Daniel Stévenin – 3 avril 2013

Les enjeux et contraintes de l’informatique industrielle et des télécommunications dans la construction du SDART

Le SDART a été défini et construit en fonction des contraintes techniques de l'époque. Il n’est pas inutile, quelques 40 années après sa conception, de se rappeler le contexte de l’informatique et des télécommunications des années 70 pour mieux comprendre la logique industrielle et technique de sa construction.

# Les débuts de l’informatique industrielle

Avant le SDART, l’informatique avait déjà fait son apparition à EDF, notamment dans le domaine de la gestion et même dans les dispatchings, alors équipés de calculateurs appelés « 9040 ». L’informatique n’était donc pas tout à fait une nouveauté, mais elle était en plein développement. Les années 70 ont été marquées par la diminution de la taille des ordinateurs. On découvrait alors la « mini-informatique » et ses « mini-ordinateurs », pouvant contenir dans une seule armoire l’ensemble du matériel nécessaire à leur fonctionnement. Les premiers d’entre eux ne disposaient que de 16 kilo-octets de mémoire. Mais cela annonçait déjà la « micro-informatique » qui permettra dans la décennie suivante l’éclosion du PC (Personal Computer) d’IBM en 1981 ou le Macintosh d’Apple.

Les concepts comme le vocabulaire étaient encore très instables, et la programmation restait très proche des mécanismes de base de la machine utilisée. Les langages de haut niveau faisaient leurs premiers pas, mais le recours au langage machine, appelé « l’assembleur », restait souvent nécessaire pour répondre aux contraintes des traitements en temps réel.

Dans ce contexte de découverte du potentiel de l’informatique pour remplir des fonctions de téléconduite, les produits industriels n’existaient pas encore. Tout au plus pouvait-on parler de solutions prototypes. Les efforts portaient aussi bien sur la demande que sur l’offre, à ajuster progressivement l’une à l’autre.

L’un des principaux enjeux consistaient à faire communiquer l’ordinateur avec le monde extérieur. Les télécommunications existaient depuis longtemps, et permettaient notamment de transporter des signaux analogiques comme la voix. Encore fallait-il transmettre des données sous forme digitale, compréhensible par les ordinateurs. C’est ainsi que ce sont construits diverses formes de « couplage » de l’unité centrale des mini-ordinateurs pour les rendre capables d’interagir avec le monde extérieur en temps réel.

# Les enjeux techniques initiaux du SDART

C’est à ce pari que prétendait répondre le SDART : faire fonctionner des équipements utilisant de nouvelles techniques numériques en les reliant avec les moyens de télécommunications existants pour remplir les fonctions de téléconduite du réseau de transport d’électricité.

Assez naturellement, on espérait ainsi profiter de nouvelles opportunités fonctionnelles, et rendre le métier de conducteur de réseau électrique plus performant : capacité à mieux connaitre l’état du réseau, à réagir plus vite et mieux, à conduire le réseau plus prêt de ses limites, etc…

Il est intéressant de noter que la principale qualité attendue de l’informatique résidait dans sa supposée souplesse et adaptabilité.

Le vocabulaire (anglais) de l’époque ne parlait-il pas de « software » et de « hardware » ? On voulait ainsi opposer ce qui est « hard » et donc rigide, à ce qui est « soft » et donc souple.

On espérait ainsi que le matériel, le « hardware », serait installé une fois pour toute, si possible avec des coûts en baisse au fur et à mesure du déploiement d’un nombre croissant d’équipements, alors que le « software » (sa traduction en logiciel n’existait pas encore) ne couterait rien ou presque. Certes, on s’attendait à des coûts de développement initiaux assez élevés, mais qui s’amortiraient sur le nombre d’équipements, et surtout, qui seraient très faibles lors des adaptations ultérieures.

On sait aujourd’hui que les choses ont été très différentes.

Les coûts des matériels informatiques ont certes fortement baissé, mais pour des raisons très différentes, lors de l’apparition de l’informatique de masse destinée au grand public.

On sait ainsi que la taille des équipements a fortement diminué, et que leurs performances se sont accrues dans de très grandes proportions (cf la fameuse loi de Moore qui prévoyait dès 1965 un doublement du nombre de transistors par circuit de même taille, à prix constants, tous les 18 mois, ce qui s’est vérifié jusque-là).

Quant au logiciel, il s’est révélé très couteux aussi bien en développement initial qu’en maintenance, et ses couts ne sont finalement acceptables qu’en regard de la taille du marché.

Une autre caractéristique de l’informatique était très présente dans l’esprit des concepteurs du SDART : la capacité d’un système numérique à s’autotester pour alléger le travail de l’exploitant et améliorer la sûreté de fonctionnement. On verra que cet objectif a été atteint, puisque la qualité des téléinformations a globalement augmenté.

Pour le reste, il fallait surtout commencer par vérifier la faisabilité d’une téléconduite informatique, compte tenu des contraintes techniques auxquelles étaient confrontés les pionniers de l’informatique temps réel à ses débuts.

Les chapitres à examiner concernent :

* Le comportement des équipements informatiques dans un environnement industriel
* La transmission des données numérisées
* Les performances « temps réel » des calculateurs
* La sûreté de fonctionnement
* La description des données de téléconduite

# L’environnement « hostile » des équipements numériques

Les premiers calculateurs informatiques ont été installés dans des salles spécialement équipées pour les accueillir. Un soin particulier devant être accordé à la température de fonctionnement, ces salles étaient et sont encore le plus souvent climatisées.

Les équipements de téléconduite, au PC comme au PA sont eux prévus pour fonctionner dans l’environnement d’un poste électrique, avec des contraintes beaucoup plus sévères.

Deux contraintes ont fait l’objet de mesures particulières : la température et l’environnement électromagnétique.

C’est ainsi que des climatisations ont fait leur apparition dans les salles de « télécommunications » initialement conçues pour accueillir les différents équipements de téléphonie et télétransmissions pré existants.

Dans la continuité du travail accompli depuis plusieurs années pour les équipements de téléphonie, un soin particulier a été accordé à l’alimentation électrique. C’est pourquoi le choix a été fait d’une alimentation à courant continu avec une alimentation secourue par batterie. Alors que les mini calculateurs étaient généralement alimentées en 220 V alternatif, et il a fallu faire développer par les constructeurs des alimentations spécifiques.

Ainsi, les premiers calculateurs de l’EDT de PCG ne s’appelaient pas Mitra 15 (leur nom d’origine), mais FDE 15 T.

(Pour l’anecdote, mais ce serait à vérifier, la CII a choisi ce nom codé pour qu’on reconnaisse EDF sans le nommer, avec le « T » de « transport »)

De nombreux essais ont été nécessaires, sur le PC mais surtout sur les PA, pour vérifier qu’ils supportaient l’environnement électromagnétique sévère d’un poste électrique. Ce sont les cartes « coupleurs » (les « handlers ») qui se sont révélées les plus fragiles, et de nombreuses évolutions de version ont été nécessaires. Cela a engendré de nombreux retards sur les premiers développements.

Ces phénomènes étaient déjà connus des spécialistes des télécommunications d’EDF qui, par exemple, connaissent bien la fameuse « NS24 » (note de service 24 des télécommunications) qui édictaient des règles à respecter pour éviter des perturbations réciproques entre télécommunications du domaine public et postes électriques. Le fait nouveau provenait de la technologie numérique utilisée, et du manque de compétence, voire d’intérêt, des informaticiens pour les questions relatives aux « courants forts ».

Ces différents points ont fait l’objet de travaux internationaux à la CIGRE ou à la CEI (Comité 57), pour normaliser les niveaux de contraintes auxquels devaient être soumis les équipements de téléconduite dans les postes électriques.

# La transmission des données numérisées

Transmettre des données constituées d’une suite de signaux binaires (0 ou 1), pose des questions différentes de la transmission de données analogiques comme la voix.

Des questions de fonctionnement d’abord : s’il est aisé de reconnaitre le début d’un message vocal, comment reconnait-on le début ou la fin d’un message codé binaire ? en cas de perturbation de la ligne de transmission, la voix se déforme ou devient inaudible, les interlocuteurs le détectent d’eux-mêmes et agissent en conséquence. Ils répètent leurs phrases ou se rappellent plus tard. Mais, comment détecter voire corriger un message composé de suites de 0 et de 1 ?

Des questions de performance aussi. Les débits de transmission de l’information disponibles aux débuts du SDART étaient beaucoup plus faibles qu’aujourd’hui. Certes suffisants pour transmettre la voix, ils devenaient problématiques pour transmettre des données volumineuses. Cela a eu un fort impact sur le cahier des charges du SDART.

### La synchronisation des trames de données digitales

Parmi les techniques alors disponibles, selon le mode asynchrone ou synchrone de transmission des données.

Plutôt que de retenir un choix unique, EDF a préféré laisser ouvertes plusieurs possibilités. Le SDART prévoyait donc des modes de synchronisation asynchrone à « coupure de porteuse » et des modes synchrones comme HDLC (une norme émergente à l’époque) ou ECMA.

Il faut observer que le choix du mode à coupure porteuse, encore très utilisé en France, ne l’a pas été ailleurs. Doit-on en déduire que c’était une erreur ? Difficile à dire puisque cela a permis au SDART de se déployer bien avant la plupart des autres systèmes, mais qu’en contrepartie, cela a ensuite constitué une forte rigidité sur les équipements concernés.

### Les mécanismes de détection d’erreur

Cette question a fait débat pendant longtemps au sein des organismes de normalisation des télécommunications utilisées en téléconduite, notamment à la CEI.

La réponse de principe consiste à ajouter de l’information redondante dans le message utile émis, pour pouvoir vérifier l’absence d’erreur sur le message, voire même le corriger sur le niveau de redondance est suffisant.

Plusieurs grands constructeurs allemands ou suédois (BBC, ASEA,..) souhaitaient interdire la norme HDLC au prétexte que son mécanisme de synchronisation de trame était trop risqué en cas d’utilisation de moyens de transmission fortement bruités. Ils préconisaient des mécanismes spécifiques à la téléconduite, et la bataille pour le meilleur standard a fait rage pendant de nombreuses années.

Là aussi, EDF a fait un choix permettant de s’adapter à toute évolution.

### Une conception originale de l’ « automate d’échange »

On le voit, la conception du SDART avait pour objectif de laisser suffisamment de souplesse pour accueillir et respecter le moment venu la norme de transmission qui était attendue de tous.

C’est ce principe qui a présidé à la conception modulaire de ce qu’il est convenu d’appeler sous un vocable global la « procédure de transmission de téléconduite » qui a été normalisée dans sa première version en 1978 par la DER. Aujourd’hui encore, elle est connue sous son nom abrégé « norme HNZ ».

Cette norme interne se voulait aussi proche que possible du modèle développé par l’ISO pour les transmissions HDLC, du moins pour sa partie « automate d’échange ».

Elle distinguait en effet un automate de transmission et un automate d’échange.

L’automate de transmission décrivait les mécanismes de base de modulation (asynchrone ou synchrone) et de synchronisation de trame (HDLC, ECMA, …)

L’automate d’échange décrivait les modalités de l’échange proprement dit, indépendamment du mode de transmission choisi : procédure d’initialisation ou de finalisation de l’échange (se dire bonjour ou au revoir), mécanismes d’envoi ou de réception d’un message, mécanisme de répétition en cas de détection d’erreur, etc…

2 modes différents ont été développés pour cet automate d’échange. Le mode « maître / maître » et le mode « maitre /esclave ».

Dans le premier, utilisable entre 2 interlocuteurs en point à point, chacun peut prendre l’initiative de l’échange, à égalité de droits.

Dans le deuxième, utile dans le cas d’une ligne de transmission sur laquelle existe plusieurs interlocuteurs, un « maitre » joue le rôle de chef d’orchestre et interroge successivement chacun des interlocuteurs, ces « esclaves » n’émettant des messages qu’à la demande du « maître ».

Ainsi conçue, cette norme interne a permis de couvrir tous les besoins rencontrés dans le cadre du SDART, en attendant qu’une ou plusieurs normes internationales s’imposent de façon indiscutable.

### Le choix de la transmission sur changement d’état.

La principale fonction d’un système de téléconduite consiste à transmettre au PC l’information sur l’état ouvert ou fermé de chacun des systèmes de coupure (disjoncteurs, sectionneurs,..) du réseau, et des mesures (courant, tension,..) en différents points de ce réseau. Dans l’idéal, si on dispose d’un réseau de télécommunications largement dimensionné et peu couteux, on peut imaginer de transmettre cycliquement l’ensemble de ces informations. Mais, dès lors que le nombre d’informations à transmettre est trop volumineux pour les « tuyaux » disponibles, il faut envisager d’autres méthodes. C’est ainsi que le choix s’est porté sur une transmission sur changement d’état pour les organes de coupure, et de cycle lent pour les mesures d’intensité ou tension.

Ce choix s’accompagne nécessairement du mécanisme de « contrôle général » qui est appelé chaque fois qu’un doute existe sur la qualité des informations transmises. Ce contrôle consiste à inviter tous les équipements de la zone douteuse à transmettre de nouveau l’état de tous les organes surveillés.

# La performance des calculateurs « temps réel »

Un programme s’exécutant en temps réel a pour spécificité de devoir traiter les événements qui surviennent dans un ordre aléatoire, non prévu à l’avance, avec la rapidité requise.

Les premiers calculateurs informatiques ont été conçus pour traiter les données dans un ordre séquentiel, sans contrainte de temps particulière. Le passage au traitement dans un ordre aléatoire posait la question de l’ordonnancement des tâches, celles-ci devant s’exécuter dans un temps inférieur au temps de réponse attendu du calculateur.

Les mini calculateurs de l’époque se constituaient schématiquement de plusieurs éléments :

* Les « handlers », ou coupleurs, chargés de réaliser l’interface entre le monde extérieur et la machine, comportant une partie matérielle (la carte de couplage) et une partie logicielle (le « handler ») ;
* Une mémoire centrale et un processeur, l’ensemble formant la machine numérique capable d’exécuter un programme, lui-même formé d’un ensemble d’instructions préalablement compilées et introduites dans la mémoire du calculateur ;
* Un châssis capable d’accueillir les différents éléments matériels nécessaires au fonctionnement du calculateur : l’alimentation électrique, le bus permettant les échanges entre les différents éléments (se présentant sous la forme de « cartes électroniques).

Le programme introduit en machine devait donc réaliser des tâches d’entrées et sorties pour gérer les échanges avec le monde extérieur et, en fonction des événements à traiter, décider des actions à entreprendre.

On comprend la nécessité d’un découpage du programme en tâches élémentaires, et d’un ordonnanceur pour les lancer, les suspendre, ou les relancer au gré des événements.

Les concepteurs des premiers équipements de téléconduite ont mis en œuvre des solutions spécifiques, programmées en langage assembleur, dont ils vérifiaient élément par élément, puis au niveau du système global, la compatibilité avec les contraintes de performances. On imagine les efforts de recherche, d’essais et d’erreurs, que cela a représenté avant d’aboutir à des solutions fiables et stabilisées.

On trouvera en annexe un rappel de l’histoire de l’évolution des « schedulers », les ordonnanceurs de tâches aussi appelés « operating system ».

# la sûreté de fonctionnement

La sûreté de fonctionnement a certainement été une préoccupation majeure du SDART : il était hors de question de détériorer la situation antérieure, et on pouvait légitimement se demander si l’utilisation de nouvelles techniques informatiques permettrait de relever le défi.

Le SDART a donc fixé des exigences de niveau élevé, en termes de fiabilité (au travers du taux de défaillance, typiquement de l’ordre de quelques pannes par an pour l’ensemble du système de téléconduite) et de maintenabilité (au travers du temps moyen de réparation du système, typiquement de l’ordre de quelques heures). S’agissant de la sécurité, c'est-à-dire la capacité du système à ne pas conduire à des accidents inacceptables, les exigences se déclinaient ainsi : pas de transmission d’information erronée sans le savoir ; pas d'ordre intempestif, à défaut ne rien faire.

Les réponses ont porté sur les différents éléments de la chaine de téléconduite, selon une approche globale de l’ensemble du système.

Si on admettait de ne pas doubler les calculateurs de postes asservis (les PA), on a dû recourir au doublement des calculateurs de niveau supérieur (les PC de PCG, de même que les calculateurs régionaux et ceux du centre national).

### L’autotest des équipements informatiques

Le recours à l’autotest a apporté un gain important dans la sûreté de fonctionnement.

Les tests pratiqués automatiquement par les équipements eux-mêmes concernaient leurs fonctions vitales, et avaient pour objectif de déclencher une alerte auprès de l’exploitant pour l’informer d’un dysfonctionnement et de la nécessité de procéder à une réparation. Dans le cas des équipements doublés au PC, outre la fonction d’alerte de l’exploitant, la détection d’un défaut grave provoquait l’arrêt de l’équipement incriminé et le basculement automatique sur l’équipement de secours.

On peut mentionner quelques exemples d’autotests.

Le test des commandes de changement de position d’organe.

Comment s’assurer que la télécommande d’un organe fonctionnera correctement lorsqu’on en aura besoin, alors que cet organe (disjoncteur ou actionneur) est rarement actionné ? Une réponse consiste à injecter périodiquement un ordre bref, de durée suffisante pour tester en retour sur la bobine du relais de déclenchement que l’ordre a été émis, puis d’envoyer immédiatement l’ordre inverse avant que l’organe n’ait eu le temps de prendre ces ordres en compte. Une défaillance du dispositif de commande sera ainsi détectée et la réparation effectuée avant que son utilisation réelle ne soit requise.

Le test global d’un calculateur.

Il est évidemment illusoire de prétendre tester en permanence l’ensemble des fonctions d’un calculateur. Des études en ont d’ailleurs prouvé l’impossibilité, tant les causes de pannes sont nombreuses, à commencer par celles qui peuvent affecter l’outil de test lui-même…

Une solution retenue sur les premiers PC de PCG consistait à implanter un programme extrêmement simple au niveau de priorité le plus bas de la machine : ce programme réalisait une boucle permanente d’activation / désactivation d’un signal externe (en l’occurrence l’allumage de voyants sur le pupitre du calculateur). En cas de surcharge prolongée du calculateur, signe d’un dysfonctionnement grave, ce signal externe n’était plus activé, et un équipement externe, appelé « bac de commutation », chargé de surveiller ce signal, détectait ainsi la panne du calculateur, et pouvait lancer le calculateur de secours.

Ce dispositif, très simple et très fiable, perdure aujourd’hui.

Ces principes généraux ont été conçus dès l’origine du SDART, et progressivement améliorés par la suite. Ils sont encore en application dans leurs grandes lignes.

### Le test des équipements avant leur mise en service

Il va de soi que le meilleur autotest n’aura les effets attendus que si l’équipement testé a été mis en service dans un état de bon fonctionnement.

Pour s’en assurer, il faut s’être préalablement assuré en amont que l’équipement «prototype » qui préfigure l’ensemble des équipements d’un même modèle, est bien conforme aux spécifications, tant sur le plan technique, fonctionnel, que des performances attendues.

C’est pourquoi le SDART a accordé une grande place aux essais sous leurs différentes formes, dans le cadre d’un processus rigoureux de validation des systèmes et de tests des équipements de série (généralement réalisés par les constructeurs eux-mêmes selon un plan de qualité convenu entre le constructeur et son client)

On notera que le SDART a été l’occasion pour les constructeurs informatiques concernés, de faire des essais dans des domaines qu’ils connaissaient peu jusque-là : essais en température, essais de « compatibilité électro magnétique », essais de résistance aux chocs électriques,…)

On peut aussi noter le soin particulier porté à l’interface entre les équipements informatiques et les organes du réseau électrique. A cet effet, la filerie préexistante a été progressivement modifiée en suivant une politique de « paliers techniques » pour des raisons de rigueur technique tout en recherchant l’optimisation économique, la grande difficulté consistant à déterminer le meilleur moment pour passer d’un palier à un autre.

### L’utilisation de simulateurs

L’emploi de simulateurs s’est révélé indispensable pour effectuer les différents essais de qualification des prototypes.

Si les tests « manuels » sont suffisants pour réaliser des essais de fonctions élémentaires, il est nécessaire de disposer d’un outil permettant de soumettre l’équipement testé aux différentes situations ou séquences d’évènements susceptibles de se produire dans son environnement réel.

Pour cela, différentes formes de simulateurs d’environnement ont été réalisés, allant des simulateurs reproduisant les comportements « normaux » du réseau de téléconduite aux simulateurs permettant de simuler différents types de pannes.

Ces simulateurs se sont révélés très efficaces pour tester des fonctions complètes, et notamment des séquences d’avalanches d’événements pour vérifier le bon fonctionnement et la tenue en performance des systèmes face à des situations complexes.

On notera, à titre de preuve, que le système de téléconduite s’est très bien comporté lors des tempêtes de fin 1999, malgré les avalanches d’informations à traiter dans des temps très courts.

Des simulateurs de comportement du réseau ont aussi été développés pour remplir une autre fonction, l’entrainement des personnels d’exploitation. Ces simulateurs permettent de générer des scénarios prévus à l’avance par un instructeur pour tester les réactions du stagiaire et le former aux bonnes pratiques. Là aussi, ces outils se sont révélés très intéressants, et ils sont aujourd’hui largement utilisés pour la formation du personnel.

# Le langage de description des données et des fonctions

Les spécifications initiales des équipements du SDART étaient écrites… en (bon) français, condition nécessaire mais non suffisante pour qu’elles soient suffisamment précises et complètes pour éviter les ambiguïtés voire les incohérences inévitables.

Des efforts importants ont été consacrés à la qualité des spécification des fonctions de téléconduite (prises au sens large, en prenant en compte notamment la descriptions des performances requises), ainsi qu’à la qualité du langage de description des données traitées, appelé langage de « configuration des données ».

### Vers un langage de configuration universel ?

Les premiers éléments de description des données de téléconduite étaient réduits à leur plus simple expression, jusqu’à être parfois mélangés à leur programme de traitement.

Un premier effort de séparation des données et des programmes a été conduit dans le cadre du projet d’EDT de PCG, et a abouti à la construction du premier « configurateur », écrit en langage Fortran. Mais ce configurateur ne concernait que le PC, les PA étant configurés par ailleurs et indépendamment.

Le projet « configurateur généralisé » a constitué l’étape suivante, pour établir des liens entre les différents fichiers de données, et les générer de façon cohérente, la difficulté étant de laisser suffisamment de marges de manœuvre aux différents constructeurs des PA pour ne pas fausser la concurrence entre eux.

Bien des années plus tard, les diverses et nombreuses tentatives de normalisation d’un langage universel des données de téléconduite n’ont pas encore vraiment abouti, moins pour des questions techniques que pour des raisons de stratégies industrielles des acteurs en présence. Rappelons que le marché concerné reste très réduit si on le compare par exemple à celui des équipements informatiques accessibles au grand public connectés à internet. Les questions de standardisation se posent donc très différemment, et n’intéressent qu’un nombre réduit d’industriels.

### Des spécifications fonctionnelles sous forme d'automates

La langue française est assurément une langue de spécification tout à fait acceptable pour spécifier des fonctions de téléconduite. Pour autant, le besoin d’outils mieux adaptés à la description d’automates s’est vite fait sentir pour éviter les ambiguïtés compte tenu de la multiplicité des séquences d’événements susceptibles d’intervenir dans un grand nombre de situations possibles.

La DER a développé plusieurs outils, complémentaires ou parfois concurrents, qui ont été utilisés à différents niveaux.

Lorsque les fonctions restent assez simples du point de vue de la combinatoire, des modélisations de type SADT ont été largement utilisés. Mais, dès lors que des combinaisons plus complexes doivent être décrites, en s’apparentant au fonctionnement d’automates séquentiels, il faut utiliser des méthodes plus adaptées.

Dans l’approche dite des « tableaux de décision », on décrit tous les états atteignables par un automate, et tous les événements susceptibles de l’affecter, ainsi que le traitement à effectuer dans chacun des cas.

Dans l’approche dite des « réseaux de Pétri », on décrit les enchainements possibles de situations / événements sous la forme d’un réseau dont la représentation graphique rend le fonctionnement plus visuel.

# Des enseignements pour aujourd’hui ?

Les techniques informatiques ont profondément évolué depuis 40 ans, et il est évident qu’on n’aborde plus aujourd’hui leur enseignement comme on le faisait dans les écoles des années 1970. Les concepts se sont enrichis, les méthodes de développement des logiciels ont beaucoup évolué, au point que certaines qui prévalaient au siècle dernier sont maintenant considérées comme obsolètes. Les ressources en télécommunications et en capacité de stockage de l’information ne sont plus, ou sont beaucoup moins des contraintes pour le programmeur.

Pourtant, certains principes demeurent, et il parait important de les repérer.

#### L’approche systémique

On emploie souvent le terme « système informatique » pour désigner un ensemble d’équipements numériques qui concourent à la réalisation d’une fonction. Mais il ne faut jamais oublier de considérer le contexte dans lequel il se place, en y incluant notamment les opérateurs humains. Le système informatique n’est jamais qu’une partie de cet ensemble qu’il faut regarder dans sa globalité.

C’est ce qui a conduit les concepteurs de SDART à faire convenablement la part de ce qui peut être automatisé et de ce qui doit rester du ressort de l’opérateur humain. L’automatisation n’est pas une fin en soi. Elle ne doit servir qu’à faciliter le fonctionnement global d’un système dans des conditions technico-économiques aussi proche que possible de l’optimum, l’homme étant inclus dans l’analyse.

Oublier ce principe peut conduire à concevoir des systèmes automatiques trop sophistiqués, risquant d’échapper au contrôle ou de ne pas apporter les réponses appropriées face à des séquences d’événements inattendues. Nous avons connu dans le passé des tentations de ce type, où une trop grande sophistication a conduit à un résultat finalement très éloigné de l’objectif initial, avec notamment une dégradation du retour sur investissement.

#### L’effet changement de génération

L’évolution très rapide des techniques pose inévitablement la question de l’obsolescence des équipements. Quand doit-on passer d’une génération à une autre ? Une politique de palier technique doit être élaborée en évitant, d’un côté, l’effet « zapping » consistant à prendre en compte la moindre nouveauté et, de l’autre côté, le maintien dangereux d’un palier obsolescent.

Cet arbitrage est délicat, et l’expérience montre qu’il n’y a pas de règle simple pour l’établir. Ce qui compte, c’est de poser périodiquement la question, et d’arbitrer en fonction des conditions du moment, sans chercher à reproduire des raisonnements qui ont pu se justifier à une époque précédente.

Les changements de génération sont des décisions difficiles à prendre car le retour sur investissement est difficile à évaluer. Tout l’art consiste à ne pas dépasser un certain seuil, au-delà duquel les coûts de maintenance dus à l’obsolescence croissent très vite.

#### La conduite des projets et la tenue des délais

La tenue des délais est une difficulté récurrente de tout projet, et particulièrement des projets à forte composante informatique.

Cette difficulté n’est pourtant pas une fatalité, à condition de savoir la traiter avec sérieux et une ténacité de tous les instants, dans le cadre d’une conduite de projet rigoureuse, partagée par tous les acteurs, avec un responsable unique clairement mandaté.

#### L’importance des essais

L’une des raisons pour lesquelles le SDART a pu avoir des difficultés de tenue des délais tient en la gestion des essais. Les plannings initiaux ne tenaient pas suffisamment compte des temps nécessaires aux tests, car on avait sous-estimé leur importance. Force a été de reconnaitre que ces essais sont à la fois nécessaires et couteux en temps. Il n’existe pas de programme logiciel qui fonctionne bien du premier coup. Les essais doivent être pris en compte à toutes les phases du processus de développement d’un système informatique. Celui-ci doit être outillé pour permettre ces essais dès l’amont du processus. Plus tard les erreurs sont détectées et plus elles coutent en correction.

### Annexe : l’évolution des « operating system »

Les premiers minis calculateurs munis de « schedulers » microprogrammés, c'est-à-dire programmés « en dur » dans le « hardware » de la machine, n’ont vu le jour qu’après 1975. Ce fut le cas du Solar développé par la société Télémécanique, et qui a été utilisé dans le développement des SIT 3 (Système Informatisés de Téléconsuite de type 3) de la Direction de la Distribution.

Le Service du transport ayant fait le choix du Mitra 15, développé par la société CII et ne disposant pas de cette technique, la question du développement d’un scheduler s’est donc posée à la Direction des Etudes et Recherches d’EDF.

Un premier projet de réalisation d’un « moniteur temps réel » a été confié au Service IMA (Informatique et mathématique appliquée » de la DER en 1974. Arrêté en 1976, ce projet a été repris par la Division « Conduite des réseaux » du Service Etudes de réseaux, alors en charge du développement du premier EDT(Ensemble de traitement) de PCG (le Pupitre de Commande Groupée du SDART).

Le premier projet avait échoué pour des raisons de complexité. Ses concepteurs avaient opté pour un système « multi tâches » capable d’exécuter en parallèle des traitements d’événements survenant de façon quasi simultanée. Cela a posé des problèmes difficiles à résoudre. Ces problèmes sont principalement dus à la question de la « réentrance » des programmes. En effet, une tâche élémentaire n’est implantée que dans un endroit unique de la mémoire. Le « moniteur » doit donc savoir traiter le cas de deux sollicitations successives de cette tâche, la deuxième survenant alors que le traitement du premier n’est pas terminé. Un mauvais traitement de ce type de problème peut conduire à plusieurs genres de défauts plus ou moins graves : le non traitement d’un événement, une perte de chronologie, un engorgement des files d’attente des événements à traiter, etc.. Cela peut aussi entrainer un blocage pur et simple de la machine dans un cas bien connu des informaticiens sous le terme de « dead lock » ou d’interblocage par « verrou mortel ».

L’interblocage se produit lorsque 2 tâches font appel successivement aux même ressources dans un ordre différent.

Supposons que le programme de la tâche 1 fasse appel successivement aux ressources A et B, et libère la ressource B avant la ressource A

Supposons que le programme de la tâche 2 fasse appel successivement aux ressources A et B, et libère la ressource B avant la ressource A

Si les tâches 1 et 2 sont appelées concurremment, un interblocage se produira dans le cas suivant :

1. la tâche 1 obtient A
2. la tâche 2 obtient B
3. la tâche 1 demande B mais ne peut l’obtenir tant que la tâche 2 de l’a pas libérée
4. la tâche 2 demande A mais ne peut l’obtenir tant que la tâche 1 ne l’a pas libérée
5. les 2 tâches sont bloquées en attente l’une de l’autre

Le 2ème projet s’est orienté vers des choix plus simples pour atteindre la fiabilité requise.

Tous les processus ont été découpés en tâches élémentaires comportant un nombre d’instructions réduit pour s’exécuter dans un laps de temps court et défini. Ces tâches élémentaires ne devaient avoir qu’un point d’entrée et un point de sortie.

A chaque tâche était associé un ordre de priorité.

Un seul système de file d’attente gérait l’ensemble : tout événement donnait lieu à un  message introduit dans cette file d’attente de type « fifo » (First In First Out).

Le moniteur donnait successivement la main aux tâches dans l’ordre leur priorité en fonction des messages à traiter dans la file d’attente.

Un tel système s’est révélé bien adapté aux traitements à réaliser dans l’EDT.

Rappelons que les moyens disponibles dans le Mitra 15 étaient rudimentaires : un jeu d’instructions très réduit, travaillant sur des octets (8 bits), la taille mémoire étant de 16 kilo-octets. La programmation se faisait en langage « assembleur » spécifique de la machine.

Les développements ultérieurs ont profité de la montée en gamme des Mitra : le Mitra 115 d’abord, permettant de passer à 32 puis à 64 koctets, puis au Mitra 225.

Plusieurs mécanismes de sauvegarde en cas d’erreur ont été implantés, notamment le mécanisme de « watch dog » ou « chien de garde ». Une tâche appelée tâche de fond, dont la priorité était la plus faible, avait pour seul but de réaliser un programme cyclique de quelques instructions pour activer un « bac de commutation » externe au Mitra 15. Accessoirement, elle lançait aussi l’allumage de voyants sur le pupitre du Mitra 15. Lorsque le système était trop chargé, les voyants s’allumaient donc avec une fréquence de plus en plus faible. Et si le système se bloquait, la tâche de fond n’était plus sollicitée, les voyants restaient immobiles, et le « bac de commutation » concluait à la panne de l’EDT et donnait la main au 2ème EDT, l’ensemble étant doublé pour des raisons de disponibilité.

(Une anecdote au passage : lors d’une publication sur ce thème à la CIGRE dans les années 1980, l’interprète avait traduit « watch dog » par « terre neuve » ! Le plus surprenant est que le public de connaisseurs n’avait pas réagi… )

Ce qui précède concerne principalement la partie PC (Poste de Conduite). Qu’en était-il de la partie PA (Postes asservis ?)

Ces PA étaient des machines très spécifiques, construits par les fournisseurs d’alors, notamment Jeumont Schneider et CGEE Alsthom. La problématique générale était équivalente, mais la partie informatique restait rudimentaire comparée aux questions posées par le traitement des entrées / sorties de nature particulière : les positions ou commandes d’organes en « tout ou rien » (appelées les entrées / sorties TOR, codées sur 1 seul bit), ou les mesures codées elles sur 8 bits.

Leur fonctionnement de la partie « scheduler » était simplifié à l’extrême : un examen cyclique permanent des entrées sorties donnait lieu à envoi d’un message vers le PC lors de toute détection de changement d’état.

Le point commun des PA des différents constructeurs et du PC est à noter : il s’agissait alors d’un développement « propriétaire », les standards n’ayant pas encore fait leur apparition.