

L'enjeu du code et de la data



Publié le 10/09/2014

 **Intervenants**

Paul Gibson

Maître de conférence, département
Logiciel et Réseaux (LOR)

Agenda SDBX 4

CONFÉRENCE PLÉNIÈRE
L'e-éducation dans la clas

Intervenant lors de la Journée internationale de réflexion autour de la e-education le 15 octobre prochain, Paul Gibson, maître de conférences au sein du Département Logiciels - Réseaux à Télécom Sud Paris, abordera le sujet de l'enseignement du code dès les plus jeune âge. En amont de cette conférence, Paul Gibson détaille sa théorie concernant les enjeux actuels du code chez les enfants.

SDBX4 : Bonjour Paul. Vous avez dirigé un projet de recherche « L'enseignement innovateur et l'apprentissage par problème », il y a 10 ans. En quoi l'enseignement a-t-il évolué depuis ?



Paul Gibson : Le plus grand changement des 10 dernières années a été le développement d'Internet et du WWW (World Wide Web) : les moteurs de recherche et les réseaux sociaux sont devenus des technologies traditionnelles.

Les ordinateurs sont omniprésents, les enfants se sont familiarisés avec les PC, les smartphones, les tablettes, les jeux

vidéos, les écrans tactiles, les gestes et les techniques de reconnaissances vocales (les interfaces technologiques ne gênent plus les enfants).

Avec les moteurs de recherches, il y a une transition importante concernant l'absorption de connaissances autour des méthodes de recherche d'information et leur application (dans un certain contexte). En d'autres termes, l'éducation doit se concentrer sur les hauts niveaux comme décrits selon la taxonomie de Bloom : Connaissance (1), Compréhension (2), Application (3), Analyse (4), Evaluation (5) et Synthèse (6).

Trouver le bon équilibre entre les niveaux est un challenge pédagogique important. Dans un cas extrême, si nous enseignons juste les faits (niveau 1), nous risquons d'avoir des élèves qui ne parviennent pas à comprendre ce qu'ils ont étudié. Et dans un autre cas, si nous n'enseignons pas les faits importants, les élèves trouveront difficile de juger la valeur de l'information qu'ils ont trouvée sur le web (ou ailleurs).

L'éducation traditionnelle (dans des pays comme la France), a trop accentué les niveaux inférieurs de la taxonomie et a été résistante au changement. D'autres pays ont adapté cette méthode plus rapidement.

Avec les réseaux sociaux et les technologies de communication, la coopération entre les élèves est devenue plus importante que la compétition. Le travail en groupe est crucial dans la résolution de problèmes et l'apprentissage par projet, et les nouvelles technologies sont un excellent support pour une telle collaboration. En revanche, l'évaluation à l'école reste centrée sur les individus seuls. Cela encourage la compétition plutôt que la coopération. Dans certains pays (comme en France), la compétition démarre à un âge très précoce et devient une coutume culturelle. A nouveau, il s'agit vraiment de problèmes d'équilibre. Dans un vieux monde déconnecté, la compétition était une clé à la réussite individuelle. Dans le nouveau monde, l'éducation doit décaler cet équilibre vers la coopération. La recherche a montré que les deux choses étaient complémentaires et essentielles à la réussite.

Avec la technologie omniprésente, les enfants s'attendent à une

classe moderne et digitale. Ils maîtrisent les technologies que leurs enseignants ne maîtrisent pas. C'est beaucoup plus difficile pour un enseignant de maintenir des élèves intéressés et motivés s'ils n'arrivent pas à adopter ces nouvelles technologies en classe. Bien entendu, les enseignants sont toujours exposés au changement, et les nouvelles technologies peuvent les seconder. Cependant, la vitesse de l'évolution des technologies est aujourd'hui plus rapide qu'à d'autres époques dans l'histoire. Il y a un gros risque à ce que les enseignants ne rattrapent pas pleinement les avancées technologiques, ayant déjà du mal à manipuler les outils qui sont supposés les aider plutôt que de les entraver.

Depuis 3-4 ans, nous avons commencé à voir encore plus de technologie de techniques d'enseignement innovante : classes inversées, apprentissage d'analyse, gamification (jeux sérieux), de l'apprentissage adapté aux environnements, assistants virtuels etc...

Mais, les 10 dernières années ont montré qu'il était difficile d'intégrer les nouvelles technologies en classe de manière cohérente. Le problème pourrait s'accroître dans le futur et il demanderait une nouvelle façon de gérer un environnement pédagogique.

The biggest change in the last 10 years has been the development of the internet and world wide web: search engines and social networks have become mainstream technologies. Computers are ubiquitous - children are familiar with PCs, smart phones, tablets, video games, touch screens, gesture and speech recognition (the interfaces to the technology are no longer hindering children).

With search engines, there is a shift away in importance from having knowledge towards knowing how to find relevant information and apply it (in a certain context). In other words, education has to focus on the higher levels of Bloom's Taxonomy {1) Knowledge, 2) Comprehension, 3) Application, 4) Analysis, 5) Evaluation and 6) Synthesis.}

Finding the right balance between the levels is a significant educational challenge. At one extreme, if we just teach facts (at level 1) we risk having learners who fail to understand what they have learned. At the other extreme, if we don't teach the important facts then learners will find it difficult to judge the value of information that they find on the web (or elsewhere). Traditional education (in countries like France) has placed too much emphasis on the lower levels of the taxonomy and have been resistant to change. Other countries have adapted more quickly.

With social networks and communication technologies, co-operation between learners is becoming more important than competition. Team-work is critical in problem solving and project-based education, and new technologies provide excellent support for such collaboration. However, evaluation in school focuses on individuals. This encourages competition rather than co-operation. In some countries (like France) competition starts at a very early age and becomes part of the culture. Again, the problem is one of balance. In the old disconnected world, competition was a key to individual success. In the new world, education has to shift the balance towards co-operation. Research has shown that the two things are complementary and critical to success.

With ubiquitous technology, children expect the classroom to be modern and electronic. They may have mastered technologies that their teachers have not. It is much more difficult for a teacher to keep students interested and motivated if they fail to adopt these new technologies in the classroom. Of course, teachers have always been exposed to change; and new technologies can support teachers. However, the pace of change of technology is now faster than in any time in history. There is great risk that teachers will never fully catch up with the technological advances, struggling to master tools that are supposed to help, rather than hinder, them.

In the last 3-4 years we are starting to see even more innovative technology and teaching techniques: flipped classrooms, learning analytics, gamification (serious games), adaptive learning environments, virtual assistants, etc...

However, the last 10 years has also shown that it is very difficult to integrate new technologies in the classroom in a coherent manner. This problem is likely to get more complex in the future, and requires a new way of managing the education environment.

SDBX4 : Quels sont les enjeux du coding en réalité ?

Paul Gibson : Tout le monde s'accorde à dire que faire apprendre le code aux enfants est une bonne chose. Premièrement, le code informatique se trouve dans tous les appareils électroniques et c'est important que les enfants comprennent les bases de ce que le code fait et comment il le fait. Sans ce code, le monde tel qu'on le connaît ne pourrait pas exister et donc, si nous voulons que nos enfants comprennent le monde dans lequel ils vivent déjà, ils ont besoin d'apprendre ce code.

Deuxièmement, il est évident qu'apprendre un programme/un code aide les enfant dans d'autres circonstances : les langues, les mathématiques, l'art, la musique, la science etc...

Programmer suppose un type de raisonnement informatique (la compréhension algorithmique) qui est fondamentale dans la manière dont le cerveau fonctionne et se développe.

Finalement, d'un point de vue plus pragmatique, la compétence de programmation apportera de bonnes opportunités professionnelles, aussi bien à court terme qu'à long terme.

La vraie question revient à se demander à quel moment le fait de savoir programmer peut être inséré dans le CV. Ma propre expérience (complétée par d'autres recherches) insinue que nous devrions commencer l'enseignement des concepts de programmation aussi tôt qu'ils entreront à l'école, et même avant qu'ils peuvent lire ou écrire. Les enfants âgés entre 5 et 11 ans ont plus de potentiel dans l'apprentissage des algorithmes et de l'informatique que ce serait une honte que d'attendre leur adolescence pour leur apprendre les bases. (Il faudrait qu'il soit noté qu'il n'y a pas de consensus là-dessus parce que d'autres enseignants-chercheurs suivent la théorie de Jean Piaget affirmant qu'un cerveau d'enfant ne sera pas suffisamment évolué pour travailler avec des concepts abstraits à un tel âge précoce).

La majorité des gens, incluant les enseignants, sont sceptiques sur un démarrage si tôt. Un argument commun - avec lequel je suis d'accord pour l'essentiel - avance qu'il est très important que nous ne prenions aucun risque inutile concernant l'éducation de nos enfants. Au moins, quand les ressources pédagogiques sont trop élastiques, il y a un risque qu'en s'investissant dans l'apprentissage de la programmation cela réduirait le soutien crucial et l'implication dans d'autres domaines clés.

Dans le pire des cas, l'apprentissage précoce de la programmation dans le développement d'un enfant peut avoir un impact négatif sur d'autres aspects de son développement de connaissances. Je pense que la formation au programme/ à la pensée algorithmique donne plus d'avantages et que nous devons essayer d'évaluer et gérer les risques, qui je crois, sont minimes, d'une meilleure façon.



Et en particulier, à la question "quand les techniques du code pourront être enseignées", il y aura un intérêt majeur concernant la légitimité de la personne qui l'enseignera.

Aujourd'hui, les enseignants d'écoles primaires n'ont - en général - pas assez eux-mêmes, de compétences en code. Plus profondément, dans beaucoup de pays (comme la France),

l'informatique - à l'école - n'est pas considérée comme étant une discipline académique dans son bon droit ayant un programme scolaire correspondant. Par conséquent, il y a quelques, voire aucun, enseignant spécialisé en informatique et qui pourraient être les plus à même de faire apprendre le code, même dans le cycle secondaire. Le code a besoin d'être enseigné par des enseignants suffisamment qualifiés. Cela ne peut pas être acquis sans un investissement de longue durée dans le développement du programme pédagogique et l'entraînement à cette formation.

There is general acceptance that it is a good thing if all children learn to code.

Firstly, computer code is found in all electronic devices and it is important that children understand the basics of what this code is doing and how it is doing it. Without this code, the world as we know it would not exist and so if we want our children to understand the world in which they live then they need to learn about this code.

Secondly, there is evidence that learning to program/code helps children in other areas : language, mathematics, art, music, science, etc... Programming involves a type of computational reasoning (algorithmic understanding) which is fundamental to the way in which the brain works and develops. Finally, from a more pragmatic view-point, programming skills will provide good employment opportunities, both in the short and long-term.

The real issue is when programming should be introduced into the curriculum. My own experience (which is backed up by other research) suggests that we should start teaching programming concepts as soon as they start school, even before they can read or write. Children aged from 5- 11 have so much potential for learning about algorithms and computation that it would be a shame to wait until they are teenagers before we teach them the foundations. (It should be noted that there is no consensus on this because other researchers/educators follow Jean Piaget's theory that a child's brain will not have evolved sufficiently to work with abstract concepts at such an early age.)

The majority of people, including teachers, are skeptical about starting so early.

A common argument - that I generally agree with - is that it is very important that we do not take unnecessary risks with the education of our children. At the least, when educational resources are so stretched, there is a risk that investing in teaching programming will reduce critical support and investment in other key areas.

At the worst, teaching programming early in a child's development may have negative impact on other aspects of the child's learning. My view is that learning programming/algorithmic thinking has so many benefits that we should try to better assess and manage the risks, which I believe are minimal.

As well as the question of 'when programming should be taught', there is a major concern regarding who should do the teaching.

Currently, teachers in primary schools do not - in general - have sufficient coding skills themselves. Further, in many countries (like France)

computing - at school - is not considered to be an academic discipline

in its own right, with a corresponding curriculum. Thus, there are few, if any, dedicated

computing teachers who would be most able to teach programming, even at secondary level. Coding needs to be taught by teachers who are

suitably qualified. This cannot be achieved without long-term investment in curriculum development and teacher training.

SDBX4 : Lors du teaser de la Semaine Digitale, Bernard Ourghanlian, de Microsoft a indiqué que « il faut coder pour décoder le monde », qu'en pensez-vous ?

Paul Gibson : Mon interprétation de la phrase est que pour comprendre le monde, vous devez comprendre le code. C'est une version plus forte de ce que j'ai dit précédemment. Même si on ignore les systèmes électroniques (programmés), il y a beaucoup de calculs par nature, guidé par un code naturel. Le cerveau est un exemple indéniable à cela. Il est un ordinateur de plusieurs types. Mais d'autres exemple incluent l'ADN, l'accroissement de la population, l'environnement, le climat, etc... Peut-être que Bernard était en train de faire allusion à cette importance plus large concernant le « code » en général ?

My interpretation of the phrase is that to understand the world you need to understand code. This is a stronger version of what I was saying earlier. Even if we ignore electronic (programmed) devices there is a lot of computation in nature driven by natural code. An obvious example of this is the brain (which is a computer of sorts), but other examples include DNA, population growth, the environment, climate, etc.... Perhaps Bernard was hinting at this wider importance of 'code' in general ?

SDBX4 : Dans une interview récente pour la Semaine Digitale, Emmanuel Davidenkoff, expert de l'éducation, a martelé le message suivant : « le numérique ne supprimera pas les enseignants ! ». Il dit aussi « La machine ne remplacera jamais l'envie d'apprendre ». Est-ce aussi votre avis ? Comment rassurer les parents et les enseignants ?

Paul Gibson: C'est très important. La technologie doit être vue et utilisée comme un soutien aux enseignants. Cela ne peut pas, à l'heure actuelle, être utilisé à leur place. Mais, ça devrait être utilisé pour réduire le temps que les enseignants passent à exécuter des tâches fatigantes, répétitives, faciles à automatiser et se dégager plus de temps pour eux à interagir avec leur élèves. Cet échange avec l'élève est ce qui se distingue entre de bons enseignants et les moins bons. Les enfants peuvent discuter directement avec la technologie, à travers les jeux et les vidéos par exemple, mais nous sommes loin d'avoir un système électronique qui peut faire ça aussi bien qu'un professeur humain en train de résoudre les difficultés de chacun des élèves (dans le contexte d'une résolution des problèmes au sein d'une classe entière).

This is very important. Technology must be seen and used as a support to teachers.

It cannot, currently, be used to replace teachers. However, it should be used to reduce the amount of time that teachers spend performing boring, repetitive, easy-to-automate tasks and free up more time for them to interact with their students. This student interaction is what distinguishes between good teachers and the not-so-good.

Children can interact directly with the technology, through games and videos for example, but we are far from having an electronic system that can do as well as a human teacher when responding to an individual's needs (within the context of meeting the needs of a whole class).

SDBX4 : La ville de Bordeaux fait de l'apprentissage des langues un thème clé dans sa nouvelle approche de l'e-éducation. Le numérique peut-il faciliter le rapport de nos enfants à la langue étrangère ?

Paul Gibson : Apprendre les langues étrangères à un jeune âge est très important. (C'est pourquoi je n'aimerais pas voir l'apprentissage du code remplacer l'apprentissage des langues étrangères). La technologie moderne est une grande assistance pour parler couramment une nouvelle langue.

Learning foreign languages is very important at a young age. (This is why I would hate to see the teaching of coding replace the teaching of foreign languages.) Modern technology is a great help in being fluent in a new language.

SDBX4 : Vous qui travaillez dans les secteurs des données, des algorithmes, des réseaux, on entend beaucoup parler de datas, et on va beaucoup en entendre parler lors de la SDBX. Pouvez-vous donner une définition ?

Paul Gibson : La data est fait d'éléments non-structurés qui ont besoin d'être traités. Quand la data est traitée, elle devient donc structurée de manière à en faire une ressource utile dans un contexte ou une présentation spécifique, et à ce moment-là, ça devient une information. Les Big Data se réfèrent au fait que dans nos vies de tous les jours, nous produisons des masses et des masses de données nous concernant et concernant le

monde qui nous entoure. En raison des avancées technologiques, cette donnée est aujourd'hui stockée. La question est comment cette donnée peut-elle devenir une information utile (et qui avantage qui). Les avantages potentiels sont énormes, mais il y a de gros risques concernant la vie privée et la sécurité.

Data is unstructured facts that need to be processed.

When data is processed it becomes structured so as to make it useful within a specific context or presentation, at which point it is called information. Big data refers to the fact that in our every-day lives we are generating masses and masses of data about ourselves and the world around us. Due to advances in technology, this data is now being stored. The question is how this data can be turned into useful information (and to whose benefit). The potential advantages are enormous, but there are enormous risks for privacy and security.

SDBX4 : Quels sont les principaux enjeux du Big Data selon vous ?

Paul Gibson : Je me concentrerai sur les questions concernant le secteur de l'éducation. Avec des logiciels pédagogiques, il y a un potentiel dans la collecte de data concernant des millions d'élèves plutôt que seulement des centaines ou des milliers (voire moins). Cette réaction peut être critique dans les outils et techniques d'évaluation et d'amélioration de la formation. La bonne donnée pourrait aider à guider la politique pédagogique dans la stratégie (dans la meilleure direction). Cependant, un risque – dans une telle situation – est que quiconque contrôle la donnée peut contrôler la politique. D'un point de vue individuel, les datas pourraient être rassemblées dans l'ordre pour forger un modèle puissant de compréhension du monde (ou d'un sujet en particulier) des étudiants. Cela peut mener à des environnements d'apprentissage adaptés installés pour répondre aux besoins de chacun des élèves de différentes façons.

Cependant, le risque d'une telle approche est que nous ajouterions de l'inégalité dans le système éducatif plutôt qu'à nous attarder à ce sérieux problème. Un autre risque serait que

cette data doit être tenue en toute sécurité.

I will focus on the issues within the educational domain. With educational software, there is a potential for collecting data concerning millions of students rather than just hundreds or thousands (or less). This feedback can be critical in evaluating and improving teaching tools and techniques. The right data could be helped to drive education policy and strategy (in the best direction). However, one risk - in such a situation – is that whoever controls the data can control the policy. From an individual point of view, data could be gathered in order to build up a rich model of a student's understanding of the world (or a particular domain/topic). This can lead to adaptive-learning environments which are fitted to meet the needs of each student in different ways.

However, the risk with such an approach is that we will add to the inequality in the education system rather than addressing this major problem. Another risk is that this data must be held in a secure manner.

