

Formes de l'action et architecture du système d'information

Michel Volle

29 janvier 2007

Le SI a pour vocation d'outiller l'action en apportant à l'être humain l'assistance de l'automate. Dans l'action, la volonté humaine rencontre le monde de la nature. Cette rencontre peut être soit individuelle, soit organisée lorsque la volonté est celle d'une institution comme l'entreprise. Elle prend des formes diverses selon le degré d'incertitude ou d'urgence qu'elle comporte. Le SI sera soumis à des exigences différentes selon la nature de l'action considérée. Si l'on s'est fait une idée trop rigide de ce que doit être un SI on risque de mal outiller l'action ou d'utiliser une solution trop coûteuse.

Si la couture était aussi immature que le sont les SI, il arriverait qu'un couturier étourdi utilise, pour faire une jupe, le tissu qui convient à un manteau : d'un « cahier des charges » à l'autre, il ne garderait pas en mémoire la différence entre ce qui convient à l'un et ce qui convient à l'autre.

Il est donc utile de disposer d'une classification des formes d'action et d'une description des outils que le SI doit fournir à chacune. La classification que nous proposons ci-dessous suit à peu près un ordre chronologique : elle part des premières utilisations historiques de l'informatique pour aller vers les plus récentes.

Cet ordre fait apparaître une succession d'étapes. À chacune d'entre elles a correspondu une représentation particulière de ce qu'est et peut faire l'informatique, associée à un style de programmation et à une famille de solutions d'architecture. Passer à une nouvelle étape a toujours été pénible pour les experts, qui maîtrisaient bien l'étape précédente, comme pour les entreprises qui, pour se forger une nouvelle représentation de l'informatique, ont dû surmonter un obstacle intellectuel.

1 L'automate pur

L'ordinateur a d'abord été utilisé pour réaliser des calculs (tables pour l'artillerie, simulations pour la mise au point des techniques nucléaires), puis pour automatiser la production physique¹ ; on trouve aujourd'hui des automates dans l'« informatique embarquée » (pilotage automatique des avions, commande des satellites) et aussi dans les machines dont on attend un fonctionnement automatique (commutateurs et routeurs des réseaux de télécommunications, moteurs de recherche sur le Web, robots de l'industrie). Enfin, la plate-forme informatique du SI (processeurs, mémoires, réseaux, systèmes d'exploitation, applications, postes de travail etc.) est elle-même un automate.

Les automates exécutent de façon répétitive des programmes qui peuvent être compliqués mais qui concrétisent un *modèle*, donc une représentation sélective et simplificatrice de la partie du monde de la nature sur laquelle l'automate doit agir. Il se produira donc toujours des événements que le modèle n'avait pas prévus et qui entraîneront des « surprises » : une pièce se rompt accidentellement dans une machine, ou bien la conception de celle-ci n'a pas pu tenir compte de phénomènes que l'on connaissait mal comme cela s'est produit sur les avions *Comet* de De Havilland, dont les accidents ont permis de mieux comprendre la fatigue du métal.

L'informatique elle-même est sujette à des imprévus. Un programme informatique de grande taille ne peut pas correspondre exactement au modèle qu'il est censé réaliser car il y reste toujours des *défauts*. D'après Jacques Printz² un code source contient en moyenne, lorsqu'il sort des mains des programmeurs et après avoir été « vérifié » par le compilateur, de 50 à 100 défauts par millier de lignes. Il subsiste encore après les tests, vérifications et corrections, un à deux défauts par millier de lignes lorsqu'on déploie et installe le logiciel. Si l'on suppose que Microsoft a poussé l'effort de correction aussi loin que le fait la NASA pour ses logiciels embarqués (hypothèse excessivement optimiste), le système d'exploitation

1. Dans le numéro spécial de *Science et Vie* sur l'automatisme paru en 1964, la gestion n'apparaît que comme un domaine relativement secondaire par rapport à la production physique.

2. Jacques Printz, *Architecture logicielle*, Dunod 2006, p. 73. L'un des théorèmes les plus importants de la théorie des automates, c'est qu'il est impossible de concevoir le programme qui pourrait vérifier et garantir qu'un autre programme ne comporte aucun défaut.

Vista, avec 64 millions de lignes de code, comporte de l'ordre de 6 à 400 défauts³.

Dans sa mise en œuvre l'automate est par ailleurs soumis à des *phénomènes physiques aléatoires*, les uns accidentels (rupture d'une connexion ou d'un câble, effet des intempéries ou des séismes), d'autres systématiques (altération des données provoquée par le rayonnement cosmique⁴, débordement des mémoires lorsque les limites de leur dimensionnement sont atteintes⁵).

L'action de l'automate sur la nature est donc soumise à trois contraintes :
- elle concrétise un modèle qui représente la nature de façon schématique et incomplète : des surprises se produiront ;

- le programme comporte des défauts, et n'est donc pas exactement conforme au modèle ;

- des phénomènes physiques aléatoires altèrent le fonctionnement de l'automate.

Ceux qui se font des illusions sur l'automate veulent croire que le modèle représente exactement le monde de la nature, que le programme est sans défaut, que les phénomènes aléatoires sont inexistantes ou négligeables : il survient alors des incidents qui suscitent chez eux à répétition, et sans qu'ils en tirent les leçons, le même étonnement douloureux.

On peut certes intégrer dans le programme une partie des actions que réclament les surprises et phénomènes aléatoires (contrôle et redondance des données, répétition des messages, reprise automatique en cas d'incident etc.) mais une partie seulement : sinon le modèle représenterait la nature de façon exacte et le programme ne comporterait aucun défaut, conditions qu'il est impossible de réunir.

Le fonctionnement de l'automate pur doit donc, pour répondre aux surprises et parer aux incidents auxquels il ne répond pas de lui-même, être *supervisé* par un être humain. *La supervision est pour l'automate pur un complément nécessaire*, mais l'intervention de l'être humain introduit les contraintes propres à celui-ci.

Certaines entreprises, cultivant les illusions optimistes évoquées ci-

3. Les logiciels embarqués de la NASA, soumis à des exigences de qualité extrême, comportent de l'ordre de 0,1 défaut par millier de lignes (soit 70 défauts pour un programme de 700 000 lignes de code source).

4. IBM Journal of R&D, Vol. 40, No. 1, 1996, <http://www.research.ibm.com/journal/rd/401/tocpdf.html>.

5. Michel Volle, *e-économie*, Economica, 2000, chapitre VIII).

dessus, négligent de superviser leur plate-forme informatique ou la dotent d'une supervision partielle (elle contrôle par exemple les *mainframes* mais non le service de bout en bout). Il en résulte une mauvaise qualité de service que l'entreprise imputera à « l'informatique ».

2 La supervision

Le programme de l'automate doit être conçu de façon à apporter au superviseur des informations sur son propre fonctionnement, ainsi que des commandes sur lesquelles le superviseur pourra agir pour répondre aux surprises, pannes et incidents. La remontée des informations peut soit être automatique, soit passer par le canal d'un centre d'appel.

Contrairement à l'automate, le superviseur humain est capable de répondre avec bon sens à des situations imprévues (c'est en tout cas ce que l'on attend de lui), mais il est par ailleurs sujet à la fatigue, à l'oubli et à l'affolement. Pour que la supervision puisse être efficace il faut donc lui épargner les tâches répétitives fatigantes ainsi que les longs intervalles d'inaction lors desquels le souvenir des objectifs et méthodes s'effacerait (il peut être opportun de sous automatiser les opérations⁶). Il faut aussi lui éviter la *surcharge mentale* qui se produit lorsque des incidents divers s'accumulent rapidement.

Le délai d'action de la supervision n'est pas le même selon la nature de l'automate. Dans un réseau télécoms, le superviseur reçoit en temps réel les « tickets d'incidents » (rupture d'un câble, avalanche d'appels causée par un événement extérieur et qui sature les commutateurs) et lance sans délai les actions nécessaires. Si par contre un incident se produit dans un satellite ou une fusée, il faudra le plus souvent un délai pour en identifier la cause et le corriger.

Dans le transport aérien ou ferroviaire on peut considérer le réseau comme un automate même si la réalisation du programme implique l'action du personnel. La supervision d'un réseau de transport aérien est assurée par le « quart-opération » qui traite notamment les problèmes que causent les caprices de la météorologie ; lorsque la situation devient inextricable, le quart-opération a recours à un système expert qui lui propose des solutions.

En cas de sinistre grave (séisme important, accident d'avion ou de

6. Voir « Éloge du semi-désordre » www.volle.com/opinion/desordre.htm.

chemin de fer), la supervision passe la main à une « cellule de crise ». Celle-ci fonctionnera d'autant mieux que l'on aura prévu à froid les outils et méthodes dont elle doit être équipée, les informations et moyens d'action que le système d'information lui fournira.

3 L'édition périodique d'états

L'une des premières applications de l'informatique à la gestion a été, dans les années 60, la production d'états périodiques : fiches de paie, documents comptables, liquidation des retraites, inventaires etc.

Alors que la programmation des automates purs est bâtie autour d'algorithmes de calcul complexes manipulant un nombre réduit de données, la production d'états périodiques nécessite l'exécution de calculs simples sur de nombreuses instances individuelles d'un même format de données. Elle utilisera donc volontiers les SGBD et parfois la modélisation par objet.

Les états sont produits périodiquement, le plus souvent sous une stricte contrainte de délai, à partir de bases de données qui sont, elles, alimentées de façon continue. La qualité des états dépend donc (1) de la qualité des données saisies en continu, (2) de l'exactitude des traitements qui leur sont appliqués.

La vérification des données saisies peut se faire en temps différé (batch) mais il est plus efficace de lui consacrer un petit programme qui dialogue avec l'agent opérationnel de telle sorte que celui-ci puisse faire pendant la saisie les corrections nécessaires. La vérification porte sur la qualité syntaxique des données (respect des contraintes formelles de codage et de cohérence) et sur leur qualité sémantique (vraisemblance des valeurs en regard des corrélations connues). Les erreurs syntaxiques doivent être systématiquement corrigées, les anomalies sémantiques peuvent être confirmées après vérification⁷.

Les données saisies sont souvent enregistrées en temps différé (les données saisies lors du jour J ne sont présentes dans la base de données que lors du jour $J + 1$), ce qui déconcerte ceux des utilisateurs qui sont habitués à travailler sur un micro-ordinateur où les données saisies sont enregistrées instantanément. Le traitement se fait lui-même usuellement en temps différé, en utilisant de nuit la puissance informatique

7. Michel Volle, *Le métier de statisticien*, Economica, 1984, chapitre 4.

que les programmes transactionnels laissent libre. Les états édités par le programme sont vérifiés : l'expérience montre qu'il s'y produit souvent des erreurs qui, même en petit nombre, sont extrêmement gênantes pour l'entreprise (c'est le cas, par exemple, pour les erreurs dans les fiches de paie). Ces défauts, qui doivent être corrigés à la main, proviennent :

- d'erreurs de saisie passées à travers les mailles de la vérification ;
- de cas particuliers trop rares pour que le programme ait pu en tenir compte utilement, mais qui existent ;
- de la mauvaise qualité des référentiels (définition des identifiants et des données), choquante en principe mais fréquente en pratique et qui provoque des erreurs de codage ;
- des problèmes inhérents à la vie des bases de données qu'a décrits Isabelle Boydens⁸ : décisions rétroactives, évolution du contexte etc.

Le taux de reprise manuelle des erreurs est un critère de qualité pour les logiciels de paie, de comptabilité etc. Il est relativement élevé lorsqu'une nouvelle version est mise en place, puis il se stabilise autour d'un niveau non nul mais jugé acceptable.

4 La communication

Le traitement de texte⁹ est devenu facile à utiliser en 1980 avec Word-Perfect. La messagerie¹⁰, d'abord limitée aux utilisateurs d'un même mainframe sur un même site, est introduite sur l'Arpanet (ancêtre de l'Internet) en 1972 et s'affranchit ainsi de la distance. Ainsi dans les années 1990 l'informatique, que l'on avait pu croire confinée au traitement des données structurées, a outillé le langage naturel et la communication¹¹.

Le Web¹², inventé en 1991, s'impose dès 1995 comme la principale utilisation de l'Internet par le grand public et le commerce électronique. Il s'articule efficacement à des centres d'appel¹³. Ses navigateurs fournissent une interface commode vers les documentations électroniques dans les

8. Isabelle Boydens, *Informatique, normes et temps*, Bruylant, 1999.

9. www.volle.com/travaux/ttpc.

10. www.volle.com/travaux/applisinternet.htm.

11. Voir « L'informatique de communication »
www.volle.com/travaux/Infocom.htm.

12. www.volle.com/lectures/bernerslee.htm.

13. www.volle.com/opinion/centrappel.htm.

Intranets.

La communication était déjà présente dans la relation entre poste travail et serveur ainsi que dans le transfert des dossiers d'une étape à l'autre d'un *workflow*. La relation entre le SI et les agents de l'entreprise, clients, fournisseurs, partenaires, devient alors *multimédia* : les divers vecteurs de communication (courrier, téléphone, messagerie, formulaire sur le Web) s'articulent de telle sorte que l'information fournie sur l'un puisse être reprise par l'autre. La mise en cohérence d'une communication qui emprunte divers vecteurs (faire en sorte par exemple que les données saisies par un client sur un formulaire HTML soient automatiquement retrouvées par l'opérateur du centre d'appel) nécessite une architecture de haute qualité.

Divers types de communication (synchrone et asynchrone ; douée d'ubiquité ou locale ; sécurisée ou non etc.) fournissent leurs options aux spécificateurs. La documentation professionnelle se consulte commodément sur l'Intranet, qui outille également la rédaction coopérative.

5 Le processus de production

Dans les années 80, les entreprises ont commencé à étendre l'utilisation de l'informatique à leurs processus de production : il s'agissait de baliser et de contrôler l'enchaînement des activités qui concourent à l'élaboration d'un produit.

Un processus de production fonctionne de façon répétitive ; typiquement amorcé par une commande, il aboutit à une livraison accompagnée d'une facturation. On peut aussi identifier dans l'entreprise des processus internes correspondant aux procédures administratives courantes (traitement des demandes budgétaires, des demandes de mutation, etc.) ainsi qu'aux achats et à la logistique (gestion des stocks, transports etc.).

Lorsque l'on automatise un processus de production, il ne s'agit pas d'une automatisation totale au sens de l'automate pur évoqué ci-dessus mais d'une coopération entre l'être humain organisé et l'automate qui l'assiste, chacun des deux acteurs de l'alliage « EHO – APU » faisant ce qu'il sait faire le mieux¹⁴.

La modélisation d'un processus¹⁵ emprunte le formalisme du *work-*

14. Voir « Approche du SI par les processus » www.volle.com/travaux/processus.htm.

15. www.volle.com/travaux/cdc.htm.

*flow*¹⁶, qui permet de décrire l'enchaînement des activités, d'automatiser l'adressage d'une étape à l'autre et les éventuels reroutages, les horloges, ainsi que les indicateurs de ressource, volume, délai et qualité nécessaires à l'administration du processus. Il outille ainsi à la fois l'agent opérationnel qui contribue au processus et le manager opérationnel ou l'administrateur qui en supervisent le fonctionnement.

Les agents impliqués dans un processus disposent sur le SI d'interfaces qui leur permettent de consulter des données, saisir des données et lancer des traitements. L'idéal – mais il est difficile à atteindre, c'est le Graal de l'informatique – est que cette interface leur présente à chaque instant exactement les informations, espaces de saisie et outils de traitement dont ils ont besoin pour leur travail, et masque ainsi l'architecture applicative du système d'information : c'est le sens que l'on peut donner, du point de vue de l'utilisateur, aux SOA (*Service Oriented Architectures*).

L'exploitation en temps différé ne convient généralement pas ici parce que le processus exige des performances élevées en termes de délai. Il faudra donc utiliser un système « en temps réel » outillé d'un moniteur transactionnel, ce qui introduit des contraintes de qualité spécifiques (« acuité » des transactions). Les données saisies, puis traitées, sont stockées dans une base de données et consultables par les agents qui participent au processus : le processus « fait la ronde » autour des données. Les divers dossiers que les agents manipulent sont commodément représentés par des « objets » au sens des langages à objets ; à l'évolution d'un dossier en cours de traitement correspond alors le « cycle de vie » de l'objet.

Le programme ne peut pas prévoir tous les incidents et cas particuliers possibles, notamment dans la première ligne (centre d'appel, agence, traitement du courrier et des messages, réception des commandes etc.) qui est au contact avec les clients et doit transcrire leur vocabulaire selon les codages du SI de l'entreprise. Il faut donc ménager des possibilités de « reprise de main » manuelle.

Des interfaces sont fournies aux clients, fournisseurs et partenaires, souvent en tirant parti du Web (voir ci-dessous). Pour les clients, il s'agira d'une aide à la commande (consultation du catalogue, configuration de la commande, paiement en ligne) ; pour les fournisseurs, d'une consultation de l'état des stocks et d'une facturation automatique. Si l'on veut que les fournisseurs et partenaires interviennent dans le processus de production il faut que les systèmes d'information soient *interopérables*, c'est-à-dire

16. www.volle.com/travaux/workflow.htm.

que les données partagées soient codées de façon compatible : l'interopérabilité passera souvent par le partage de formats XML.

Lorsque l'informatique est passée de l'édition d'états périodiques à l'automatisation des processus de production, son rôle dans l'entreprise a changé. D'un auxiliaire utile de la gestion elle s'est transformée en acteur de la production, ainsi que de la relation avec les clients, fournisseurs et partenaires. Cette évolution a naturellement rencontré des obstacles institutionnels et intellectuels. Elle a exigé par ailleurs des architectures nouvelles qui impliquaient des prouesses techniques.

Le couple formé par le transactionnel et les bases de données a formé, avec la gestion du cycle de vie des objets et les contraintes de concurrence et de persistance, un ensemble complexe, soumis de surcroît à des exigences de performance – car les utilisateurs s'attendent, lorsqu'ils utilisent le SI, à une rapidité analogue à celle qu'ils rencontrent sur leur micro-ordinateur – et à une volumétrie élevée (certains processus supposent que des dizaines de milliers d'utilisateurs, dispersés géographiquement, puissent travailler simultanément sur une base de données comportant des millions de dossiers individuels).

6 L'aide à la décision

Les outils qui équipent la supervision, les indicateurs que produisent les workflows, sont des aides à la décision en temps réel, au « pilotage ». Nous considérons ici l'aide à la décision stratégique, qui définit en quelque sorte le « plan de vol » de l'entreprise¹⁷.

Dans le SI de l'entreprise s'accumulent les données opérationnelles, les dossiers en cours. Lorsqu'elle n'a plus besoin d'eux l'entreprise les archive en les classant dans une mémoire peu coûteuse et difficile d'accès (par exemple sur une bande magnétique). L'archivage a une finalité juridique : il s'agit de pouvoir retrouver un dossier en cas de contentieux.

Les bases de données opérationnelles, « vivantes », ne gardent donc pas trace du passé. En outre beaucoup de données y manquent (on a pu les comparer « à une plage anglaise en hiver ») et certaines sont fausses, l'agent opérationnel ne contrôlant soigneusement que les données qui importent à son action – ce qui est d'ailleurs parfaitement normal.

17. Voir « Le système informatique d'aide à la décision » www.volle.com/rapports/siad.htm et « Histoire d'un tableau de bord » www.volle.com/travaux/tdb.htm.

Le SI opérationnel n'est donc pas construit pour fournir les séries chronologiques qui permettraient au stratège¹⁸ (i. e. au comité de direction) de comprendre le fonctionnement de l'entreprise, la situer dans son environnement concurrentiel, percevoir de façon précoce les changements de tendance. On doit, si l'on veut fournir au stratège les données dont il a besoin pour faire son travail, compléter le SI opérationnel par des extractions de données, des traitements et redressements statistiques, des méthodes d'interprétation, une activité éditoriale etc.

Le système d'aide à la décision s'appuie alors sur une « base morte » que les bases vivantes alimentent périodiquement, sur des progiciels statistiques et économétriques, sur des hypercubes et tableaux de bord.

Beaucoup d'entreprises peinent à définir leur SIAD. Elles croient que la connaissance peut provenir des données brutes et refusent l'interprétation, voire même le redressement, qui leur semblent « subjectifs ». La même crainte les empêche de faire un tri dans les données, qu'elles présentent en masse au stratège. À supposer que ces obstacles intellectuels soient surmontés, les obstacles techniques sont redoutables : il n'est pas facile de construire un datawarehouse¹⁹, de produire des hypercubes, d'alimenter régulièrement et dans les délais un tableau de bord sélectif et bien conçu.

* *

Ainsi l'automate s'est mis progressivement au service des divers acteurs de l'entreprise, de l'agent opérationnel au stratège, en passant par le manager opérationnel, l'organisateur et l'expert²⁰). En sortant du territoire de la gestion pour outiller les processus de production, il s'est lié organiquement à l'entreprise à laquelle il fournit une « doublure informationnelle²¹ ».

Cela concrétise le fait que l'entreprise ne s'incarne, ne s'organise pas seulement dans l'espace physique avec ses bâtiments, équipements et bureaux, mais aussi dans l'« espace logique²² » des concepts, définitions, représentations, méthodes, documents etc. que les réseaux détachent de l'espace physique, que l'informatique outille et structure en délimitant les habilitations et en facilitant le classement et la recherche.

18. www.volle.com/opinion/strategie3.htm.

19. www.volle.com/travaux/dwh.htm.

20. Voir « Le marketing interne » www.volle.com/travaux/marketinginterne.htm.

21. Voir « L'émergence d'un alliage » www.volle.com/travaux/alliage.htm.

22. Voir « Explorer l'espace logique » www.volle.com/travaux/espacelogique.htm.

La prochaine « marche d'escalier » de l'informatisation sera vraisemblablement celle de l'exploration et de la colonisation délibérées de cet espace logique sur lequel les entreprises ont pénétré sans s'en rendre compte : elles montent l'escalier en marchant à reculons.