
Derrière la technique, la science informatique

Sacha Krakowiak¹, Jacques Mossière², Jean-Pierre Verjus³

1. Professeur émérite, Université Joseph Fourier
sacha.krakowiak@inria.fr

2. Professeur émérite, Grenoble INP
jacques.mossiere@grenoble-inp.fr

3. Professeur honoraire, Grenoble INP
jean-pierre.verjus@inria.fr

RÉSUMÉ. Dans le grand public, et même dans des milieux relativement informés, l'informatique est encore rarement perçue en tant que science, au-delà de ses aspects techniques. Étant donné la place croissante de l'informatique et plus généralement du numérique dans l'économie et la société, cet état de fait est regrettable. Il conduit en effet à une mauvaise perception de la nature et de l'impact de la recherche en informatique et à une désaffection des jeunes pour des études dans cette discipline, pourtant créatrice d'emplois.

Comment renverser cette tendance ? Nous pensons qu'une présentation de l'informatique dans le cadre d'un musée virtuel peut y contribuer, par deux voies d'approche : (1) montrer que les applications bien connues de tous reposent sur des notions fondamentales, bases d'une démarche rigoureuse de conception et de garantie de qualité ; (2) montrer comment ces notions se sont dégagées au cours de l'histoire. Une présentation virtuelle, par la souplesse et l'interactivité qu'elle permet, semble bien adaptée à ces objectifs.

ABSTRACT. The general public, even including well-informed circles, is seldom aware of the science that underlies information technology, beyond its purely technical aspects. This is unfortunate, considering the increasing role of information processing, and, more generally, of digital techniques, in our economies and societies. This attitude leads to a wrong perception of the nature and impact of computer science research, and to a reluctance of the young to engage in computer science studies, in spite of the high demand for trained personnel in this area.

How to reverse this trend? We believe that a presentation of computer science in a virtual museum can contribute to this goal, using two approaches: (1) showing that a number of familiar applications rely on fundamental concepts, which provide a base for rigorous design principles and for guarantees of quality; and (2) showing how these notions have emerged in the course of history. A virtual presentation, through the flexibility and adaptability it allows, seems to be well adapted to these goals.

Vers un Musée de l'Informatique et de la société Numérique en France ?

1. Introduction

L'image courante de l'informatique se réduit trop souvent à celle d'une technique. Plusieurs facteurs expliquent cet état de fait. La « numérisation du monde », évidente dans presque tous les domaines de la vie courante, contribue paradoxalement à cette situation en renvoyant l'image d'une collection d'avancées certes spectaculaires mais réduites à leur aspect technologique immédiatement perceptible et mis en évidence par les médias. L'absence, jusqu'ici, de la science informatique parmi les disciplines enseignées dans le secondaire ne fait que renforcer cette vision. Il faut enfin remarquer que cette réticence à admettre l'existence d'une science informatique ne se limite pas au grand public : il a fallu attendre 2003 pour que l'informatique entre enfin à l'Académie des Sciences sous la forme d'une section « Sciences mécaniques et informatiques », 2009 pour qu'une chaire d'accueil « Informatique et sciences numériques » soit créée au Collège de France et 2012 pour que l'informatique entre au lycée comme discipline à part entière.

Une telle situation est dommageable et il y a plusieurs raisons pour tenter de la modifier.

– la place croissante du numérique dans l'économie et la société impose une large information du public sur tous les aspects de ce domaine ;

– l'innovation dans les nouvelles technologies, moteur de la croissance, repose sur la maîtrise de l'informatique et des sciences du numérique et donc sur la qualité de la recherche dans ces disciplines ; il faut aider à la bonne perception de la nature et de l'importance de cette recherche ;

– la crise des vocations scientifiques chez les jeunes (et spécialement chez les jeunes filles) touche particulièrement l'informatique, domaine où de nombreux emplois qualifiés attendent pourtant d'être pourvus ; une meilleure vision de ce qu'est l'informatique, du type d'activités qu'on peut y mener et de ses rapports avec les autres sciences pourrait changer cette attitude.

Dans cet article, nous proposons une voie d'approche, parmi d'autres, pour contribuer à mieux faire connaître l'informatique en tant que science. Cet effort s'inscrit dans un mouvement plus global visant à améliorer la culture scientifique du public, objet actuel de nombreuses initiatives.

Nous proposons d'introduire les aspects scientifiques de l'informatique dans le cadre d'un musée virtuel (Aconit, 2000) consacré à tous les aspects de cette discipline. Loin d'une présentation *ex cathedra*, nous pensons qu'il faut partir des objets et des applications familiers au public pour tenter d'expliquer les concepts scientifiques qui leur sont sous-jacents. Nous pensons aussi qu'une organisation de

musée virtuel, par la multiplicité de points de vue et l'interactivité qu'elle autorise, est bien adaptée à une présentation attrayante.

Dans la suite de cet article, nous présentons d'abord les principes d'une approche muséale de la science informatique. Nous montrons ensuite, sur des exemples, comment ces principes peuvent être mis en œuvre à divers niveaux de compréhension. Nous indiquons enfin les grandes lignes d'une approche historique de l'évolution des concepts et des techniques de l'informatique.

2. Introduire la science informatique : principes d'une approche muséale

Comment un musée de l'informatique peut-il contribuer à une meilleure appréhension de l'informatique comme science ? Pour répondre à cette question, il faut d'abord rappeler ce qu'est la science informatique.

L'objet de l'informatique est le traitement de l'information. Le volet technique de l'informatique a pour objet la réalisation de ce traitement par des procédés automatisés. La science sur laquelle repose ce traitement comporte quatre concepts fondamentaux (Dowek, 2011) : information, algorithme, machine, langage. Ces concepts doivent être pris au sens large : ainsi une « machine » n'est pas nécessairement un ordinateur, mais peut être tout dispositif capable de réaliser des actions selon un mode opératoire défini : des exemples en sont un robot, un réseau, voire une machine virtuelle (une machine réelle dissimulée sous une couche de logiciel), ou tout simplement un boulier.

Une science ne se définit pas uniquement par son objet et ses concepts ; elle est également identifiée par sa démarche, par les méthodes qu'elle met en œuvre. De ce point de vue, l'informatique occupe une place originale parmi les sciences (Dowek, 2012). La démarche informatique (Wing, 2008) est caractérisée, entre autres, par le rôle central de l'abstraction et de la décomposition, par la séparation des préoccupations, par la modélisation et l'analyse, par la complémentarité et l'interaction entre théorie et expérience. On parle de « sciences numériques » à propos de la transformation de nombreuses disciplines par l'adoption de la démarche informatique et des outils conceptuels et techniques qui lui sont associés : physique, chimie, biologie, économie, linguistique, pour n'en citer que quelques-unes, présentent ainsi un nouveau visage. Plus généralement, c'est la société entière qui, par la large diffusion des objets et des applications informatiques dans les activités de tous les jours, devient « numérique ».

Comment introduire la science informatique par une approche muséale ? Une présentation abstraite *ex abrupto* a peu de chances de retenir l'attention. Il faut au contraire partir de notions familières au visiteur : les objets informatiques, aujourd'hui omniprésents, et les applications, largement connues de tous. Qui n'a pas utilisé un téléphone portable, consulté des informations sur le web ou fait des achats en ligne ? Et même l'utilisateur d'une automobile ou d'une simple machine à laver fait exécuter à son insu des programmes informatiques.

Nous faisons donc le choix d'une démarche inductive. À partir d'une application ou d'un objet usuel, nous amenons le visiteur à voir l'envers du décor pour dégager les grandes notions de base. Nous montrons ensuite l'universalité de ces notions, le fait qu'elles sous-tendent une variété d'objets ou de situations. Et nous invitons les visiteurs qui le souhaitent à aller un peu plus loin en leur faisant découvrir, toujours à partir de notions connues, comment est assuré le bon fonctionnement (efficacité, disponibilité, sécurité) d'un objet ou d'une application informatique.

Dans les sections qui suivent, nous montrons quelques exemples de l'application de cette démarche. Nous pensons que l'utilisation du virtuel (concept largement développé au sein même de l'informatique) offre de nombreux avantages (Aconit, 2000) pour l'élaboration des présentations : souplesse de mise en œuvre, diversité des points de vue et des parcours, capacité d'adaptation aux attentes et aux besoins des utilisateurs.

3. Introduire la science informatique : mise en œuvre

Pour illustrer la démarche que nous souhaitons appliquer, nous allons montrer comment dégager quelques notions de base de la science informatique à partir de la présentation, dans le cadre du musée virtuel, d'une application usuelle.

Cet exemple illustratif, bien que simplifié, a nécessairement un caractère technique qui peut en rendre l'accès malaisé à un lecteur non-spécialiste. C'est pourquoi, en préambule à sa description, nous en donnons une explication résumée, en termes plus accessibles.

L'application choisie est un site de vente en ligne ; nous détaillons plus loin les raisons de ce choix. Un premier temps de notre démarche peut se résumer ainsi :

- nous décrivons une version simplifiée de l'application et de la configuration matérielle sur laquelle elle est implantée, sous forme d'un ensemble de composants dont nous expliquons les fonctions et les interactions ;

- nous montrons comment donner une idée plus simple de cette structure encore relativement complexe, à travers une vue abstraite du traitement des requêtes d'un client unique ; cela nous permet d'introduire l'abstraction (et son complémentaire, le raffinement) comme un outil intellectuel permettant de maîtriser la complexité des systèmes construits.

À partir de l'analyse de l'organisation et du fonctionnement de l'application, nous dégageons quelques notions de base qui nous servent à rendre compte de cette organisation et de ce fonctionnement : information (la « matière première », support des connaissances), algorithme (un schéma pour le traitement de l'information), bases de données (mode d'organisation de l'information et outil d'accès à cette information), communication, interaction.

Rappelons que nous décrivons ici la démarche explicative que nous prévoyons de suivre dans le musée virtuel. À ce stade, nous avons donc introduit simplement plusieurs concepts de base de la science informatique et justifié leur utilité sur un

exemple. Nous montrons ensuite que ces concepts ont un caractère générique, en les appliquant à la description d'objets et d'applications informatiques variés.

Un deuxième temps (section 4) nous permet d'aller plus loin en montrant, toujours dans le cadre du musée virtuel, comment doter une application des propriétés qui la rendent effectivement utilisable : validité (elle fait ce qu'elle doit faire) ; performances (elle le fait efficacement) ; sûreté de fonctionnement (on peut lui faire confiance en toutes circonstances) ; ergonomie (elle est bien adaptée aux besoins des utilisateurs). Nous montrons que la garantie de ces propriétés est une tâche difficile pour laquelle sont requis des méthodes et outils reposant sur des avancées scientifiques. On est loin de l'image d'empirisme et de « bricolage » encore souvent attachée, notamment chez le public le plus jeune, à la perception de l'activité de développement de produits informatiques.

Nous donnons maintenant une vue plus technique de notre démarche. Rappelons que l'application choisie est celle d'un site de vente en ligne. Plusieurs raisons motivent ce choix.

- ce type d'application est largement répandu et utilisé et son interface avec les utilisateurs est relativement simple et intuitive ;
- l'architecture des sites de commerce électronique a été étudiée et raffinée depuis de nombreuses années, et son schéma d'implantation est bien connu.
- l'application est assez simple pour qu'on puisse expliquer de manière compréhensible les grandes lignes de sa réalisation, et assez complexe pour illustrer les principales notions que nous souhaitons introduire.

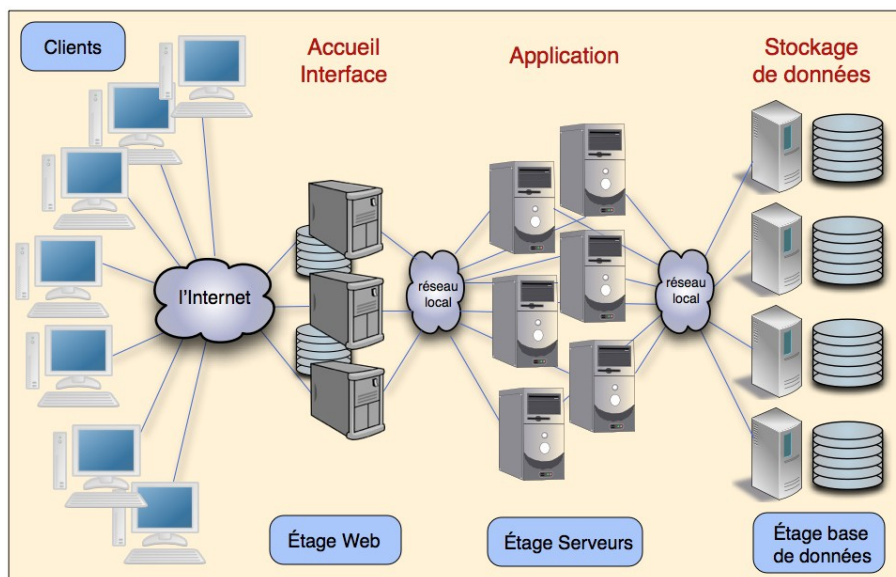


Figure 1. Un site de commerce électronique

Nous ne prétendons pas décrire un site particulier, mais un modèle générique simplifié représentatif du commerce électronique. La figure 1 montre à la fois le support matériel et l'organisation générale de l'application.

De gauche à droite, on voit sur la figure :

- les terminaux qui permettent l'accès au serveur par l'Internet ;
- les trois étages du site :

- l'étage web qui réalise l'interface présentée aux utilisateurs et le stockage des pages web qui changent relativement peu souvent (typiquement, les catalogues de produits mis en vente) ;

- l'étage des serveurs, qui met en œuvre l'application proprement dite, c'est-à-dire la réalisation des différentes fonctions accessibles aux utilisateurs (recherche de produits et de prix, commande, etc.), ainsi que la gestion interne du site (état des stocks, mise à jour des prix et produits, etc.) ;

- l'étage des bases de données, qui stocke l'ensemble des informations utilisées par l'application (état des stocks, tarifs, fichiers des clients, etc.).

On voit que chaque étage comporte de nombreux composants physiques (ordinateurs, batteries de disques), ce qui garantit à la fois les performances et la disponibilité du site (comme ce sera expliqué plus tard).

À cette vue physique, relativement complexe, on peut opposer une vue abstraite plus simple, montrant le cheminement d'une requête émise par un utilisateur (figures 2 et 3).

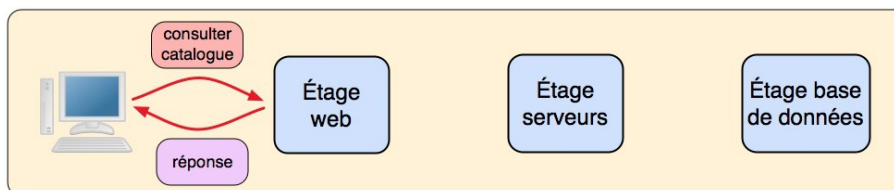


Figure 2. *Trajet d'une requête de consultation*

Sur la figure 2, on peut voir le cheminement d'une requête simple (consultation de catalogue) qui ne met en jeu que l'étage web. Le schéma est celui d'un simple échange de message par question-réponse. Le traitement se réduit à la recherche des pages du catalogue dans les fichiers locaux.

La figure 3 montre le traitement d'une demande d'achat pour un utilisateur particulier¹. Cette fois, le message de requête transite entre les différents étages,

¹ On doit expliquer ici que le système assure l'indépendance mutuelle du traitement des requêtes de ses différents utilisateurs ; c'est un aspect de la notion de transaction, développée par ailleurs. On peut alors raisonner sur une requête particulière comme si elle était seule à s'exécuter.

chacun étant le siège d'un traitement spécifique : gestion de l'interface dans l'étage web, traitement de la commande dans l'étage application, comportant une interaction avec l'étage stockage pour la consultation et la mise à jour des données (établissement de la facture, gestion des stocks). On note également une interaction avec le monde physique pour le lancement des opérations de livraison des produits commandés (dont la disponibilité a été au préalable vérifiée), pour le paiement de l'achat, et pour un éventuel réapprovisionnement.

Dans le cadre du musée, ces opérations sont présentées sous forme d'animations, avec les explications nécessaires.

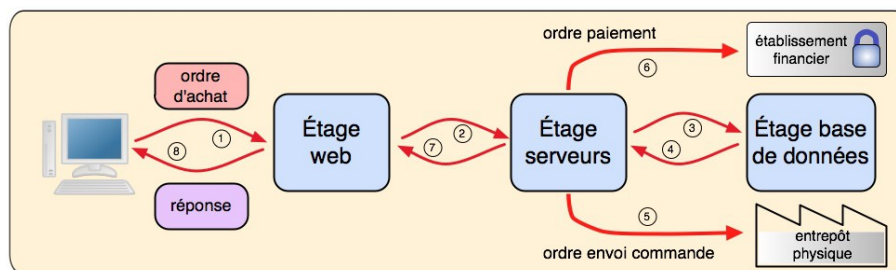


Figure 3. Trajet d'une requête d'achat

Comment à présent sont introduites les notions scientifiques ? Voici le principe de cette présentation.

– Le passage de la figure 1 aux figures 2 et 3 illustre la notion d'**abstraction**² opération intellectuelle qui joue un rôle central en informatique. On montre qu'une réalité complexe peut être modélisée par un schéma abstrait simple, sur lequel on peut utilement raisonner. Une fois défini, à ce niveau, le schéma logique des actions, une étape de **raffinement** permet sa traduction en opérations effectives réalisées par les machines physiques. Si nécessaire, on introduit des niveaux supplémentaires pour faciliter la tâche. Le raffinement s'effectue ainsi dans des vues « à la loupe » des différentes parties du système avec des grossissements de plus en plus forts, les détails de mise en œuvre étant précisés à des niveaux de plus en plus fins.

– Le traitement d'une requête est montré comme une suite d'actions élémentaires, chacune déclenchée par la précédente. Ces actions suivent un programme préétabli, mais pouvant comporter des variantes selon la situation (comme le traitement du cas où un article demandé est indisponible). On retrouve là tous les ingrédients de la notion d'**algorithme**.

² En toute rigueur, le découpage en trois étages schématisé sur la figure 1 résulte déjà d'un premier effort d'abstraction, tant pour l'aspect logique (définition des fonctions de chaque étage) que pour l'aspect physique (regroupement des machines selon leur fonction).

– Le traitement des données (catalogue, stocks, tarifs, factures, etc.) introduit le concept d'**information**, dont on peut expliquer deux aspects : la nécessité d'une organisation logique (la liste d'adresses des clients, les rubriques du catalogue, etc.), et le besoin d'un mode de stockage physique permettant la conservation et l'accès. En combinant ces deux aspects, on arrive naturellement à la notion de **base de données**.

– La communication du client avec le site, prise en charge par l'étage web, permet d'introduire la notion d'**interaction** (ici illustrée par le schéma requête-réponse). En découle la nécessité d'une **interface**, notion intuitivement familière à tout utilisateur d'un écran. Les principes d'ergonomie sont introduits dans une étape ultérieure (voir section 4).

L'application est réalisée par la collaboration de divers composants : les trois étages, et à l'intérieur de chacun d'eux, ses différents composants logiques et physiques. Ainsi est introduite la notion de **communication**, dont on peut en outre distinguer les aspects logiques (messages de requête et réponse) et les aspects physiques (transmission sur les réseaux). On retrouve la notion d'abstraction, développée ailleurs dans le musée à propos des protocoles de communication et du fonctionnement de l'Internet.

On a donc présenté, de manière intuitive, quelques notions importantes de la science informatique. On peut alors montrer que ces notions, introduites sur un exemple particulier, s'appliquent très largement à la compréhension générale du fonctionnement d'objets ou d'applications très divers. Des exemples en sont :

- un ordinateur,
- un téléphone portable,
- un robot,
- un jeu vidéo,
- une application usuelle du web, telle que Google,
- le web lui-même, en tant que support d'applications.

Les visiteurs qui souhaitent approfondir le sujet peuvent alors s'orienter vers une présentation plus systématique des notions de base de la science informatique : information, algorithme, machine, langage, dont chacune est développée en s'appuyant sur des exemples. Nous ne détaillons pas ici l'organisation de cette présentation. Le principe du traitement de l'aspect historique (comment ces notions sont progressivement apparues) est décrit dans la section 5.

4. Pour aller plus loin : rendre une application utilisable

Pour réaliser un système³ utilisable, il ne suffit pas de le construire. Il faut garantir qu'il possède diverses propriétés :

³ Nous utilisons le terme générique de « système », pour désigner aussi bien un objet matériel qu'un programme (logiciel de base ou application), ou qu'un ensemble matériel et logiciel.

- validité (le système fait-il ce qu'on lui demande ?),
- efficacité (le fait-il rapidement, en consommant peu de ressources ?),
- sûreté de fonctionnement (le système fournit-il son service en toutes circonstances ?),
- ergonomie (est-il adapté aux besoins et aux attentes des utilisateurs ?).

Nous souhaitons faire passer deux idées : ces propriétés ne se manifesteront pas « toutes seules » si on ne s'en occupe pas explicitement ; assurer leur garantie est une entreprise difficile pour laquelle est requis le secours des outils les plus avancés, tant conceptuels que techniques. La première idée n'est sans doute pas neuve pour un public familier des dysfonctionnements des produits et services qu'il utilise (transports, appareils ménagers, etc.). La seconde, en revanche, met l'accent sur la grande complexité des objets informatiques, qui fait que les difficultés rencontrées pour les rendre utilisables sont aujourd'hui encore très loin d'être maîtrisées.

Soit l'exemple de la **validité** : le système doit faire « ce qu'on veut qu'il fasse », et rien d'autre. Encore faut-il définir « ce qu'on veut qu'il fasse », et ce n'est pas si facile. Le langage naturel, encore souvent utilisé, n'est pas adapté à cette tâche car il est imprécis et ambigu. Il faut donc inventer un formalisme : modèle, langage de spécification, outils graphiques, etc. Supposant résolu le problème de la spécification (et ce n'est pas toujours le cas), il faut s'assurer que la réalisation est conforme à cette spécification. Ici encore diverses approches existent : test, construction raisonnée, preuve, dont chacune a son domaine d'application et surtout ses limites. Si nous sommes encore très loin d'une solution générale, des avancées notables ont été réalisées, qui reposent toutes sur des bases scientifiques solides : *model checking*, interprétation abstraite, assistants de preuve, dont on ne peut donner qu'une idée approchée car elles mettent en jeu des notions avancées. *A contrario*, on peut exhiber des exemples de systèmes défectueux largement popularisés en leur temps par les médias (l'échec d'Ariane 501, le *bug* du Pentium, d'autres encore), dont on a montré après coup que l'application de méthodes rigoureuses aurait pu assurer leur validité.

L'**efficacité** touche aux aspects quantitatifs des performances d'un système : temps de réponse, espace occupé, stabilité des performances, etc. C'est une qualité très directement ressentie par les utilisateurs. Garantir les performances repose avant tout sur les caractéristiques des composants matériels utilisés (vitesse, capacité de stockage, etc.), mais aussi sur l'application de méthodes de conception appropriées (raccourcir les circuits, bien utiliser l'information disponible, etc.) et sur l'usage d'outils de modélisation perfectionnés comme les modèles probabilistes ou les réseaux de files d'attente. Ces problèmes d'efficacité seront également illustrés sur des applications courantes : encore le site de vente en ligne, les jeux vidéo, un ordinateur, Google, etc.

La **sûreté de fonctionnement** garantit qu'un système assure « au mieux » sa mission en toutes circonstances, y compris en cas d'événement imprévu : défaillance, pic de charge, attaque malveillante. Le public est sensibilisé aux problèmes de pannes, de virus, de violation de confidentialité, d'usurpation

d'identité et autres situations indésirables. Ici encore, une approche scientifique commence à donner des résultats, même si on est encore très loin de maîtriser tous les problèmes. On peut ainsi montrer quelques avancées comme la cryptographie à clé publique, les techniques de tatouage, l'utilisation raisonnée de la redondance, les systèmes adaptatifs.

Dans le musée, ces aspects seront illustrés par des exemples de systèmes bien connus comme l'Internet, Google, des systèmes embarqués critiques, etc. L'exemple du site de vente en ligne, présenté plus haut, servira à illustrer la notion de transaction, outil (invisible) largement utilisé dans toutes les applications grand public, ainsi que l'usage de la redondance pour assurer la disponibilité du service.

L'**ergonomie**, adaptation du système aux besoins et aux attentes des utilisateurs, est sans doute l'un des aspects les plus familiers au public. Ici encore, les contre-exemples ne manquent pas. On mettra en évidence quelques principes de conception souvent violés : le choix des « bonnes » options par défaut en fonction des attentes exprimées ; l'uniformité (présentation et actions analogues pour fonctions analogues) ; la réutilisation à bon escient des informations disponibles ; la prévisibilité, etc.

En conclusion de cette section, nous adoptons pour les aspects étudiés ici la même démarche inductive que pour l'introduction des notions de base. Il reste à adapter le mode de présentation à la variété des situations qui viennent d'être évoquées.

5. Une approche historique

Comment les notions introduites jusqu'ici se sont-elles progressivement dégagées ? Une voie d'approche pour la compréhension des concepts est l'histoire de leur formation. Cette vision historique a plusieurs avantages :

- elle met en évidence le contexte dans lequel telle innovation conceptuelle ou technique est apparue, en réponse à un besoin, à la suite d'une intuition de génie, ou encore comme cristallisation d'idées encore « dans l'air » ;

- elle montre les étapes de l'évolution depuis les origines jusqu'à l'état actuel, et conduit à s'interroger sur une évolution future. Deux aspects de l'évolution nous paraissent particulièrement dignes d'intérêt :

- les rapports et les interactions entre les aspects scientifiques, techniques, industriels et sociétaux de l'informatique,

- l'histoire des impasses, des fausses-routes, des occasions manquées, qui est tout aussi instructive que celle des avancées et des succès.

- elle permet de faire connaître les grandes figures historiques de l'informatique, aspect qui a son intérêt propre ; on peut illustrer à ce propos différents modes d'avancement des connaissances et des techniques : prescience de l'évolution des besoins, capacité à capter et à concrétiser les idées non formulées, ou au contraire rupture radicale avec le mode de pensée courant, etc.

Sans entrer dans les détails, nous illustrons l'approche historique à l'aide de deux exemples schématiques de parcours prévus au sein du musée virtuel.

Un premier exemple est la notion d'**algorithme**, concept central de l'informatique. Les premiers algorithmes sont apparus dès lors qu'on a cherché à décrire rigoureusement un mode opératoire (initialement, un calcul) ayant un caractère générique. On cite ainsi, parmi les premiers algorithmes connus, celui inventé par Euclide vers l'an -300 pour le calcul du plus grand commun diviseur de deux entiers. Cet exemple est particulièrement intéressant car la réponse aux deux questions toujours d'actualité aujourd'hui pour tout algorithme (se termine-t-il ? donne-t-il le résultat attendu ?) se formule en termes élémentaires, compréhensibles par un lycéen. Une figure importante est celle du mathématicien et astronome arabe al-Khwārisīmī, qui est le premier à aborder, dans les années 800, l'étude des algorithmes (terme dérivé de son nom) de façon systématique. Bien que de très nombreux algorithmes aient été inventés et décrits ultérieurement, surtout dans le domaine mathématique, il faut attendre les années 1930 et la création de la métamathématique par Gödel, Church, Kleene, Turing et d'autres, pour avoir une définition rigoureuse de la notion d'algorithme et le résultat d'indécidabilité du problème de la terminaison. Les travaux actuels s'attachent aux problèmes de la construction systématique et de la preuve des algorithmes, et l'on peut noter des avancées importantes dans les toutes dernières années (preuve de systèmes complexes tels qu'un processeur, un compilateur ou un noyau de système d'exploitation).

Un second exemple concerne la notion de **langage de programmation**, mode d'expression d'un algorithme pour une machine (qui peut être physique ou abstraite). L'histoire des langages est particulièrement riche (on dénombre des milliers de langages, dont seuls quelques dizaines ont eu une vie durable), et il est très instructif de montrer les motivations qui ont conduit aux différentes avancées, de même que les considérations, pas toujours rationnelles, qui ont conduit au succès ou à l'abandon de tel ou tel langage.

Ces deux exemples illustrent parfaitement l'importance du substrat scientifique, souvent ignoré ou mésestimé, qui est à la base d'avancées concrètes dont l'importance pratique est considérable. Bien d'autres parcours sont tout aussi instructifs à cet égard (les machines, l'information, les bases de données, les réseaux, la robotique, etc.).

6. Conclusion

Les idées développées dans cet article ne sont encore qu'à l'état de propositions et n'ont pas été validées par l'expérience. Elles seront mises en œuvre dans les prochains mois, au cours de la construction d'une première version d'un musée virtuel de l'informatique. Nous sommes conscients des difficultés de cette entreprise, qui nous paraissent être de deux ordres.

– Difficultés conceptuelles. Le choix de l’approche inductive (partir d’applications et d’objets connus pour introduire les concepts scientifiques) est un pari. La définition des différentes étapes et leur articulation demanderont sans doute une mise au point après évaluation. Enfin, comme dans toute démarche de vulgarisation, l’équilibre devra être maintenu entre rigueur scientifique et simplification didactique.

– Difficultés pratiques. La présentation des exemples devra être suffisamment attractive pour ne pas décourager la poursuite des visites. Cela implique une élaboration rigoureuse de la mise en images des exemples et de la progression des explications, notamment par l’utilisation bien pensée des supports visuels et auditifs. Nous aurons besoin du concours de spécialistes de ces techniques.

Nous pensons que notre démarche présente un caractère original, car les musées virtuels existants pour l’informatique sont essentiellement articulés autour de collections d’objets, les présentations mettant surtout l’accent sur les aspects techniques. D’un autre côté, il existe des présentations des concepts scientifiques de l’informatique, mais elles ne sont généralement pas orientées vers le grand public.

Une fois réalisée une première version de notre présentation, nous pensons l’expérimenter auprès de publics divers : lycéens, étudiants, grand public, professionnels de l’informatique, en collaboration avec des spécialistes de la médiation culturelle et scientifique. Nous pourrions ainsi juger de la pertinence de notre approche et recueillir les informations qui nous permettront de l’améliorer.

Références

- Aconit (2012). Vers un musée virtuel de l’informatique, *Colloque « Vers un Musée de l’informatique et de la société numérique en France ? »*, Paris, novembre 2012.
- Dowek G. (2011). Les quatre concepts de l’informatique, *Didapro 4*, octobre 2011. Voir <https://who.rocq.inria.fr/Gilles.Dowek/philo.html>
- Dowek G. (2012). L’informatique dans la classification des sciences. *Séminaire « Philosophie de l’informatique, de la logique et de leurs interfaces »*, École normale supérieure, 30 janvier 2012. Voir <https://who.rocq.inria.fr/Gilles.Dowek/philo.html>
- Wing J. (2008). La pensée informatique, *Bulletin Specif n° 60*, décembre 2008. Voir <http://www.specif.org/bulletins/specif060.pdf>

Les auteurs

Sacha Krakowiak est professeur émérite à l’université Joseph Fourier (Grenoble). Il a travaillé dans le domaine des systèmes d’exploitation, du génie logiciel et des systèmes et applications répartis. Il s’intéresse actuellement à l’histoire de l’informatique et à sa vulgarisation.

Jacques Mossière est professeur émérite à Grenoble INP. Il a dirigé l’École Nationale Supérieure d’Informatique et de Mathématiques Appliquées de Grenoble (ENSIMAG) ainsi que le laboratoire de Génie Informatique de Grenoble. Il a mené des recherches en systèmes d’exploitation et systèmes répartis.

Jean-Pierre Verjus est professeur honoraire à Grenoble INP. Il a créé et dirigé deux centres de recherche d'Inria : Rennes - Bretagne Atlantique et Grenoble - Rhône-Alpes, et mené des recherches dans le domaine des langages et systèmes informatiques. Il est actuellement conseiller du président d'Inria et président du conseil scientifique d'Aconit.