Bertrand LEMAIRE

La division du travail en informatique
Un modèle original

Mémoire en Economie du Changement Technique

CNAM
DEA Economie et Gestion de l'Innovation 2003-2004

Table des matières

L'informatique est-il un cas particulier ?

Lorsque Marx écrit « Le Capital » en 1867 (le livre I du moins), il ne peut étudier que la première époque de la révolution industrielle et les modes productifs qui ont précédé, du moins de mémoire d'homme. Il cite donc les métiers de son temps et le contexte de son temps. De même, lorsque les autres théoriciens « classiques » du travail et de son organisation (Ford, Taylor, Mayol…) écrivent, nous sommes encore loin des « nouvelles technologies » et métiers induits.

Or l'organisation économique va prodigieusement changer au cours du vingtième siècle, de même (et surtout) que les techniques employées.

L'informatique est la seule industrie née *ex-nihilo* au milieu du vingtième siècle. Il n'y a pas d'artisan calculateur au Moyen-Age, pas plus que de corporation du traitement de l'information ! Bien sûr, on peut toujours assimiler telle ou telle fonction de l'informatique à tel ou tel métier pré-existant (imprimeur, moine copiste, archiviste, bibliothécaire, écrivain public…) mais aucun n'est réellement assimilable à ceux de l'informatique elle-même (analyste, programmeur, administrateur système…).

Créées alors que les autres professions ont déjà vécu la révolution industrielle, les professions de l'informatique subissent-elles directement l'organisation répartie telle qu'analysée par Karl Marx dans le chapitre XIV du livre I du Capital ? Au contraire, évoluent-elles de manière similaire ou bien différentes ?

Bases théoriques

Les trois théoriciens de l'organisation du travail par la répartition des tâches qui seront retenus ici sont Karl Marx, Frédéric Wilson Taylor et Henry Ford. Il est évident que seule l'analyse de l'existant par ces auteurs sera prise en considération, en dehors de toute implication de leurs œuvres, qu'elles soient politiques ou morales, directes ou indirectes, revendiquées ou non.

Le patronat se distingue petit à petit des producteurs

Ces trois auteurs ont décrit et expliqué l'évolution de l'organisation du travail à l'occasion des révolutions industrielles. L'élément essentiel et conducteur, commun aux trois bien que leurs orientations politiques soient pour le moins différentes, est l'appauvrissement de la tâche des travailleurs et la perte du contrôle du capital productif par ceux-ci. Autrement dit : le travailleur cesse d'être un offreur de service pour devenir un utilisateur des machines possédées par un patronat distinct de la classe laborieuse. Cette évolution constitue une rupture par rapport au modèle du Moyen Age : même si la hiérarchie sociale y était très forte, les « patrons » étaient avant tout des travailleurs eux-mêmes.

De l'artisanat à la grande industrie

Plusieurs stades ont été définis par les auteurs considérés :

1) Chaque producteur est indépendant, travaille pour son propre compte avec ses propres outils (« le bon ouvrier a de bons outils » dit encore le proverbe). A ce stade, les moyens de production sont éparpillés, la productivité faible, la spécialisation de chacun est encore plus faible. Chaque produit fini est unique.
2) Réunion de producteurs indépendants pour faciliter la production. Lors du premier stade de la « manufacture » telle que décrite par Karl Marx (La Capital, Livre I, chapitre XIV), les anciens artisans agissent au sein d'un même local et enchaînent leurs activités. De métiers distincts, leurs fonctions deviennent donc des parties d'un métier plus vaste. Karl Marx prend l'exemple de la fabrication de carrosses où se réunissent des selliers, des tailleurs, des menuisiers, etc… qui deviennent tous des fabricants de carrosses mais n'intervenant que sur un stade de la production. Les travailleurs se spécialisent petit à petit, les tâches que chacun assume sont de plus en plus

parcellaires au sein de la chaîne de fabrication. De fait, les anciens artisans perdent leur autonomie et deviennent pleinement « salariés », donc soumis à une subordination hiérarchique. La connaissance du métier reste un élément fondamental de la productivité individuelle.

3) Le deuxième stade de la manufacture suppose une répartition des tâches la plus parcellaire possible. Simultanément, l'emploi de machines suppose la concentration d'un capital nécessaire à de tels investissements. Le « métier » cesse donc de nécessiter une qualification importante. Nous passons au stade de la « grande industrie » où la combinaison capitalistique a plus d'importance que le savoir-faire des personnels employés.

4) L'organisation scientifique du travail, théorisée par Frédéric Wilson Taylor puis par Henry Ford, est le dernier stade de l'usine classique. La répartition maximale des tâches implique que le travailleur ne possède plus de vue globale de son métier. Le savoir professionnel lui échappe au profit d'une part de la direction de l'usine, d'autre part de services spécialisés dans la conception. Enfin, l'ouvrier n'a plus d'initiative : sa direction ne lui indique plus de missions mais des tâches (ou mieux : une tâche) précises qu'il doit se contenter de répéter. Cette répartition des tâches prédéfinies est liée à la création des chaînes de production et donc à la standardisation des produits finis. L'ouvrier ne disposant plus de qualification, il devient aisément interchangeable.

Naissance de l'informatique

Selon le dictionnaire Le Robert (édition 1993 - extraits) : l'« informatique [est l'] ensemble des techniques de la collecte, du tri de la mise en mémoire, du stockage, de la transmission et de l'utilisation des informations traitées automatiquement à l'aide de programmes mis en œuvre sur des ordinateurs ». La notion d'ordinateurs renvoie, quant à elle, à celle d'automate programmable. La machinisme est donc fondateur dans l'informatique : on ne peut pas envisager d'informatique « artisanale » car il y aurait alors contradiction dans les termes.

Même si l'informatique ne naît pas *ex nihilo* un beau jour au milieu du vingtième siècle, puisqu'elle est l'aboutissement d'un long processus d'évolution théorique et technique (abaques, bouliers, pascaline, machine de Babbage…), l'industrie informatique, elle, naît sans connaître de précurseurs. Il s'agit d'un ensemble de nouveaux métiers qui ne sont pas des évolutions de métiers anciens.

Sur le plan technique, les premiers automates programmables naissent dans l'industrie textile (les métiers Jacquart). Puis ce sera l'heure des calculateurs électro-mécaniques (tabulating machines) utilisées dans les opérations du recensement américain dès la fin du XIX^{ème} siècle. Le recensement de 1890 fut ainsi traité en six semaines grâce aux machines de Hermann Hollerith. Celui-ci créa la Tabulating Machine Company qui devint en 1924 l'International Business Machines Company (IBM).

Sans rentrer dans les détails de la naissance effective de l'informatique, sujet de maints débats, ce n'est qu'au cours des années 1940 que naît véritablement l'informatique. Le premier calculateur réellement programmable non-spécialisé sur une tâche précise est sans doute l'Eniac (Electronic Numerical Integrator And Computer), achevé en 1945, deux ans après l'appareil qui est considéré généralement comme le premier ordinateur (bien qu'il n'en ait pas toutes les caractéristiques), le Mark I.

Le caractère essentiel de l'ordinateur est, au delà d'être un automate capable de traiter des données, d'être programmable, ce de manière variable. Le premier « informaticien » est, de ce point de vue, une informaticienne : Augusta Ada Byron, comtesse de Lovelace. Celle-ci écrivit, de manière très théorique, des programmes pour la machine de son grand ami Charles Babbage. Pour lui rendre hommage, un langage de programmation fut nommé « Ada ».

Les débuts de l'industrie informatique

L'industrie informatique se développe à partir de celle, pourtant récente à l'époque, des machines électromécaniques. Ses initiateurs sont connus : IBM, bien sûr, mais aussi Bull, Burroughs, Remington, Univac, Sperry (ces quatre dernières ont progressivement fusionné pour donner Unisys),…

Les systèmes centraux

Très innovante, l'industrie informatique nécessite de très lourds investissements. La recherche fondamentale reste universitaire. Et tous les premiers ordinateurs (au sens propre du mot) seront conçus dans des institutions publiques (office du recensement américain pour les tabulateurs, armées américaine ou anglaise pour les ordinateurs).

D'abord domaine exclusif de chercheurs, l'informatique entre ensuite dans les entreprises pour faire ce qu'elles savent faire de mieux : compter. Autrement dit : réaliser la comptabilité. Le coût considérable des machines les réserve aux très grandes entreprises. Les services informatiques sont donc centralisés et entre les mains de spécialistes hautement qualifiés.

Autour de l'outil, les professions sont multiples, toutes détentrices d'un savoir précis : analystes (qui conçoivent les chaînes logiques de traitement des informations), les programmeurs (qui convertissent les analyses en programmes ou *logiciels*), les administrateurs (qui font marcher l'ensemble)… jusqu'au moins qualifié : le pupitreur.

Au contraire des autres métiers, l'analyse de Karl Marx ne s'applique donc pas à l'informatique lors de sa naissance. Bien que hautement capitalistique, l'informatique échappe à l'organisation scientifique du travail et à la déqualification des travailleurs, malgré qu'il y ait division des tâches.

Au départ, les programmeurs doivent écrire les logiciels en utilisant le langage machine (suite d'instructions directement interprétables par la machine) voire le binaire (suite de 0 et de 1, c'est-à-dire codage strict des ouvertures et des fermetures des circuits électriques de la machine).

La programmation : du simple texte

En 1954, la naissance du premier langage de programmation (le Fortran, ou FORmula TRANslator) va faciliter le travail des informaticiens. Les langages vont se succéder et être de plus en plus complexes.

Désormais, les informaticiens n'ont plus à se préoccuper des réalités physiques, c'est-à-dire des circuits électriques des machines employés. Ils peuvent concevoir les logiciels sous la forme de simples textes écrits dans un langage particulier mais intelligible par un humain, avec une syntaxe et une orthographe précises.

De ce fait, on peut considérer que la « qualification » des informaticiens baisse puisque seuls ceux qui conçoivent les compilateurs (c'est-à-dire des programmes servant à convertir des textes écrits en langages de programmation en véritables logiciels utilisables) connaissent réellement les machines. Ainsi, on assiste à une distinction entre d'une part des concepteurs de machine ou de compilateurs (assimilables aux « bureaux d'études » de Taylor) et d'autre part des « informaticiens d'entreprise » (ou de sociétés de services) qui écrivent des logiciels pour les besoins de leurs employeurs ou administrent au quotidien des systèmes informatiques.

Mais, au contraire de l'industrie traditionnelle, cette distinction études/réalisations (en fait deux niveaux différents d'études) ne va pas de pair avec une division croissante du travail ni avec une maîtrise des métiers par la direction des entreprises. Au contraire, analystes et programmeurs deviennent des analystes-programmeurs, réalisant les deux types de tâches…

Au contraire des autres secteurs, et de l'analyse traditionnelle, l'informatique échappe donc à l'organisation scientifique du travail, notamment à la parcellisation des tâches, ce malgré la mobilisation importante de capitaux nécessaires à son exercice.

La manufacture informatique

La parcellisation a finalement lieu

Progressivement, la programmation va devenir modulaire, notamment avec les langages « objet » (C++…). Les programmes vont donc pouvoir être écrits par certains et réutilisés par d'autres dans des logiciels n'ayant rien à voir (sous réserve des problèmes de propriété industrielle, bien entendu).

De plus, les ordinateurs possèdent de plus en plus de « couches logicielles » distinctes : système d'exploitation, interface graphique, serveur d'application… Chacune, dans cet ordre, apparaît comme une nouvelle « machine » qui réalise certaines tâches et qui sera exploitée par les programmes écrits pour elle. On assiste ainsi à une sorte de poupée russe : le programme écrit par l'informaticien en entreprise ne s'exécute plus sur une machine physique mais sur une machine logicielle qui, elle-même, s'exécute sur une autre machine logicielle, et ainsi de suite jusqu'à la machine physique.

Il en résulte donc bien une parcellisation des tâches mais qui échappe à l'employeur de l'informaticien en entreprise utilisatrice : le pouvoir de décider de la parcellisation est du côté des fournisseurs d'ordinateurs et de « machines logicielles », malgré le fait que l'informatique nécessite pour l'entreprise (ou l'administration) utilisatrice de très lourds investissements.

Autre paradoxe, cette parcellisation ne s'accompagne pas d'une déqualification mais au contraire d'une requalification. De plus en plus complexe, l'informatique, devenue « système d'information » échappe de plus en plus au non-spécialiste…

Désappropriation de l'outil

L'une des caractéristiques de la manufacture est que l'ouvrier cesse d'être propriétaire de son outil de travail. De fait, en informatique, l'ordinateur n'appartient (du moins au début) jamais à l'informaticien : son coût est trop énorme.

En matière logicielle, la désappropriation est progressive. Au départ, l'informaticien écrit ses programmes *ex nihilo* lui-même. Puis, il écrit au travers de langages de programmation. Enfin, il utilise des « modules » qu'il n'a pas conçu. Dans certains cas, il n'aura pas à utiliser ce langage mais définira dans des AGL (ateliers de génie logiciel) les procédures à programmer, l'AGL se chargeant d'écrire les lignes de programme induites. On retrouve ainsi quasiment la distinction entre analyste et programmeur où ce dernier est devenu une machine logicielle.

Le stade ultime est le « framework » : un ensemble d'outils et de modules packagés permettant à un informaticien, même médiocre, de réaliser rapidement des programmes extrêmement complexes. L'exemple typique est *Microsoft .Net*.

La parcellisation des tâches aboutît donc enfin à une déqualification.

Il y a cependant une différence importante avec le phénomène observé dans l'industrie traditionnelle : la propriété des « machines logicielles » échappe totalement aux entreprises utilisatrices au profit des « éditeurs de logiciels ».

La réaction à la manufacture informatique

La micro-informatique

Avec la micro-informatique (née au début des années 1970), l'outil peut être acheté par n'importe quel travailleur puis, dès la fin des années 1980, par n'importe quel amateur. Le « capital informatique » échappe ainsi de plus en plus aux services dédiés dans les entreprises mais aussi aux entreprises elles-mêmes.

A la déqualification des informaticiens, répond alors la qualification des cols blancs qui, tous, plus ou moins, se mettent à utiliser l'outil informatique puis à la programmer eux-mêmes grâce aux multiples outils mis à leur disposition. L'outil qui séduisit les contrôleurs de gestion fut ainsi le tableur. Lorsque celui-ci devint programmable, bon nombre de traitements dans les ordinateurs centraux devinrent inutiles…

De même, la plupart des métiers des cols blancs se numérisent : traitement de texte, base de données programmable aisément (*Microsoft Access*…). Chaque travailleur conçoit de plus en plus ses outils.

Si les tâches se parcellisent toujours plus, c'est pour accroître la qualification des utilisateurs des technologies et non l'inverse.

Le yoyo

Ce mouvement se verra combattu. Le tandem parcellisation/qualification subit des mouvements de yoyo sur de brèves périodes. A l'heure actuelle, l'informatique se centralise et s'uniformise, au nom de la productivité et de la sécurité. Les utilisateurs se voient donc déqualifiés… L'informatique décisionnelle remplace ainsi petit à petit le bon vieux tableur.

Mais, à l'inverse, la généralisation de la micro-informatique induit que, de plus en plus, les cols blancs réalisent par eux-mêmes énormément de tâches qui étaient autrefois disjointes, notamment en matière de télécommunication (messagerie au lieu du telex) ou de secrétariat (agendas logiciels, traitements de texte…).

Le logiciel libre, en conclusion

Les informaticiens ont comme caractéristiques d'être une population à la base très qualifiée et à tendance contestataire qui ne peut se satisfaire d'être dépossédée de ses outils de travail, surtout au profit d'une entreprise qui n'est pas la leur. De plus, la micro-informatique permet à de multiples travailleurs indépendants de disposer de leurs propres outils.

Restait à se réapproprier ce qui n'appartenait pas à l'employeur : les « machines logicielles » de différents niveaux (systèmes d'exploitations, ateliers de génie logiciel, serveurs d'applications, etc…).

Né dans les université américaine dans les années 1970 (notamment à Berkeley), le mouvement du « logiciel libre », communautaire, répond à ce besoin. Même si chacun ne « fabrique » pas ses logiciels à partir de rien, la construction des « usines logicielles » est collective. De plus, chacun peut étudier et modifier les constituants « open-source ». Le décollage de ce type d'informatique est néanmoins récent.

L'informatique aboutit donc, par ce mouvement original, à la « propriété collective des moyens de production » !

Du moins, en ce qui concerne la partie logicielle de l'informatique.