

# Boite noire ou boite transparente ?

**Anne BRYGOO**

Université Paris 6, UFR d'Informatique,  
Tour 55-65, porte 309,  
4 place Jussieu, F 75252 Paris Cedex 05  
Tél +33 1 44 27 58 77  
Fax +33 1 44 27 62 86  
Email : abrygoo@ccr.jussieu.fr

## RESUME

Le principe de la boîte noire, brique élémentaire dans l'élaboration des programmes, imprègne fortement notre enseignement d'informatique. C'est un outil efficace pour la construction des systèmes informatiques complexes. Mais, ne doit-on pas, dans les enseignements d'informatique destinés soit à des futurs informaticiens soit à des non-informaticiens, essayer d'introduire une certaine transparence de ces boîtes noires ? Cette transparence donnerait des éléments pour comprendre la logique interne de fonctionnement des logiciels tant aux utilisateurs de ces logiciels qu'aux citoyens pour qui ces logiciels interfèrent dans leur vie quotidienne.

## 1. INTRODUCTION

Je partirais d'une triple expérience :

- l'une d'enseignante chargée d'initier à l'informatique des étudiants en DEUG scientifique (2<sup>o</sup> année après le baccalauréat français);
- l'autre de formatrice à l'utilisation de progiciels (traitement de textes et tableurs) pour des non-informaticiens;
- et enfin une troisième expérience d'enseignante en "Informatique et Société" dans le cadre de la maîtrise d'informatique (4<sup>o</sup> année après le baccalauréat français).

Ces enseignements se font au sein d'équipes pédagogiques. Ces trois formations sont très différentes tant par leurs objectifs que par leur contenu. Il est cependant possible de les confronter et de tirer partie des critiques que je formule sur ma pratique d'enseignante dans chacune d'elles pour tenter de progresser dans l'apprentissage tant de l'informatique-discipline que de l'informatique-outil et de proposer de nouvelles fonctionnalités à certains logiciels.

## 2. INITIATION A L'INFORMATIQUE EN TANT QUE DISCIPLINE

### 2.1. Noyau incontesté

Traditionnellement l'enseignement d'initiation à l'informatique, en tant que discipline, a toujours contenu un noyau "dur", incontesté, plus ou moins développé comportant la

présentation de la structure de l'ordinateur, de son fonctionnement, de la représentation des informations, de la nécessité de la programmation, de l'existence des différents langages et des styles de programmation. Nous retrouvons ce noyau dur dans tous les enseignements d'informatique des premiers cycles scientifiques [SPECIF 90]. Au delà les repères s'estompent, les pratiques divergent.

## 2.2. Programmation

Enseigner la programmation ? Certes nous le faisons. Nous apprenons à faire faire [DUCHATEAU] avec souvent pour contrainte l'utilisation d'un seul langage de programmation. Notre pédagogie repose sur un savoir-faire d'artisan, accumulé au fil des ans, avec des certitudes, des doutes. L'ensemble forme pour moi un tout pas toujours cohérent.

Des valeurs sûres à enseigner semblent cependant se dégager. Parmi celles-ci on trouve la nécessité de préciser les données d'entrée et les résultats souhaités pour construire le traitement permettant de passer d'un état initial à un état final ; pour ce traitement, la nécessité d'un affinement successif aboutissant à un assemblage plus ou moins complexe de briques-Lego, **boîtes noires**, au contenu qui deviendra invisible à la fin du processus de construction, ne laissant transparaître pour chacune des boîtes que leurs entrées/sorties.

L'enseignant a un autre rôle, souvent ignoré car ingrat mais bien réel, celui de noter les étudiants. Pour juger des programmes des étudiants, les critères retenus et leur pondération ne font pas l'unanimité. Ils sont révélateurs des priorités, des objectifs que se fixe l'enseignant : le programme "qui tourne", la lisibilité du programme, des commentaires, l'analyse associée, la justification du choix de l'algorithme, sa validité, sa complexité, son originalité, l'ergonomie des échanges utilisateurs-machine,...

## 2.3. Algorithmique ou processus d'informatisation

La palette des sujets proposés aux étudiants est vaste. Leur formulation peut aller de l'énoncé d'un problème quasi formalisé permettant de se concentrer presque uniquement sur l'algorithme, à l'énoncé prenant appui sur des situations du quotidien, simples mais déjà chargées d'implicites démontrant l'importance de l'analyse.

En effet, dans un temps nécessairement limité, l'enseignant doit faire des choix. Il se trouve face au dilemme suivant :

- privilégier l'algorithmique, valeur sûre et qui de plus est nécessaire mais non suffisante, pour être reconnu socialement par les confrères faisant des formations "supérieures" en informatique-discipline,
- favoriser l'étude de situations concrètes comme moyen de faire sentir la nécessité d'une analyse préalable à toute informatisation, réflexion nécessaire mais peu valorisée.

De façon un peu caricaturale, nous voulons faire de l'algorithmique mais en fait nous ne faisons généralement que de la programmation, sans référence aux utilisateurs potentiels et sans vraiment étudier le processus d'informatisation dans son ensemble.

## 3. FORMATION A L'INFORMATIQUE EN TANT QU'OUTIL

---

La formation de non-informaticiens à l'utilisation de progiciels grand public, tels que les traitements de texte et les tableurs, est souvent considérée comme une formation à l'informatique-outil. Cette formation me semble cependant être vue, plus comme une initiation à l'utilisation d'un **instrument** que d'un **outil** au sens présenté par J.P.POITOU [92]. L'instrument est considéré comme approprié et exploité en vue d'une fin immédiate alors que l'outil est, lui, considéré comme "*inscrit dans un processus de production et adapté dès sa conception, à une classe spéciale d'opérations, elle-même insérée dans une certaine catégorie de chaînes opératoires*". L'outil est "*un dépôt de savoir, et lorsque les conditions de sa mise en œuvre sont réunies, il transforme ces savoirs en action*".

En effet dans le cadre des formations à l'utilisation de progiciels nous travaillons sur un objet, certes complexe mais sans réelle référence ni au passé ni au processus informatisé. Souvent dans cet enseignement nous retrouvons, comme pour l'initiation à l'informatique-discipline, la structure de l'ordinateur, la nécessité pour un traitement spécifié de préciser les données et les résultats souhaités mais, entre données et résultats, l'utilisateur ne voit qu'une boîte noire, une fonctionnalité. Le logiciel étudié est présenté comme un assemblage structuré de boîtes noires. Il n'est pas question de programmation ni même d'algorithmique. L'utilisateur devra, bon gré mal gré, s'adapter, construire sa propre représentation de l'ensemble des boîtes noires et tant pis si sa logique d'utilisation ne coïncide pas avec la logique de fonctionnement [BARTHET 88].

Nous avons cependant cette année, dans le cadre d'un nouvel enseignement d'informatique-discipline en DEUG, essayé d'intégrer une approche informatique-outil, ce qui n'était pas le cas auparavant. Notre hypothèse est qu'il est possible en formant à l'utilisation de progiciels (savoir directement utilisable quand l'étudiant parvient à entrer dans la vie active) d'introduire des concepts de bases de l'informatique, des notions d'algorithmique et de programmation. Ceci est rendu possible par l'étude des fonctions générales du progiciel mais aussi par la distinction entre les rôles respectifs de l'utilisateur et du logiciel. L'utilisateur introduit du texte, sélectionne des commandes et des paramètres pour ces commandes et le logiciel fournit des résultats. L'étude (simplifiée) du contenu de certaines boîtes noires permet de prendre conscience que derrière chaque commande il existe un ou plusieurs algorithmes.

Cette expérimentation permet d'aborder le logiciel non comme un instrument mais comme un outil porteur de savoir mais elle le fait sous l'angle de la logique de fonctionnement et non sous celui de la logique d'utilisation à savoir outil dans un processus de production complètement extérieur à l'informatique-discipline.

#### **4. FORMATION A "INFORMATIQUE ET SOCIETE"**

Nous allons maintenant aborder un autre volet de notre enseignement, "Informatique et Société" où l'expérience que j'ai acquise montre la nécessité d'ouvrir ces fameuses boîtes noires et les possibilités pour cela, que semblent offrir les nouveaux développements de l'informatique.

##### **4.1. L'enseignement d'"Informatique et Société"**

Cet enseignement a pour but d'apporter aux étudiants d'informatique un ensemble de connaissances et de susciter leur réflexion sur le rôle et les responsabilités qu'ils auront à assumer en tant que futurs professionnels. Cet enseignement doit permettre de situer le développement et l'emploi de l'informatique dans un contexte juridique, économique, social et culturel.

#### 4.2. Apports de l'étude d'"Informatique et Libertés"

Dans cet enseignement d'"Informatique et Société" une partie concerne l'étude d'"Informatique et Libertés" à savoir l'étude des menaces qu'un traitement automatisé d'informations nominatives peut faire peser sur la vie privée et les libertés des citoyens, les principales dispositions de la loi française du 6/1/78 relative à "l'Informatique, aux fichiers et aux libertés" et les difficultés d'application de cette loi.

Cette étude montre la difficulté pour chaque citoyen de connaître tous les **traitements** automatisés dont il fait l'objet, d'accéder aux **informations** nominatives qui le concernent et ce, bien qu'existe un droit reconnu par la loi, le droit d'accès, et encore moins "*de connaître et de contester les raisonnements utilisés dans les traitements automatisés dont les résultats lui sont opposés*" (article 3).

#### 4.3. Apports de l'étude sur l'ergonomie

Un autre champ d'étude d'"Informatique et Société" est l'étude de l'ergonomie. Historiquement il y eu d'abord le développement de l'ergonomie du matériel pour répondre aux plaintes des opérateurs, puis le développement de l'ergonomie du logiciel avec ses différents champs d'application.

Ces recherches en ergonomie ont permis des améliorations très appréciables pour le confort visuel et postural des opérateurs, une utilisation plus aisée des logiciels pour les non-informaticiens par exemple par l'apparition des menus déroulants, des fenêtres, d'une aide en ligne,....

Des recherches actuelles dans la conception de nouveaux logiciels portent sur l'intégration des logiques d'utilisations des opérateurs, sur des aides en lignes concernant non seulement l'utilisation du logiciel mais aussi les raisonnements utilisés dans le logiciel (en particulier les modules d'explications dans les systèmes experts).

Ces recherches devraient pouvoir être étendues plus spécifiquement aux traitements automatisés d'informations nominatives. Des interfaces devraient pouvoir être mises en place pour répondre aux diverses demandes d'information des "fichés". Ces demandes seraient relatives aux informations les concernant, leur durée de conservation, leur circuit d'utilisation, les traitements les utilisant, les raisonnements relatifs à ces traitements.

### 5. UNE APPROCHE INTEGREE

Le principe de la boîte noire comme nous venons de le voir, imprègne fortement notre enseignement. En informatique-discipline il s'est certes révélé être un outil efficace de

construction des systèmes informatiques devenus très complexes. Par contre en informatique-outil, il semble être un frein à la maîtrise réelle de l'outil. En "Informatique et Société" il apparaît comme un obstacle à l'exercice de certains droits du citoyen.

Ce principe est à l'évidence issu de notre société elle-même, qui pousse à l'extrême la spécialisation, la division du travail. Mais il faudrait essayer de le remettre en cause dans les enseignements qui s'adressent aux futurs informaticiens comme aux non-informaticiens et y introduire des principes de transparence. Il est à noter que ce mot de "transparence" a une signification ambiguë, soit qualité qui permet de voir la réalité intérieure de l'objet lui-même soit qualité qui permet de voir à travers, au delà de l'objet. Nous prenons ici pour sens du mot transparence la première acception.

Cette transparence concerne toute personne qui élabore des traitements automatisés, qui les utilise régulièrement ou encore tout usager impliqué dans les résultats de ces traitements. Elle doit se traduire par la possibilité d'accéder à une partie du savoir accumulé dans les logiciels, d'atteindre les raisonnements intégrés, les paramètres mis en jeu. Elle doit porter nécessairement sur la logique de fonctionnement du logiciel mais aussi sur la logique d'utilisation qui implicitement ou explicitement a guidé la conception du logiciel

Cette possibilité de transparence passe par la mise au point d'outils adéquats, adaptés au niveau de l'utilisateur du logiciel, à ses demandes et à celles des usagers. Elle passe aussi par la formation, bien sûr des utilisateurs et des usagers, mais surtout des futurs informaticiens. Pour ces derniers, l'introduction de la nécessité de la transparence des boîtes noires doit se faire dès l'initiation. Il n'y a pas de réelles difficultés d'apprentissage et cela me semble relever plus d'un état d'esprit. En effet au lieu de privilégier les seules performances techniques (programme qui "tourne", validité de l'algorithme, moindre complexité,...), des critères de communicabilité sont à mettre en avant (lisibilité du programme et des commentaires, justification du choix de l'algorithme, raisonnement traduit par l'algorithme choisi, ergonomie des échanges utilisateurs-machine,...).

La volonté de transparence, si elle est introduite dès le début de l'enseignement, peut participer à une prise en compte plus importante de l'utilisateur et de l'utilisateur. Les étudiants ainsi formés, pourront au cours de leur vie professionnelle fournir des aides, intégrés dans les logiciels, indispensables à la compréhension de ceux-ci. Ces nouvelles aides seraient fort utiles pour une réelle maîtrise des logiciels et leur intégration par les utilisateurs dans leur propre processus de production. Les citoyens disposeraient d'outils plus efficaces pour comprendre ces logiciels qui interfèrent de plus en plus dans leur vie quotidienne.

## 6. BIBLIOGRAPHIE

[BARTHET 88] M.F.BARTHET, "Logiciels interactifs et ergonomie : modèles et méthodes de conception", Paris, Dunod, 1988.

[DUCHATEAU] C.DUCHATEAU, "Algorithmique, programmation et résolution de problème"

[POITOU 92] Jean-Pierre POITOU, "Le mythe, la cathédrale, l'atelier : trois dispositifs artificiels de pensée. Essai d'anthropologie cognitive", Colloque Sciences

sociales et IA.

Technologie, idéologie, pratique, Vol 10, n°24, p67 à 85, 1992.

[SPECIF 90] "Enseigner l'informatique en tant que discipline dans les premiers cycles scientifiques", Journées SPECIF , Nantes 27-29 Mars 1990.