducing informatics is a country-by-country, or even state-by-state, effort. However, encouraging signs, such as the EU's recommendation to introduce it, suggest that political will exists, and a fair wind may help advance the subject in many global regions, and lead towards informatics for all.

References

[1] Susan M Barnett and Stephen J Ceci. 2002. When and where do we apply what we learn?: A taxonomy for far transfer. *Psychological bulletin* 128, 4 (2002), 612.

[2] Elizabeth Bate. 2020. Justifying music in the national curriculum: The habit concept and the question of social justice and academic rigour. *British Journal of Music Education* 37, 1 (2020), 3–15. doi:10.1017/S0265051718000098

[3] Emily C. Bouck, Phil Sands, Holly Long, and Aman Yadav. 2021. Preparing Special Education Preservice Teachers to Teach Computational Thinking and Computer Science in Mathematics. *Teacher Education and Special Education* 44, 3 (2021), 221–238. doi:10.1177/0888406421992376

[4] Nina Bresnihan, Richard Millwood, Elizabeth Oldham, Glenn Strong, and Diana Wilson. 2015. A critique of the current trend to implement computing in schools. *Pedagogika* 65, 3 (2015), 292–300.

[5] Neil C. C. Brown, Michael Kölling, Tom Crick, Simon Peyton Jones, Simon Humphreys, and Sue Sentance. 2013. Bringing Computer Science Back into Schools: Lessons from the UK. In *Proceeding of SIGCSE'13*. ACM, 269–274. doi:10.1145/2445196.2445277

[6] Neil C. C. Brown, Sue Sentance, Tom Crick, and Simon Humphreys. 2014. Restart: The Resurgence of Computer Science in UK Schools. *ACM Trans. Comput. Educ.* 14, 2, Article 9 (June 2014), 22 pages. doi:10.1145/2602484

[7] BØRNE- OG UNDERVISNINGSMINISTERIET. 2021. *FORSØG MED TEKNOLOGIFORSTÅELSE I FOLKESKOLENS OBLIGATORISKE UNDERVISNING: Slutevaluering*. Technical Report. Danish Government. https://emu.dk/sites/default/files/2021-09/gsk_teknologiforst%C3%A5else_evalueringsrapport.pdf

[8] Michael Caspersen, Judith Gal-Ezer, Andrew McGettrick, and Enrico Nardelli. 2023. Informatics Education for School: A European Initiative. *ACM Inroads* 14, 1 (feb 2023), 49–53. doi:10.1145/3583088

[9] Michael E. Caspersen. 2022. *Informatics as a Fundamental Discipline in General Education: The Danish Perspective*. Springer, Cham, 191–200. doi:10.1007/978-3-030-86144-5_26

[10] Michael E. Caspersen, Ira Diethelm, Judith Gal-Ezer, Andrew McGettrick, Enrico Nardelli, Don Passey, Branislav Rovan, and Mary Webb. 2022. *Informatics Reference Framework for School*. Technical Report. Informatics for all. https://www.informaticsforall.org/the-informatics-reference-framework-for-school-release-february-2022/

[11] Michael E. Caspersen and Palle Nowack. 2013. Computational Thinking and Practice: A Generic Approach to Computing in Danish High Schools. In *Proceedings of ACE'13*. Australian Computer Society, Inc., AUS, 137–143.

[12] Meggie Copsey-Blake, Jessica Hamer, Peter Kemp, and Billy Wong. 2021. Should we be concerned about who is studying computing in schools? *Raspberry Pi Foundation Research Seminars* 2 (2021), 1–8.

[13] Bronwyn Cumbo and Neil Selwyn. 2022. Using participatory design approaches in educational research. *International Journal of Research & Method in Education* 45, 1 (2022), 60–72. doi:10.1080/1743727X.2021.1902981

[14] European Comission / Eurydice. 2022. *Informatics Education at School in Europe*. Technical Report. European Commission. https://eurydice.eacea.ec.europa.eu/publications/informatics-education-school-europe

[15] European Commission. [n. d.]. Digital Education Action Plan (2021–2027). https://education.ec.europa.eu/focus-topics/digital-education/action-plan Accessed 30 June 2023.

[16] Michael R Fellows. 1991. *Computer science and mathematics in the elementary schools*. University of Victoria, Department of Computer Science.

[17] Carol L. Fletcher and Jayce R. Warner. 2021. CAPE: A Framework for Assessing Equity throughout the Computer Science Education Ecosystem. *Commun. ACM* 64, 2 (Jan. 2021), 23–25. doi:10.1145/3442373

[18] Megean Garvin, Michael Neary, and Marie Desjardins. 2019. State Case Study of Computing Education Governance. *ACM Trans. Comput. Educ.* 19, 4, Article 35 (July 2019), 21 pages. doi:10.1145/3320491

[19] Tony Greening. 2000. Pedagogically Sound Responses to Economic Rationalism. *SIGCSE Bull.* 32, 1 (March 2000), 149–156. doi:10.1145/331795.331845

[20] Judith Gregory. 2003. Scandinavian approaches to participatory design. *International Journal of Engineering Education* 19, 1 (2003), 62–74.

[21] Mark Guzdial, Barbara Ericson, Tom Mcklin, and Shelly Engelman. 2014. Georgia Computes! An Intervention in a US State, with Formal and Informal Education in a Policy Context. *ACM Trans. Comput. Educ.* 14, 2, Article 13 (June 2014), 29 pages. doi:10.1145/2602488

[22] Peter Hubwieser. 2012. Computer Science Education in Secondary Schools – The Introduction of a New Compulsory Subject. *ACM Trans. Comput. Educ.* 12, 4, Article 16 (Nov. 2012), 41 pages. doi:10.1145/2382564.2382568

[23] Peter Hubwieser, Michail N. Giannakos, Marc Berges, Torsten Brinda, Ira Diethelm, Johannes Magenheim, Yogendra Pal, Jana Jackova, and Egle Jasute. 2015. A Global Snapshot of Computer Science Education in K-12 Schools. In *Proceedings of ITICSE-WGR'15*. ACM, 65–83. doi:10.1145/2858796.2858799

[24] Peter E. J. Kemp, Billy Wong, and Miles G. Berry. 2019. Female Performance and Participation in Computer Science: A National Picture. *ACM Trans. Comput. Educ.* 20, 1, Article 4 (nov 2019), 28 pages. doi:10.1145/3366016

[25] Richard E. Ladner, Andreas Stefik, Jill Naumann, and Erin Peach. 2020. Computer Science Principles for Teachers of Deaf Students. In *2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, Vol. 1. 1–4. doi:10.1109/RESPECT49803.2020.9272432

[26] Jane Margolis and Allan Fisher. 2002. *Unlocking the clubhouse: Women in computing*. MIT press.

[27] Anders D. Olofsson, J. Ola Lindberg, Alex Young Pedersen, Ann-Thérèse Arstorp, Christian Dalsgaard, Even Einum, Francesco Caviglia, Liisa Ilomäki, Marjaana Veermans, Päivi Häkkinen, and Sara Willermark. 2021. Digital competence across boundaries - beyond a common Nordic model of the digitalisation of K-12 schools? *Education Inquiry* 12, 4 (2021), 317–328. doi:10.1080/20004508.2021.1976454

[28] L Pinkerton. 2018. Big Tech's privatization of schools: a case study of computing education in England. *BSA Digital Sociology Study Group Event (ACM WebSci'18)* (2018). https://ora.ox.ac.uk/objects/uuid:29cf1c13-10f5-4686-9d7b-952a262d24b7/files/mf33375b86895fae36e17f61d22aa7878

[29] Nerys Roberts. 2021. The school curriculum in England. *Parliament. House of Commons.* (2021). https://commonslibrary.parliament.uk/research-briefings/sn06798/

[30] Giovanni Sala and Fernand Gobet. 2016. Do the benefits of chess instruction transfer to academic and cognitive skills? A meta-analysis. *Educational Research Review* 18 (2016), 46–57. doi:10.1016/j.edurev.2016.02.002

[31] Giovanni Sala and Fernand Gobet. 2017. Does Far Transfer Exist? Negative Evidence From Chess, Music, and Working Memory Training. *Current Directions in Psychological Science* 26, 6 (2017), 515–520. doi:10.1177/0963721417712760 PMID: 29276344.

[32] Cueponcaxochitl D. Moreno Sandoval, David I. Hernández Saca, and Adai A. Tefera. 2021. Intersectional Rights of Teachers and Students in Computer Science and Special Education: Implications for Urban Schooling. *Urban Education* 56, 5 (2021), 675–704. doi:10.1177/0042085917714512

[33] Glenn Savage. 2017. Neoliberalism, education and curriculum. *Powers of curriculum: Sociological perspectives on education* (2017), 143–165.

[34] Sue Sentance and Simon Humphreys. 2018. Understanding professional learning for Computing teachers from the perspective of situated learning. *Computer Science Education* 28, 4 (2018), 345–370. doi:10.1080/08993408.2018.1525233

[35] Sue Sentance, Simon Humphreys, and Mark Dorling. 2014. The Network of Teaching Excellence in Computer Science and Master Teachers. In *Proceedings of WiPSCE'14*. ACM, 80–88. doi:10.1145/2670757.2670789

[36] Craig Skerritt. 2019. 'I think Irish schools need to keep doing what they're doing': Irish teachers' views on school autonomy after working in English academies. *Improving Schools* 22, 3 (2019), 267–287.

[37] Andreas Stefik, Richard E. Ladner, William Allee, and Sean Mealin. 2019. Computer Science Principles for Teachers of Blind and Visually Impaired Students. In *SIGCSE 2019*. ACM, 766–772. doi:10.1145/3287324.3287453

[38] Paul Stephens, Finn Egil tønnessen, and Chris Kyriacou. 2004. Teacher training and teacher education in England and Norway: a comparative study of policy goals. *Comparative Education* 40, 1 (2004), 109–130. doi:10.1080/0305006042000184908

[39] Matthew S. Taylor, Eleazar Vasquez, and Claire Donehower. 2017. Computer Programming with Early Elementary Students with Down Syndrome. *Journal of Special Education Technology* 32, 3 (2017), 149–159. doi:10.1177/0162643417704439

[40] Ethel Tshukudu, Sue Sentance, Oluwatoyin Adelakun-Adeyemo, Brenda Nyaringita, Keith Quille, and Ziling Zhong. 2023. Investigating K-12 Computing Education in Four African Countries (Botswana, Kenya, Nigeria, and Uganda). *ACM Trans. Comput. Educ.* 23, 1, Article 9 (Jan. 2023), 29 pages. doi:10.1145/3554924

[41] Sarah Wille, Jeanne Century, and Miriam Pike. 2017. Exploratory Research to Expand Opportunities in Computer Science for Students with Learning Differences. *Computing in Science & Engineering* 19, 3 (2017), 40–50. doi:10.1109/MCSE.2017.43

[42] Daniel T Willingham. 2021. *Why don't students like school?: A cognitive scientist answers questions about how the mind works and what it means for the classroom*. John Wiley & Sons.

[43] Jeannette M. Wing. 2006. Computational Thinking. *Commun. ACM* 49, 3 (March 2006), 33–35. doi:10.1145/1118178.1118215

[44] World Economic Forum. 2016. New vision for education: Fostering social and emotional learning through technology. World Economic Forum Geneva. http://www3.weforum.org/docs/WEF_New_Vision_for_Education.pdf

[45] World Economic Forum. 2017. Realizing human potential in the Fourth Industrial Revolution: An agenda for leaders to shape the future of education, gender and work. In *World Economic Forum*. https://www3.weforum.org/docs/WEF_EGW_Whitepaper.pdf

[46] Aman Yadav, Sarah Gretter, Susanne Hambrusch, and Phil Sands. 2016. Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education* 26, 4 (2016), 235–254.