Le Cahier des Charges: Contenu, Forme et Analyse (en vue de la Formalisation)

par J.-R. Abrial (Consultant)

Juin 1998

Le Cahier des Charges: Contenu, Forme et Analyse (en vue de la Formalisation)

par J.-R. Abrial (Consultant)

Introduction

Malgré toutes les solutions proposées depuis des dizaines d'années, la construction des Systèmes Informatiques reste toujours problématique. Il faut donc se demander une fois de plus pourquoi, et tenter une fois de plus d'y remédier. De nombreux "responsables" de cet état de fait ont été montrés du doigt au cours des années. Parmi ceux-ci, il y en a un qui semble faire l'unanimité aujourd'hui: le Cahier des Charges.

Pourquoi? Peut-être ce document est-il mal rédigé (il serait tout simplement écrit en mauvais français), ou bien son contenu est-il insuffisant (il ne serait pas complet), ou bien encore n'est-il pas assez structuré pour être exploitable. Probablement un peu toutes ces raisons à la fois, et surement d'autres. En tout cas, il est maintenant reconnu par la grande majorité des acteurs concernés (ceci constitue déjà un atout positif) que le Cahier des Charges (sa rédaction, son exploitation) pose un problème, qui est d'autant plus aigu qu'il intervient dès l'origine de la construction du système.

Nous avons l'ambition d'apporter ici certaines réponses, voire certaines solutions, à ces questions. Cet article ne contiendra pas, comme il est pourtant d'usage, d'exemples illustratifs, cette lacune est comblée dans un autre document [2]. Nous ne proposerons pas non plus d'outil miracle. Nous avons voulu simplement nous en tenir à l'exposé d'un certain nombre de *principes* qui s'inscrivent d'ailleurs dans une problématique beaucoup plus large que celle du Cahier des Charges proprement dit. Une problématique qui couvre, en fait, le *processus de création/construction* à l'oeuvre dans la réalisation particulièrement complexe d'un Système Informatique. Pour cela, il nous faut nécéssairement procéder à une analyse transversale qui dépasse le cadre limité du Cahier des Charges, de façon à bien comprendre les *relations* que celui-ci entretient avec l'ensemble du processus.

Nous commencerons donc cet étude par un bref rappel des différentes phases du Cycle de Développement d'un Système Informatique. Nous en ferons ensuite une analyse critique basée essentiellement sur la difficulé d'en maitriser la cohérence,

et nous montrerons, à ce sujet, que la situation n'est pas homogène sur tout le cycle. Ceci nous conduira à nous focaliser sur la rédaction de deux documents particuliers: le Cahier des Charges et la Spécification Technique. Ces deux documents correspondent à deux phases voisines situées vers le début du cycle. Nous étudierons pourquoi les relations qu'elles entretiennent sont délicates. Nous entrerons ensuite dans l'énoncé d'une série de propositions visant à systématiser la rédaction du Cahier des Charges aussi bien dans son contenu que dans sa forme. Nous en viendrons alors à proposer l'introduction dans le cycle d'un certain nombre de phases supplémentaires, qui s'intercaleront entre la rédaction du Cahier des Charges et celle de la Spécification Technique, et dont le but est de s'assurer que le passage de l'un à l'autre s'effectue correctement et, surtout, qu'il peut être validé. Nous couvrirons enfin très rapidement la question de la construction de la Spécification Technique Formelle Structurée.

Notre analyse, comme on le constatera, sera sous-tendue, en toile de fond, par l'emploi de "Méthodes Formelles avec Preuves" telles que B [1] [3]. Nous avons explicitement ajouté la mention (inusitée) "avec Preuves" dans le but de reserrer le sens de l'expression "Méthodes Formelles", qui couvre aujourd'hui un domaine très large. Les Méthodes Formelles utilisent des langages rigoureux (formels, voire mathématiques) pour décrire des problèmes. Lorsque cette expression est accompagnée de la mention "avec Preuves", cela signifie pour nous que le formalisme n'est pas uniquement destiné à décrire ces problèmes rigoureusement, à les "formaliser" comme on dit, mais qu'il a aussi pour but d'énoncer les conditions de leur résolution, et même d'y participer à l'aide d'outils spécialisés, que l'on a pris l'habitude d'appeler des "prouveurs". Au passage, nous montrerons que l'usage de Méthodes Formelles avec Preuves n'est en rien un prérequis de notre approche, mais qu'il peut, au contraire, être plus ou moins modulé.

Il existe une bibliographie assez abondante sur le sujet. Le terme technique anglo-saxon qui correspond aux questions qui nous occupent est celui de "Requirements Engineering". Nous ne citerons qu'une seule référence [5] assez récente, qui fait un peu allusion aux Méthodes Formelles. On y trouve d'autres références qui peuvent être intéressantes. C'est un champ qui se développe assez fortement ces jours-ci à côté de celui beaucoup plus ancien consacré par le terme de "Software Engineering".

Le Cycle de Développement Étendu[1]

Étudier dans l'absolu les problèmes liés à la rédaction du Cahier des Charges d'un

[1]Nous avons précisé l'expression "Cycle de Développement" par la mention "Étendu" car nous avons ajouté aux phases bien connues du Cycle en V, telles qu'elles sont largement admises et pratiquées aujourd'hui, un certain nombre de phases amont, qui sortent de ce cadre habituel.

Système Informatique n'a pas grand sens. En effet, on ne peut tenter d'apporter certains éclairages sur cette question que si l'on sait bien à quoi sert un tel document et surtout si l'on connait bien sa place dans le Cycle de Développement.

Dans la suite, nous allons donc rapidement passer en revue les principales étapes de ce cycle, telles qu'elles sont communément admises à l'heure actuelle. Nous avons réduits ces étapes à six pour simplifier. On peut, dans la pratique, en trouver parfois davantage, qui apparaissent cependant, selon nous, commes des subdivisions de celles que nous avons retenues. Nous avons estimé que chacune de celles-ci correspondait à une fonction bien déterminée, nettement distincte de celle des autres. Nous les avons sélectionnées sur cette base.

Les Études Système

Contrairement à ce que l'on pourrait croire, la rédaction du Cahier des Charges ne constitue pas, à notre avis, la première tâche à effecter dans le cycle de développement étendu d'un Système Informatique. Ce travail est déjà pour nous la conséquence d'études antérieures, appelées Études Système, dont le but est de déterminer le *bien-fondé* de la réalisation que l'on a en tête, et même de déjà valider certaines options très générales concernant son architecture globale.

Ces études sont fondamentales. Elles portent la responsabilité de la viabilité du futur système. Elles fournissent essentiellement certaines hypothèses, mais aussi divers ordres de grandeur, et même certaines options d'architecture, qui doivent garantir que le système que l'on veut construire *peut fonctionner* tout en respectant certaines propriétés fondamentales de disponibilité (le système remplit ses fonctions correctement) et de sécurité (le système remplit ses fonctions sans danger).

Il ne doit pas, d'après nous, revenir au réalisateur du futur système de porter la responsabilité des mauvais fonctionnements du système qui seraient imputables à des erreurs portant sur les conclusions de ces études. Autrement dit, le réalisateur doit pouvoir escompter que les conclusions générales de viablité issues de ces études sont bien garanties dès lors qu'il respecte les hypothèses, les ordres de grandeur et les options architecturales qui ont conduits à ces conclusions. Ces études se doivent donc d'apporter une certaine "preuve" de leurs conclusions. Il est clair que, pour nous, ces études "système" ne sont pas menées exclusivement par des informaticiens, mais bien par des équipes mixtes comprenant aussi des ingénieurs "système" rompus aux techniques du métier concerné: automaticiens, hydrauliciens, thermiciens, aérodynamiciens, voire physiciens, etc.

Il ne rentre pas dans le cadre du présent document de proposer une méthodologie apte à faciliter la conduite de ces études. Disons simplement qu'elles consistent, d'après nous, à effectuer un certain nombre de *simulations* (à très haut niveau) du

comportement du futur système *et de son environnement matériel*. Ces "simulations" peuvent être réalisées soit à l'aide d'authentiques programmes de simulation, soit à l'aide de modèles mathématiques abstraits, soit, bien entendu, à l'aide des deux approches menées simultanément.

Dans le premier cas, les résultats de l'exécution du programme de simulation montrent, ou au contraire infirment, que le système que l'on a en tête est viable au regard des *paramètres* que l'on a introduits. Dans le deuxième cas, on peut être amené à faire une véritable preuve de la viabilité du système, cette fois-ci par rapport à un certain nombre *d'hypothèses* que l'on a formalisées.

La deuxième approche est plus satisfaisante que la première mais elle n'est guère pratiquée que dans des cas relativement simples où l'on a un modèle analytique, c'est-à-dire essentiellement régi par des lois physiques. Les preuves associées à ce type de modèles se traitent généralement de façon analytique "sur le papier" ou à l'aide de logiciels dédiés. Dans les autres cas, on s'est heurté jusqu'à présent à des problèmes de complexité que l'on n'a pas su résoudre de façon satisfaisante. En effet, on appréhende dans ces modèles des sytèmes qui sont essentiellement *distribués et asyncrones*. Nous pensons que les Méthodes Formelles avec Preuves basées sur la technique du raffinement peuvent apporter une aide significative à ce niveau. Nous serons amenés à préciser nos vues sur ce sujet dans un autre document.

Lorsque ces Études Systèmes sont terminées, on peut estimer que les grandes options techniques et les principaux paramètres du futur système, *dans sa globalité logicielle et matérielle*, sont maintenant fixés. En particulier, l'organisation du système en sous-systèmes, dont les relations mutuelles sont bien clarifiées, est probablement déjà plus ou moins stabilisée. On peut donc se diriger vers la rédaction des Cahiers des Charges.

Le Cahier des Charges

Le Cahier des Charges d'un Système Informatique est constitué par l'ensemble des documents qui rassemblent, du point de vue du client, tous les *éléments techniques*, à partir desquels un futur contractant va pouvoir réaliser le système en question.

À côté de cette finalité technique, le Cahier des Charges remplit aussi souvent un *rôle contractuel*: il contient en fait une certaine définition "légale" du système à réaliser. Cela signifie que tout litige, qui pourrait apparaitre entre le client et le contractant au sujet du système final, devrait pouvoir être tranché, en dernière analyse, après consultation du Cahier des Charges.

Cette double caractéristique (technique et contractuelle) du Cahier des Charges justifie son importance et donc le soin que l'on doit apporter à son écriture.

La Spécification Technique

L'étape suivante est celle de la rédaction de la Spécification Technique. Cette tâche est la première dans laquelle soit engagé le réalisateur du système. Elle consiste à reprendre, d'une façon ou d'une autre, tous les "éléments" du Cahier des Charges, pour en faire une analyse exhaustive débouchant sur un document beaucoup plus précis, qui peut être partiellement, ou même totalement, formalisé. Dans ce dernier cas, on peut être amené à effectuer une preuve de cohérence qui va donc constituer une sorte de *validation a posteriori* du Cahier des Charges. À la fin de cette étape, l'architecture du système se précise davantage sans toutefois qu'aucun programme ne soit encore écrit.

La Conception du Système

C'est au cours de la phase suivante, dite de conception (générale puis détaillée), que l'on prend un certain nombre de décisions très importantes concernant la *décomposition* du système en modules informatiques séparés. Cette décomposition permet éventuellement de sous-traiter la suite du travail à des équipes distinctes travaillant en parallèle. On y fait aussi des choix précis en matière *d'organisation des données*.

Le Codage

Nous n'insisterons pas sur cette étape bien connue à propos de laquelle de nombreuses approches méthodologiques ont déjà été proposées. Disons simplement ici que la réussite de cette phase est grandement conditionnée par la qualité du travail effectué dans les tâches situées plus en amont.

Les Tests

Dans les descriptions ci-dessus, nous n'avons parcouru que la "branche de gauche", descendante, du fameux cyle en V. Comme on le sait, la "branche de droite", montante, contient des phases de tests, situées en face des étapes correspondantes de la branche voisine.

Résumé

Nous avons rapidement passé en revue les principales étapes classiques de la construction d'un système informatique. Les phases que nous avons retenues sont les suivantes:

1. les Études Système,
2. le Cahier des Charges,
3. la Spécification Technique,
4. La Conception,
5. Le Codage,
6. Les Tests.

Analyse Critique du Cycle de Développement Étendu

La Maitrise des Phases du Cycle de Développement Étendu

Une des difficultés majeures de la maitrise du cycle de développement est, à notre avis, celle de la *garantie du respect des phases amont par les phases aval.* La démarche, que nous venons d'esquisser très succintement ci-dessus, a en effet pour but d'affiner petit à petit la construction du système par une série de phases dont le rôle est de plus en plus précis. Mais le bien-fondé de cette approche repose, évidemment, sur la certitude que chaque phase est bien cohérente avec la précédente, c'est-à-dire qu'elle n'introduit pas d'éléments qui seraient en contradiction avec elle.

Comment être sûr, par exemple, que le code final est fidèle aux décisions de conception effectuées au préalable? Comment avoir la garantie que la conception est elle-même en accord avec la spécification technique qui la précède? Il faut bien dire que, jusqu'à une période assez récente, les réponses que l'on pouvait apporter à ces questions n'étaient guère satisfaisantes.

La Solution Apportée par les Méthodes Formelles avec Preuves

L'utilisation de Méthodes Formelles avec Preuves apporte certains éléments de réponses aux questions posées ci-dessus. Ces méthodes ont d'ailleurs été développées en grande partie pour cela. La raison en est que le passage d'une étape à la suivante est maintenant garanties par un ensemble de *preuves mathématiques.*

Ceci ne concerne cependant que les phases basses du cycle (spécification technique, conception, codage). Les concepts fondamentaux qui sont mis en oeuvre par ces méthodes sont ceux de *raffinement* et de *décomposition*. À noter que ces techniques peuvent aussi être utilisée à l'intérieur d'une même phase, qui se structure alors en sous-phases liées par les relations de raffinement et de décomposition.

Par exemple, la construction de la Spécification Technique Formelle peut être réalisée par une suite de raffinements consistant à extraire *petit à petit* les détails du Cahier des Charges pour les incorporer dans une Spécification Technique Structurée. Au cours de ce travail, on peut être amené à scinder la Spécification Technique en plusieurs morceaux relativement indépendants par décomposition, ces morceaux étant eux-mêmes ensuite raffinés (de nouveaux détails sont alors extraits du Cahier des Charges), puis de nouveau décomposés, etc.

De même, la Conception Formelle peut-elle être réalisée par une suite de raffinements consistant à prendre *petit à petit* des décisions d'implantation qui déterminent de plus en plus la construction du système en l'entrainant dans la direction du programme final. En faisant cela, on peut être amené, comme ci-dessus pour la Spécification Technique, à scinder le système en morceaux relativement indépendants. À noter que ces morceaux pourraient éventuellement correspondre à des sous-sytèmes déjà étudiées, voire déjà complétement codés et prouvés. Mais méfions-nous ici d'être trop enthousiaste: la réutilisation, dont on a tant parlé, n'ayant pas vraiment reçu jusqu'ici de véritable confirmation pratique. Pourquoi?

À chaque étape de raffinement, des preuves sont exigées, qui ont pour but, non seulement de valider la cohérence de ce que l'on est en train d'introduire, mais aussi de montrer que l'on ne contredit pas le travail effectué jusque là.

Ainsi donc, si ces preuves sont effectivement réalisées (et ce, à l'aide d'outils fonctionnant en grande partie de façon automatique) il n'y a plus de place pour une quelconque distorsion entre les différentes phases que nous avons retenues. Bien sûr, ceci nécéssite que ces phases soient décrites à l'aide d'un formalisme *homogène* très rigoureux (un langage formel manipulant des concepts mathématiques éprouvés), de façon à ce que les conditions, que l'on cherche à prouver pour garantir que chaque phase est bien fidèle à la précédente, ne souffrent pas d'ambiguités.

L'Utilisation des Méthodes Formelles avec Preuves

Ces méthodes ont, bien entendu, leurs limites d'utilisation, essentiellement dues, d'ailleurs, à l'effort de preuves qu'elles sous-tendent. Par exemple, à l'heure actuelle, l'expérience montre que le développement complet d'un module de n lignes de code nécéssite grosso-modo la preuve de $n/2$ petits lemmes. Si 80% de ces lemmes peuvent être prouvés de façon automatique par un outil comme le prouveur au-

tomatique de l'Atelier B [6], les 20% restants nécéssitent un effort de preuves supplémentaire (effectué, par exemple, à l'aide du prouveur interactif de l'Atelier B), qui est loin d'être négligeable: on estime qu'un expert prouve, en effet, avec l'aide de cet outil, une quinzaine de lemmes par jour en moyenne.

Ces éléments montrent que l'on ne peut pas appliquer de telles méthodes sur une trop grande échelle, du moins au niveau d'un seul module: on estime qu'un programme final d'une centaine de milliers de lignes de code représente à peu près les limites de l'état de l'art. Bien entendu, lorsqu'un système se compose de plusieurs sous-systèmes, cette limitation porte sur chaque sous-système, non sur l'ensemble. On voit donc toute l'importance de la mise en oeuvre par B du concept de *décomposition*, auquel nous avons fait allusion plus haut. Par ailleurs, il faut bien préciser que, lorsque l'on s'engage dans cette voie, on n'est pas obligé de développer l'intégralité d'un système de cette façon. On peut se contenter de le faire sur les parties les plus sensibles, pour lesquelles une garantie de bon fonctionnement est absolument nécéssaire.

Le Problème des Phases Amont

L'usage des Méthodes Formelles avec Preuves peut donc apporter, comme nous venons de le voir, une aide relativement satisfaisante en ce qui concerne la maitrise des phases basses du cycle de développement (spécification technique, conception, codage). Mais qu'en est-il des phases hautes? En particulier qu'en est-il du passage du Cahier des Charges à la Spécification Technique?

Entre ces deux phases, en effet, on passe de textes écrits dans un langage qui n'est certainement pas formel dans le cas du Cahier des Charges, à des textes qui sont écrits dans un langage qui est d'une toute autre nature dans le cas de la Spécification Technique. Il n'y a donc *pas d'homogéneité* entre ces univers linguistiques. Ceci rend alors impossible l'usage de Méthodes Formelles avec Preuves.

Et pourtant, le client, qui a rédigé le Cahier des Charges, voudrait bien avoir une certaine garantie que ce qu'il a écrit a bien été transcrit correctement dans la Spécification Technique. Il voudrait aussi savoir si ce qu'il a spécifié informellement était bien cohérent. La phase préalable des Études Système lui donne déjà une certaine assurance, mais rien ne l'empêche d'avoir ajouté des contraintes nouvelles qui pourraient être contradictoires avec d'autres. Il peut aussi se poser des questions quant à la complétude de son travail.

Le Maillon faible

On peut donc dire que le passage du Cahier des Charges à la Spécification Technique constitue effectivement désormais le *maillon faible* du cycle de développement. Il est intéressant de noter à ce sujet que ce maillon faible se situait, il y a quelques années encore, entre la spécification technique et les phase situées plus en aval (conception et le codage). Ce maillon faible est donc remonté avec le temps grâce, notament, à la mise en oeuvre *industrielle* des Ateliers de Génie Logiciel. L'usage de ces Ateliers a considérablement affermi ce maillon, cependant ils ne garantissent pas au moyen d'une *démonstration rigoureuse*, comme le font les Méthodes Formelles avec Preuves, que ces phases sont cohérentes entre elles.

Lorsque la pratique de la rédaction du Cahier des Charges que nous préconisons sera, d'une façon ou d'une autre, rentrée dans les moeurs, le maillon faible pourra être encore remonté. Il se situera alors au niveau des Études Système et de leurs liaisons avec le Cahier des Charges. Nous pensons qu'il y a encore beaucoup à faire de ce côté là, en particulier en ce qui concerne le partage des responsabilités entre ce qui relève de ces Études Système et ce qui relève du Cahier des Charges proprement dit.

À propos du Cycle en V

Le concept de cycle en V, auquel nous avons fait allusion ci-dessus, peut être significativement reconsidéré lorsque l'on adopte l'utilisation de Méthodes Formelles *avec Preuves*. Dans ce cas en effet, on peut envisager de supprimer les phases de tests associées aux parties basse du cycle, c'est-à-dire la conception et le codage. Car, pour ces phases, on estime que la preuve (si elle est effectivement sanctionnée par l'usage d'un outil très sûr) offre une bien plus grande garantie d'exhaustivité que celle procurée par les tests, qui ne seront jamais qu'une approximation. Par contre, il est clair que les tests globaux (fonctionnels) qui, eux, valident les phases hautes du cycle, c'est-à-dire la Spécification Technique et le Cahier des Charges, sont évidemment toujours d'actualité.

À noter cependant, et c'est toute la thèse défendue dans la suite de ce document, que l'approche proposée ici tend à anticiper cette phase de tests globaux, qui est *postérieure* à la réalisation du système, par une réflexion et une action complétement *antérieures* à celle-ci. N'oublions pas, en effet, que dans le cycle en V, les tests globaux sont situées en haut et à droite du V et ne sont donc effectués qu'à la fin du cycle. Il peut sembler curieux que ces tests, qui valident en quelque sorte les grandes options du système, soient pratiqués *en dernier*. Cette validation ne devrait-elle pas, au contraire, avoir été effectuée, dans la mesure du possible, *en premier*?

Il faut noter toutefois que la pratique du *prototypage*, qui se développe depuis un certain temps déjà, a pour but de "visualiser", en quelque sorte, ce que le client désire. Elle apporte donc certains renseignements sur le bien-fondé de ce qui est demandé, sans, toutefois, le *valider* en profondeur. Or c'est bien cela que nous avons l'ambition de proposer ici.

Conclusion

Nous avons désormais planté le décor du problème que nous nous proposons d'aborder dans la suite de ce document. Il peut se résumer ainsi: Comment rédiger le Cahier des Charges et aussi la Spécification Technique de façon à ce que le passage de l'un à l'autre soit, dans une large mesure, rendu tout d'abord simplement possible, mais aussi *controlable* par les deux parties en présence, à savoir le client et le réalisateur?

Corrélativement, nous voudrions aussi savoir dans quelle mesure il serait possible de garantir une certaine cohérence, voire une certaine complétude, du Cahier des Charges, et ce par l'intermédiaire d'une Spécification Technique formellement prouvée dont on aurait une certaine confiance (à défaut d'une véritable preuve) qu'elle serait fidèle aux prescriptions de celui-ci.

Résumé

Nous avons examiné

1. la difficulté du problème lié à la cohérence relative entre les phases successives du cycle de développement,
2. les possibilités offertes par les Méthodes Formelles avec Preuves pour résoudre cette difficulté dans les phases basses du cycle de développement,
3. les difficultés résiduelles concernant les phases hautes de ce cycle,
4. la position du maillon faible dans le cycle de développment étendu,
5. les conséquences de l'utilisation de Méthodes Formelles avec Preuves sur la nature du cycle en V.

Options d'Utilisation des Méthodes Formelles avec Preuves

Dans les paragraphes ci-dessus, nous avons souvent parlé d'utiliser des Méthodes Formelles avec Preuves au cours de certaines phases du cycle de développement. Il

serait peut-être bon, avant d'aborder la suite, de se demander si cet usage relève du "tout ou rien" ou si, au contraire, on peut considérer qu'il existe différentes options permettant de le moduler. Nous pensons que la deuxième alternative est raisonnable. Dans ce qui suit, nous considérons donc différentes possibilités et nous les analysons brièvement.

Nous avons retenu six options, qui sont schématisées dans le tableau ci-dessous. Chaque option correspond à une colonne dont les cases correspondent aux différentes phase du cycle. Le symbole "$\surd$" signifie que la phase correspondante du cycle est effectivement réalisée, dans l'option concernée, à l'aide d'une Méthode Formelle avec Preuves. Le symbole "+" signifie que le Cahier des Charges est rédigé, dans l'option concernée, en suivant les techniques que nous proposons ci-après.

OPTIONS	1	2	3	4	5	6
Études Système				$\surd$	$\surd$	$\surd$
Cahier des Charges	+	+	+	+	+	+
Spécification Technique		$\surd$	$\surd$		$\surd$	$\surd$
Conception & Codage			$\surd$			$\surd$

Les différentes options d'utilisation de Méthodes Formelles avec Preuves

Ce tableau pourrait contenir davantage d'options. Nous n'avons cependant retenus que celles-ci car il existe un certain nombre de contraintes. Nous avons admis, en effet, que lorsque la Spécification Technique était rédigée en utilisant une Méthode Formelle avec Preuves, il était impératif que le Cahier des Charges soit lui-même rédigé très soigneusement en utilisant les techniques que nous proposons. Autrement dit, lorsque l'on trouve le symbole "$\surd$" dans la ligne Spécification Technique d'une certaine option, on trouve nécéssairement le symbole "+" dans la ligne Cahier des Charges de cette option. Nous avons groupé conception et codage dans une rubrique unique car on ne voit pas très bien ce que cela signifierait

d'entreprendre la formalisation de la conception (avec les preuves correspondantes) sans pousser jusqu'au code. Enfin, bien entendu, la formalisation de la conception et du codage n'ont pas de sens sans une spécification technique, elle-même formalisée. Autrement dit, lorsque l'on trouve le symbole "$\surd$" dans la ligne Conception & Codage d'une certaine option, on trouve aussi ce symbole dans la ligne Spécification Technique de cette option.

La présence des options 4,5 et 6, qui ne diffèrent des options 1,2 et 3 que par l'utilisation pour les Études Système de Méthodes Formelles avec Preuves, est le reflet de l'indépendance de cet usage sur le reste du cycle de développement.

L'option 1 est un peu batarde, mais pas inintéressante pour autant. Elle consiste à porter un effort important sur la rédaction du Cahier des Charges, mais à ne pas poursuivre du tout vers une quelconque formalisation, en espérant que cet effort initial sera tout de même suffisamment payant. L'option 2 représente le cas intéressant, où l'on n'utilise la Méthode Formelle avec Preuves que dans le cadre de la Spécification Technique. En fait, cet investissement vise essentiellement à valider le Cahier des Charges au moyen d'une Spécification Technique Formelle prouvée. On poursuit ensuite la construction du système par des moyens habituels. L'option 3 représente le cas le plus riche, où l'on a décidé d'investir à fond dans cette technologie. On peut aller encore plus loin en prenant l'option 6. Les options 2 et 3 sont celles que nous recommendons désormais. À noter qu'elles complémentent les options pratiquées aujourd'hui, à savoir, les options 7 et 8, que nous mentionnons ci-après pour mémoire.

OPTIONS	7	8
Études Système		
Cahier des Charges		
Spécification Technique	$\surd$	$\surd$
Conception & Codage		$\surd$

Retour sur le Cahier des Charges[2]

Difficultés de la Rédaction du Cahier des Charges

L'observation courante des CDC que l'on voit circuler dans la pratique montre que ces documents sont souvent d'une qualité qui n'est pas à la hauteur des enjeux qu'ils représentent. Et l'on constate que, bien souvent, les systèmes réalisés à partir de là ne répondent pas non plus aux attentes des clients.

On doit cependant constater, à la décharge des rédacteurs de CDC, que la question posée par la bonne rédaction de tels documents n'est pas simple à résoudre. Quelles sont, en effet, les critères qui font d'un CDC un "bon" CDC ? Existe-t-il des approches systématiques permettant de rédiger des CDC qui soient satisfaisants à la fois pour le client (dire l'essentiel) et pour le contractant (ne dire que l'essentiel) ?

Il est clair, en effet, que le CDC doit contenir l'énoncé de tous les critères susceptibles de caractériser la bonne marche du système qu'il est censé décrire, ainsi que ses limites de fonctionnement. Il est fréquent de trouver, malheureusement, des CDC qui, de ce point de vue, sont particulièrement défaillants. Il se trouve, par exemple, des CDC qui ne mentionnent même pas pas à quoi sert le système en question. D'ailleurs, les rédacteurs de tels CDC, voire ses lecteurs, semblent, bien souvent, ne même pas s'apercevoir de l'existence de ces lacunes pour la simple raison que les propriétés importantes qui manquent sont "tellement évidente pour tout le monde" que l'on a simplement oublié de les mentionner.

Mais un CDC ne doit pas contenir non plus une collection trop importante de détails superflus. Ceux-ci risquent en effet de cacher, voire d'occulter complétement, les éléments importants du sytème. De tels détails risquent aussi de contraindre trop fortement le contractant, l'empêchant par là même de proposer des solutions intéressantes. Mais il est vrai que l'importance éventuelle d'une propriété n'est pas si évidente que cela à déterminer a priori: le rédacteur consciencieux peut donc avoir tendance à en mettre trop.

Problèmes Posés par l'Introduction des Méthodes Formelles avec Preuves

L'apparition des Méthodes Formelles avec Preuves augmente encore l'importance du CDC. Celui-ci constitue, en effet, le point de départ naturel de l'écriture de la Spécification Technique Formalisée, à partir de laquelle le système va ensuite être conçu, architecturé, puis réalisé de façon complétement systématique jusqu'au code final inclus.

[2]Par soucis de brièveté, nous utiliserons dans la suite de ce document l'abréviation CDC pour désigner le Cahier des Charges

Notre opinion, qui va sans doute apparaitre paradoxale à certains, est que plus l'on désire formaliser (mathématiquement, c'est-à-dire avec preuves à l'appui) la construction d'un système informatique, plus le CDC (écrit en français) doit être particulièrement soigné. Il en va, en fait, du succès de l'entreprise. Au point que l'on peut même se demander si c'est, en fin de compte, à la formalisation proprement dite que l'on doit la réussite des sytèmes construits de cette façon, ou bien si celle-ci est due, en réalité, à l'intensité et à la qualité du travail de réflexion et de rédaction qui ont présidé à l'élaboration du CDC. Il est certes difficile de quantifier ce genre d'arguments, nous sommes néanmoins persuadés que ce travail très important, effectué *en amont* de la réalisation technique, fournit une sérieuse option quant au succès final. Donc, suivi ou non de formalisation, la qualité du CDC est, à notre avis, un atout très sérieux à considérer.

Mais il faut reconnaitre aussi que la formalisation mathématique fournit son apport personnel, très important, à la qualité du résultat: il est essentiellement le fait de la *preuve formelle*, qui peut accompagner tout ou partie du dévelopement jusqu'au code final, et dont les exigences jouent un rôle prépondérant dans la clarification de l'architecture du système.

Difficultés d'Établissements de Liens Significatifs entre le Cahier des Charges et la Spécification Technique

Le "formalisateur" initial s'appuie donc, comme nous venons de le voir, sur le CDC pour construire progressivement sa première vision technique du système. Il est clair que cette approche, de par sa nature même, pousse à transcrire le plus directement possible les exigences du CDC dans la Spécification Technique. Il en résulte tout naturellement que les propriétés manquantes ne seront évidemment pas transcrites et, inversement, que les propriétés trop concrètes (voire superflues) auront tendance à rendre cette spécification technique passablement compliquée et illisible.

Mais il y a plus grave. En effet, l'expérience montre que la grande majorité des CDC que l'on trouve dans la pratique (même les "bons") est très mal adaptée à la suite du travail, sans parler même de l'utilisation éventuelle de Méthodes Formelles avec Preuves. Il se trouve qu'il est souvent très difficile de constituer des passerelles entre le CDC, écrit en français, et la Spécification Technique, écrite "en formel": *la distance est trop grande entre ces deux univers linguistiques.* Par exemple, le formalisateur est souvent très mal à l'aise pour savoir comment extraire du CDC les propriétés qu'il doit incorporer dans la spécification technique. Et, quelle que soit la qualité du travail de formalisation, le client éprouve, quant à lui, les plus grandes difficultés à valider la spécification technique qu'on lui présente par rapport au CDC qu'il a écrit: il ne sait pas du tout comment rapprocher les deux textes.

Prenons donc acte de la difficulté considérable du rapprochement direct entre la Spécification Technique Formelle et le CDC tel qu'il est. Et considérons qu'une partie de la distance entre les deux textes pourrait probablement être couverte avec plus de profit par le CDC lui-même plutôt que par la Spécification Technique. Cette approche consiste donc essentiellement à effectuer une *deuxième formulation* du CDC en vue de le préparer à la formalisation future. Comme nous l'avons indiqué plus haut, on peut cependant s'engager dans cette nouvelle rédaction sans penser à poursuivre par une Spécification Technique Formelle.

Il faut noter que cette nouvelle formulation n'est pas autre chose, en fait, que ce que certains appellent la Spécification Générale (SG). En toute rigueur, nous aurions donc du incorporer cette phase dans la description que nous avons faites ci-dessus du cycle de développement étendu, puisqu'il est communément admis qu'elle lui appartient bien à part entière. Nous ne l'avons pas fait cependant car les deux textes que constituent le CDC et la SG sont, en fait, *fonctionnellement voisins*. En effet, ils sont tous les deux écrits en français, ils décrivent tous les deux ce que le système doit faire (ses exigences) sans (trop) préjuger de la façon dont il va être réalisé, enfin ils constituent tous les deux, à des stades différents, la base sur laquelle s'appuie le rapport formel entre le client et le fournisseur. À noter enfin que cette reformulation aurait pu être évitée si le CDC se trouvait *déjà* sous une forme adéquate: avis aux donneurs d'ordre et à leurs consultants!

Cette approche présente un certain nombre d'avantages. Elle offre, entre autres, la possibilité d'impliquer le client. Autrement dit, le CDC n'est évidemment pas rejeté, il constitue même le point de départ de la rédaction de la SG. Celle-ci implique évidemment aussi la participation du réalisateur, qui, d'ailleurs, tient, cette fois-ci, la plume. Il s'agit donc d'un travail mené en collaboration entre les parties intéressées. À noter que la participation du client, sous une forme ou sous une autre, doit permettre de lever plus facilement, par questionnement direct, les ambiguités du CDC et aussi de combler ses insuffisances. Ces ajustements ne sont pas trop difficiles à effectuer car la question, en effet, est maintenant de préciser le contenu d'un premier texte écrit en français, le CDC, à l'aide d'un second, la SG, écrite elle aussi en français: *on est bien dans le même univers linguistique*. À noter que la SG va constituer, une fois sa rédaction terminée et approuvée par le client, une nouvelle base qui remplace la première, que l'on peut dès lors oublier.

Comme nous allons le voir, la rédaction de la SG va s'effectuer en observant un certain nombre de règles, qui vont permettre, non seulement de formaliser relativement facilement ce nouveau texte (elles sont faites pour cela), mais qui vont avoir aussi l'effet bénéfique de pouvoir ensuite le structurer de façon assez systématique.

Il n'est donc pas interdit de penser que l'on peut suivre ces règles même si l'on n'envisage pas à terme d'utiliser de Méthodes Formelles avec Preuves.

Résumé

Nous avons constaté

1. que le Cahier des Charges était un élément très important de la construction d'un système informatique,
2. que la rédaction d'un "bon" Cahier des Charges était une tâche difficile,
3. que le texte d'un Cahier des Charges, même bien fait, s'adaptait en général assez mal à son analyse en vue du travail futur,
4. qu'il était en général nécéssaire de procéder à une reformulaton du Cahier des Charges, sous la forme d'une Spécification Générale écrite, elle aussi, en français,
5. qu'il pouvait exister des règles dont l'application permettait de mieux adapter cette Spécification Générale à sa formalisation future,
6. que l'observation de ces règles avaient probablement une valeur en soi indépendante d'une éventuelle formalisation avec preuves.

Principe de Base de la Rédaction de la Spécification Générale

Dans la suite de ce document, nous allons donc détailler les règles que nous venons d'annoncer. Mais, auparavant, il nous faut d'abord considérer le principe fondamental sur lequel est basée la rédaction de la SG.

Les Deux Textes Constitutifs de la Spécification Générale

Le principe de base de la rédaction de la SG en vue de sa formalisation future consiste à rédiger ce document sous la forme de *deux textes imbriqués*.

La SG se présente de prime abord (et ce, au bon vouloir du rédacteur) sous la forme de plusieurs opuscules, contenant chacun des chapitres, paragraphes, etc, composés, eux mêmes, de textes libres. Ceci constitue le *texte explicatif*, qui n'obéit, comme on le voit, à aucunes contraintes particulières.

Au milieu du texte explicatif, on peut trouver des parties qui forment ce que l'on appelle le *texte de référence*[3]. Les divers fragments de textes de référence doivent pouvoir être facilement identifiés comme tels par le lecteur. Ils doivent aussi pouvoir être extraits sans difficulté du texte explicatif au moyen d'un outil. Ils sont, pour cela, référencés à l'aide d'un système de repérage autonome qui est *indépendant* de celui utilisé pour le texte explicatif (celui des chapitres, paragraphes, etc). Ce repérage suit donc une logique propre, un peu analogue à celle qui est utilisée pour identifier les documents iconographiques d'un livre.

Le Texte Explicatif

Le texte explicatif est essentiellement consacré à l'exposé du besoin. Il est dicté par le désir du rédacteur de faire partager au lecteur la *compréhension* qu'il a du problème. Ce texte a pour but de faciliter la lecture de la SG lors des premières prises de contact, qui peuvent s'effectuer, en principe, en sautant les textes de référence. À noter que des parties substantielles de la SG peuvent consister en une copie pure et simple de certaines parties du CDC originel.

Le Texte de Référence

Le texte de référence constitue, comme son nom l'indique, la *référence* du futur système. Autrement dit, en séparant, le texte de référence du texte explicatif, et ce *une fois que la compréhension en est bien claire*, on obtient un ensemble, certes aride, mais qui expose, sous une forme condensée *tout ce que l'on doit savoir* pour réaliser le futur système. Dans cette optique, le texte explicatif peut, à ce moment-là, être complétement oublié puisqu'il n'apporte rien de plus. Si, malgré tout, ce n'était pas le cas, cela signifierait simplement que la séparation entre les deux textes n'a pas été faite correctement et qu'il faut donc revoir le travail de rédaction.

Relations entre les Deux Textes

Il est clair que les textes explicatifs et de référence présentent certaines *redondances*, qui sont précisément là pour faciliter la compréhension. En fait, sans le texte explicatif, le texte de référence serait par trop indigeste et probablement incompréhensible. Par contre, lorsque la compréhension est acquise, le texte explicatif devient lourd et inutile.

[3] À noter, comme on le verra plus bas, que sous ce vocable on entend ici aussi bien des textes proprement dit, que des tableaux, des diagrammes, des schémas, voire des formules, etc

En bref, le texte de référence expose le "quoi" du système (lequel sera repris formellement dans la spécification technique). Le futur programme contiendra évidemment le "comment" de ce système. Le texte explicatif, quant à lui, dit pour commencer quel est, en quelque sorte, son "pourquoi".

Une Analogie Intéressante

On pourrait faire une analogie entre cette organisation en deux textes imbriqués et celle que l'on trouve dans les ouvrages de mathématiques. Dans ceux-ci, en effet, les définitions, les lemmes et les théorèmes (qui sont d'ailleurs effectivement pourvus d'un repérage autonome) correspondraient à ce que nous avons appelé le texte "de référence", tandis que les explications intuitives, voire les démonstrations, formeraient plutôt le texte "explicatif".

Il est intéressant de noter, à ce sujet, que dans les traités encyclopédiques de Bourbaki, on trouve des fascicules séparés, appelés "Fascicules de Résultats", qui sont très exactement constitués de la somme des différents textes de référence (définitions, théorèmes) que l'on a extrait du corpus général. Leur raison d'être est claire: ils forment un tout commode, une sorte de formulaire généralisé, auquel on peut se référer lorsque l'on a bien compris l'ensemble ; ils concrétisent exactement ce que le professionnel doit savoir pour continuer à pratiquer les mathématiques dans le domaine en question sans être gêné par le lourd appareil des preuves et des explications intuitives devenus pour lui inutiles, mais qui sont, bien entendu, indispensables au débutant.

Résumé

Nous avons

1. partitionné la SG en deux catégories de textes facilement identifiables: les textes explicatifs et les textes de référence,
2. indiqué que les texte explicatifs présentaient le "pourquoi" du système, essentiellement orientés donc vers sa compréhension intuitive,
3. indiqué aussi que les textes de référence en présentaient le "quoi", c'est-à-dire qu'ils rassemblaient tout ce dont le réalisateur devait exclusivement avoir besoin pour réaliser le futur système.

Le Contenu du Texte de Référence

Dans la suite, nous allons affiner ce que nous entendons précisément par le "quoi" présenté dans le texte de référence, d'abord sous l'angle de son contenu dans cette partie-ci, puis sous celui de sa forme dans la suivante.

Les Propriétés

Les textes de référence, dont est émaillé la SG, contiennent essentiellement l'énoncé des propriétés que le futur système se doit de satisfaire et *dont le réalisateur se doit d'apporter la preuve qu'il en est bien ainsi.*

À noter que, d'après nous, les propriétés énoncées dans la SG, *ne comprennent pas* celles qui découlent des conclusions des Études Système auxquelles nous avons fait allusion plus haut. On ne saurait, en effet, imputer au réalisateur du système de porter la responsabilité de garantir des propriétés qui découlent d'études qu'il n'a pas menées lui-même, et qui ne sont ni de son ressort, ni bien souvent de sa compétence. Par contre, bien entendu, les hypothèses, ordres de grandeur et options architecturales qui rentrent dans les prérequis des conclusions de ces Étude Système constituent maintenant des *propriétés* à part entière que le réalisateur doit garantir et qui, donc, rentrent dans celles que la SG doit décliner.

Toutes ces propriétés du système sont donc incorporées (sous forme de textes de référence) au fur et à mesure qu'elles se présentent au fil de la rédaction. Le rédacteur de la SG doit donc constamment se poser la question de savoir si ce qu'il écrit ne relève que du texte explicatif ou bien si, au contraire, il s'agit d'une information qui est indispensable à la réalisation du système, auquel cas elle doit être incorporée sous la forme d'un texte de référence.

Pour ne pas briser la continuité du texte explicatif, on pourra parfois être amené à trouver dans ce dernier des informations qui doivent aussi se trouver dans le texte de référence. En fait, comme nous l'avons déjà dit, une certaine redondance ne nuit pas. Il est bon à ce moment-là de trouver seulement une césure naturelle dans le texte explicatif pour y incorporer une certaine synthèse condensée dans une enclave formant un petit texte de référence.

Niveaux d'Abstraction

À notre avis, le rédacteur ne doit pas trop se poser de questions quant à la "valeur" de l'information qu'il est prêt à mettre dans un texte de référence à un moment donné. Il importe peu, en effet, qu'il s'agisse d'un détail mineur ou d'une propriété vitale du système, ce type d'analyse viendra plus tard.

Dans le même ordre d'idée, le rédacteur ne doit pas se soucier outre mesure du *niveau d'abstraction* de l'information en question. On sait que cette préoccupation agite beaucoup les esprits: "Suis-je au bon niveau d'abstraction ?", "Vous n'êtes pas au bon niveau d'abstraction", voilà une question et un reproche que l'on entend souvent. Au point où nous en sommes (la rédaction de la SG), ce n'est pas une question pertinente, car, en effet, on n'en sait rien. Ce n'est que lorsque la rédaction de la SG sera terminée qu'une analyse "transversale" du texte de référence permettra de construire une hiérarchie des propriétés, hiérarchie qui, d'ailleurs, conditionnera leur entrée progressive dans la Spécification Technique Formelle Structurée.

Le Problème de la Sur-spécification

Un question plus délicate, qui peut éventuellement gêner le rédacteur de la SG, est celle de la *sur-spécification*. Pour des raisons contingentes, le client désire parfois, en effet, imposer au réalisateur certaines contraintes qui semblent très particulières (concrètes). Par exemple, dans un système où il y a beaucoup de communications sur un réseau, le client peut vouloir imposer l'utilisation d'un certain protocole de transmission bien défini.

Que faire de ces exigences ? Doit-on les écarter dans un premier temps, ou bien faut-il au contraire les incorporer dans le texte de référence au motif que, manifestement, elles ne peuvent pas être ignorées puisque le client veut les voir représentées dans le système final ? La réponse est claire, il faut les incorporer. Mais, ce faisant, le rédacteur se doit de *signaler* par une convention connue qu'il estime qu'il s'agit manifestement d'une sur-spécification.

La meilleure preuve de ce fait pourrait d'ailleurs en être administrée par l'invention d'une propriété *plus abstraite* dont la sur-spécification en question serait un cas particulier. Dans l'exemple du protocole de transmission évoqué ci-dessus, la propriété plus abstraite en question pourrait correspondre au fait qu'il s'agit seulement d'un protocole de transfert opérant à l'intérieur de certaines contraintes de fonctionnement, énoncé effectué à ce niveau sans aucune référence à un protocole particulier.

En fait, comme nous le verrons par la suite, il reviendra plutôt à la phase d'analyse du texte de référence de proposer une telle abstraction, qui aura pour but de surseoir à la prise en compte prématurée de cette exigence concrète dans le processus de construction du système. Autrement dit, cette exigence ne sera pas écartée de la réalisation, son entrée en scène en sera seulement retardée par l'artifice de son remplacement, pendant quelque temps, par une propriété plus abstraite.

Une Taxinomie des Propriétés

Il est bon, aux fins d'analyse ultérieure, que les propriétés choisies soient cataloguées dans le texte de référence avec une mention de leur appartenance à certaines catégories pré-définies. Cette taxinomie doit permettre au rédacteur de la SG de penser à adopter plusieurs points de vue complémentaires possibles au cours de son travail. Elle doit évidemment se concrétiser dans le système de repérage des propriétés par des conventions particulières.

Par exemple, on peut envisager de considérer les propriétés *fonctionnelles*, qui précisent ce que le système doit faire en général (sans, bien sûr, dire comment). Ce sont des propriétés positives. Parmi ces propriétés , on peut définir celles qui relèvent de la statique du système, appelons-les des propriétés *permanentes*, ou bien au contraire celles qui relèvent de sa dynamique, appelons-les des propriétés *de transition*. On peut aussi distinguer les propriétés *de sécurité*, qui précisent ce que le système ne doit pas faire, ses limites. Ce sont des propriétés négatives. On peut considérer les propriétés dites *de disponibilité* qui définissent l'ensemble des contraintes (temporelles ou de service) à l'intérieur desquels le système doit se mouvoir pour pouvoir être considéré comme disponible à ses "usagers". Il y a aussi les propriétés *de fonctionnement dégradé*, qui définissent les comportements particuliers du système lorsqu'il est en présence de situations spéciales. À l'intérieur de ces trois dernières catégories de propriétés (de sécurité, de disponibilité, et de fonctionnement dégradé), on peut envisager, comme précédemment, de bien séparer les propriétés permanentes de celles qui sont transitoires. On peut enfin être amené à définir des propriétés *ergonomiques*, qui précisent certaines règles d'utilisation du futur système.

La liste des catégories de propriétés que nous venons de présenter n'est pas exhaustive, bien entendu. Notre opinion est qu'il est bon, cependant, *d'en préciser le contenu* avant de commencer la rédaction de la SG. Cette liste comprendra probablement une partie des propriétés que nous avons proposées ci-dessus, mais elle pourra aussi en contenir d'autres ; en effet, chaque système particulier peut faire appel à des classes de propriétés qui ne relèvent que de lui seul.

Comment Imbriquer des Deux Textes

Précisons de nouveau que les textes de référence et explicatifs sont essentiellement *imbriqués* à l'intérieur de la SG. Pour autant, c'est plutôt, à notre avis, le texte explicatif qui doit prévaloir, c'est-à-dire constituer la trame immédiatement visible, le fil rouge, de la SG. Nous estimons en effet qu'il est important que la rédaction suive d'abord un ordre "naturel" qui en facilite la compréhension.

Nous ne prônons donc pas un exposé didactique qui serait structuré par les propriétés. Cette structuration viendra, bien sûr, mais seulement plus tard lors de la phase d'analyse de la SG en vue de sa formalisation. Par contre, il nous semble important qu'a propos de tel développement explicatif, on mette à ce moment-là en exergue telle propriété dans un texte de référence. Le rédacteur de la SG doit donc souvent penser à saupoudrer le texte explicatif de courts textes de référence, qui rendent plus précis et, en quelque sorte, définitif ce qu'il est en train d'exposer.

Rappelons encore une fois que l'ensemble des textes de référence va constituer le point de départ de la formalisation à venir. Afin que cet ensemble soit exploitable, il est donc nécéssaire que ces textes soient déjà découpés en *petites unités autonomes* bien référencées. Pour cela, les énoncés des propriétés que nous avons envisagées ci-dessus seront rédigées en autant de petits textes de référence indépendants. Ces textes devront être très soigneusement écrits de façon à ce qu'il n'en résulte aucune ambiguité résiduelle identifiée.

Résumé

Nous avons déterminé que le texte de référence était constitué à partir de l'énoncé des propriétés du sytème et que ces propriétés étaient disposées "en vrac" au sein du texte explicatif sans attention particulière à leur importance ou à leur niveau d'abstraction. Nous avons mentionné le problème de la sur-spécification et lui avons apporté une ébauche de solution sous la forme de l'élaboration de propriétés plus abstraites. Nous avons aussi déterminé une certaine taxinomie des propriétés que l'on peut voir décrites:

1. propriétés fonctionnelles,
2. propriétés de sécurité,
3. propriétés permanentes,
4. propriétés de transition,
5. propriétés de disponibilité,
6. propriétés de fonctionnement dégradé,
7. propriétés d'ergonomie.

Les Différentes Formes du Texte de Référence

Voyons maintenant quelles sont les différentes formes que peut prendre l'énoncé de ces propriétés dans le texte de référence.

Des Textes Rédigés en Français

Les éléments du texte de référence sont principalement rédigés *en français*. Nous présenterons cependant plus bas d'autres formes possibles d'écriture de ces "textes". Ces parties écrites en français devront être courtes, concises et relativement autonomes.

Des Tableaux ou des Diagrammes d'Objets

Au fur et à mesure de leur apparition, les propriétés que nous venons de mentionner ci-dessus doivent la plupart du temps être précédées de la description de certains "objets" qui en sont la cible. Sans préjuger aucunement d'une quelconque organisation future des dits objets (ni même savoir s'il deviendront un jour des "objets" au sens informatique), il est nécéssaire d'en donner cependant une certaine description dans la SG sous forme de textes de référence, ne serait-ce que par soucis de complétude. On rappelle en effet avoir admis que l'ensemble du texte de référence devait être autonome.

Lorsque les objets en question sont complexes, il est possible que leurs descriptions en français conduisent à des textes relativement lourds. De façon à alléger la présentation de tels objets, on n'hésitera donc pas à utiliser, en plus du français, toute autre forme adéquate. On peut par exemple présenter un objet au moyen d'une table, dont chaque ligne contient une description en français d'un de ses attributs caractéristiques.

C'est une forme possible, il y en a manifestement d'autres: on peut par exemple penser aussi à utiliser une forme graphique. Mais ceci doit être fait uniquement s'il existe une convention reconnue, capable de donner un sens relativement précis et clair au graphisme en question (ce n'est pas toujours le cas). En tout état de cause, cette représentation constitue un "texte" de référence à part entière, qui doit donc être dûment référencé comme tel.

Des Automates de Transition

Les descriptions de données que nous venons de mentionner ci-dessus correspondent à ce que nous avions appelé plus haut des "propriétés permanentes": elles

décrivent en effet les objets en question quelles que soient leurs évolutions possibles. On peut aussi être amené à utiliser l'autre forme de propriétés que nous avions envisagée, à savoir celle des "propriétés de transition". Par exemple, on pourrait vouloir exprimer que l'évolution de certains objets s'effectue toujours en accord avec certaines transitions bien définies.

Là encore, une description en français pourrait s'avérer parfois trop lourde. On n'hésitera donc pas à utiliser une forme plus adéquate, par exemple celle d'un diagramme de transition ou toute autre forme également pertinente. Comme plus haut, de telles représentations particulières constituent des "textes de référence" à part entière, qui doivent donc aussi être dûment répertoriés comme tels.

Des Formules Mathématiques et des Tableaux d'Unités

On peut enfin trouver des propriétés qui s'expriment tout naturellement par des formules mathématiques, que l'on n'hésitera donc pas à utiliser si besoin est. On n'oubliera pas non plus qu'un système ayant pour mission de réguler un certain processus manipule des informations représentant des grandeurs physiques. Dans ce cas, un ensemble de propriétés importantes à définir concerne le système d'unités que l'on pense devoir employer pour mesurer ces grandeurs, de même que la précision requise par leurs mesures. Voilà encore des propriétés qui s'expriment certainement très bien sous forme de tableaux. Formules mathématiques ou tableaux d'unités constituent donc, là encore, autant de "textes de référence". Il est clair qu'il y en a certainement d'autres, revêtant encore d'autres formes.

En Guise de Conclusion

Bref, comme on le voit, il ne saurait y avoir à ce niveau une forme universelle qui permettrait de procéder facilement à toutes ces descriptions. Il n'existe pas non plus de place ici pour un langage conventionnel que l'on pourrait utiliser pour écrire ces "textes" de référence. La règle fondamentale à suivre consiste donc plutôt à utiliser un certain *style* qui doit être simple et rigoureux et qui doit relever de la pratique technique courante: pas de jargon donc, ni de formalisme abscons. Tout individu doté d'une certaine connaissance du métier doit pouvoir *lire et comprendre sans difficulté* la SG.

Résumé

Nous avons identifié les différentes formes qui pouvaient convenir pour procéder à l'énoncé des propriétés qui constituent le texte de référence de la SG. Ces formes sont les suivantes (parmi d'autres):

1. un texte court en français,
2. un tableau de description de données,
3. un graphisme de description de données,
4. un diagramme de transition,
5. une formule mathématique,
6. un tableau d'unités physiques.

Analyse de la Spécification Générale et Construction de la Spécification Technique

Il nous reste maintenant à envisager quelle sorte d'analyse nous pouvons entreprendre sur la SG avant d'entamer la suite, c'est-à-dire la construction de la Spécification Technique Formelle.

La Spécification Générale de Référence

Lorsque la rédaction de la SG est terminée et qu'elle a reçu l'aval du client, on procède à son analyse en vue de la formalisation qui va suivre. Pour cela, comme nous l'avons déjà dit plus haut, on sépare le texte de référence du texte explicatif, qui n'a désormais plus d'usage. C'est, en effet, maintenant le texte de référence qui doit permettre *à lui seul* de continuer le travail. Appelons ce texte, extrait de la SG, la Spécification Générale de Référence (SGR).

La Pré-formalisation

Par le fait même de sa génèse (extraction aveugle à partir de la SG), la SGR se présente juste sous la forme d'une collection de propriétés, qui est "plate" et sans structure. Nous nous proposons d'effectuer maintenant une analyse dont le but est précisément de lui en donner une. Cette analyse s'effectue à l'aide d'une *pré-formalisation*. Pour cela, nous allons considérer individuellement les différents fragments qui constituent la SGR et effectuer sur chacun d'eux un certain travail de mise en forme. Rappelons que ces fragments sont relativement autonomes les uns par rapport aux autres et qu'ils correspondent chacun à une certaine propriété du système.

Le Document des Propriétés Formalisées

Pour les besoins de cette analyse, on va donc construire un nouveau document, que l'on appelle le Document des Propriétés Formalisées (DPF). Chacune des propriétés, dûment répertoriée, que l'on trouve dans la SGR, se retrouve, avec le même repérage, dans le DPF. Celui-ci est donc une sorte de catalogue, dont chaque entrée correspond à une certaine propriété identifiée par le même repérage que celle qu'elle avait dans la SGR (et donc dans la SG)[4].

On trouve ensuite une certaine *représentation formelle* de chaque propriété ainsi référencée. Il s'agit d'un petit modèle qui transcrit sous une forme mathématique la propriété en question. Pour ce faire, on peut être amené à effectuer dans le DPF une formalisation *préalable* des définitions et des propriétés qui correspondent aux "objets" rentrant en jeu dans ce modèle. Un certain ordre de préséance entre les différentes entrées du DPF commence donc à se mettre en place à cette occasion.

Introduction de Nouvelles Propriétés plus Abstraites

Le DPF peut aussi contenir des propriétés qui ne se trouvaient pas explicitement présentes ni répertoriées dans le SGR. Nous y avons déjà fait allusion plus haut à propos de la question de la sur-spécification. Ces propriétés ne sont pas vraiment nouvelles ; elles ne correspondent donc pas à une fonctionalité ou à une exigence qui n'existait pas déjà dans la SGR, et que l'on inventerait subitement. Elles sont seulement *plus abstraites* que certaines des propriétés de la SGR. Elles doivent donc être introduites en tant que telles dans le DPF et, bien entendu, contenir une référence aux propriétés qu'elles sont censées abstraire (ce qui restera à prouver en toute rigueur, bien entendu). Par mesure de cohérence, elle doivent aussi être introduites sous la forme la plus adéquate dans la SGR, comme si elles s'y trouvaient initialement.

À noter que le formalisateur doit, à ce niveau, faire preuve d'une certaine créativité pour exhiber ces nouvelles propriétés plus abstraites que d'autres. Ces propriétés sont importantes, car elles permettent, comme nous l'avons déjà mentionné plus haut, de constituer petit à petit une *hiérarchie d'abstraction* entre les propriétés. Elles serviront aussi à construire la Spécification Technique Formelle de façon très rigoureuse, par raffinements successifs.

[4]De cette façon, grâce à un outil simple, on peut établir un lien immédiat entre DPF, SGR et même SG.

Formalisation des Propriétés de Transition

Les propriétés dynamiques, dites "de transition", auxquelles nous avons fait allusion plus haut se formalisent de deux manières différentes: soit à l'aide de prédicats munis de conventions spéciales, soit à l'aide "d'opérations" explicitement définies. On pourra, à cette occasion, être amené à définir de telles opérations spontanément au fur et à mesure que les besoins s'en feront sentir. Bien entendu, il faudra alors reporter ces nouvelles propriétés dynamiques dans la SGR.

Rétro-action sur la Spécification Générale de Référence

Il est très possible que le processus de pré-formalisation, que nous venons de décrire, entraine, de par sa rigueur même, une certaine remise en cause de la SGR. En effet, malgré toute l'attention apportée à leur rédaction, les propriétés décrites dans la SGR peuvent se trouver sous une forme qui est encore incomplète, ambigue, voire contradictoire. Le but de la pré-formalisation est précisément de lever, dans la mesure du possible, toutes ces incohérences (nous verrons plus bas que la preuve va aussi contribuer à en éliminer encore). La difficulté même de traduire une propriété en un modèle mathématique simple peut aussi être l'indice révélateur d'un problème.

Lorsque ceci se présente au cours de la pré-formalisation, il ne faut pas hésiter à reprendre la SGR, soit en reformulant complétement certaines propriétés, soit même en en introduisant d'autres (comme nous l'avons déjà envisagé plus haut). Bien sûr, ces corrections doivent être dûment répertoriés de façon à ce qu'il n'y ait pas de dérives dangereuses entre la SGR et le CDC.

Validation des Propriétés Formalisées

Lorsque *toutes* les propriétés de la SGR ont ainsi été transcrites dans le DPF, accompagnées éventuellement de quelques autres comme nous l'avons signalé, la SGR devient désormais inutile. C'est maintenant le DPF qui fait foi. À ce sujet, il peut être intéressant de "valider" le DPF par rapport à la SGR, et ce de façon à pouvoir montrer au client qu'il n'y a pas eu de dérives entre les deux. La question est délicate, bien entendu, puisqu'il s'agit de comparer des textes écrits "en français" à d'autres écrits "en mathématiques", "langue" que le client ne maitrise pas forcément.

Pour cela, celui-ci peut faire appel à un expert, indépendant du réalisateur, capable de comparer chaque propriété de la SGR à sa contrepartie formelle du DPF. Une telle validation a ses limites évidemment, puisqu'elles s'effectue entre deux textes situés à des niveaux linguistiques très différents, mais elle n'est tout

de même pas trop difficile car l'on compare uniquement de *petits fragments* entre eux. C'est ici que la *finesse de l'échantillonage* des propriétés du système dans les textes de référence prend toute son importance.

Hiérarchisation des Propriétés Formalisées

À ce point du travail, la construction de la spécification ne peut pas cependant être encore mise en chantier. Nous nous trouvons en effet en présence d'un grand nombre de propriétés qu'il nous faut maintenant *hiérarchiser* afin de pouvoir les incorporer petit à petit dans le document suivant qui est donc la Spécification Technique Formelle (STF).

Il sort du cadre de cette étude d'expliquer comment l'on va pouvoir effectuer cette hiérarchisation dans la pratique. Cette question a été abordée et résolue de façon intéressante dans [4]. Disons simplement que la recherche de cette structure va se faire en analysant essentiellement la façon dont les propriétés sont reliées entre elles au moyen des "objets" auxquels nous avons fait allusion plus haut et aussi en déterminant les relations d'abstraction qu'elles peuvent entretenir entre elles. À la fin de ce tavail, on obtient une certaine relation d'ordre partiel entre les propriétés, relation qui indique la façon dont on va pouvoir construire la STF par raffinements successifs.

Construction de la Spécification Technique

On va maintenant exploiter la hiérarchie que nous venons d'élaborer pour construire la Spécification Technique Formelle. Grosso modo, on commence par incorporer dans la STF les propriétés qui ne dépendent d'aucunes autres, ce seront a priori les plus "abstraites" ; elles sont introduites en même temps que les objets qui les concernent. La STF est ensuite *raffinée*, en incorporant les propriétés qui suivent dans l'ordre de préséance envisagé plus haut, etc. On pourra aussi être amené à *décomposer* la STF en plusieurs morceaux relativement indépendants. La STF sera considérée comme achevée lorsque *toutes* les propriétés du DPF auront ainsi été incorporées.

À noter que les propriétés passent du DPF à la STF *sans distorsion* d'aucune sorte: le texte formel de chaque propriété est simplement recopié de l'un dans l'autre. Cest pourquoi le problème de la validation de la STF par rapport au DPF ne se pose pas dans les mêmes termes que dans les cas précédents. Il s'agit ici d'un simple test de nature mécanique vérifiant que les copies ont bien été effectuées.

Preuves de la Spécification Technique Formelle

Pendant la construction de la STF, on est amené à faire un certain nombre de preuves. Elles ont d'abord pour but de garantir que les propriétés dynamiques que l'on a introduites sont bien compatibles avec les propriétés statiques. Elles ont aussi pour fonction de montrer que chaque niveau de raffinement de la ST n'est pas contradictoire avec les niveaux plus abstraits qui le précède.

Au cours de ces sessions de preuves, de nouvelles difficultés peuvent surgir. Soit parce que ce que l'on doit prouver *ne peut pas l'être*, soit parce que la preuve *semble trop compliquée*, voire infaisable, avec les outils dont on dispose. Nous sommes là au coeur du problème: la tentative de preuve avortée est en train de nous "dire" que le problème doit probablement être posé d'une façon différente. C'est l'indice d'une erreur qui nous est ainsi signalée, et qu'il faut donc corriger dans *tous* les documents concernés: le CDC, la SGR et le DPF. Comme on le voit donc, la preuve peut nous amener à de nouveau rétro-agir sur les phases situées plus en amont.

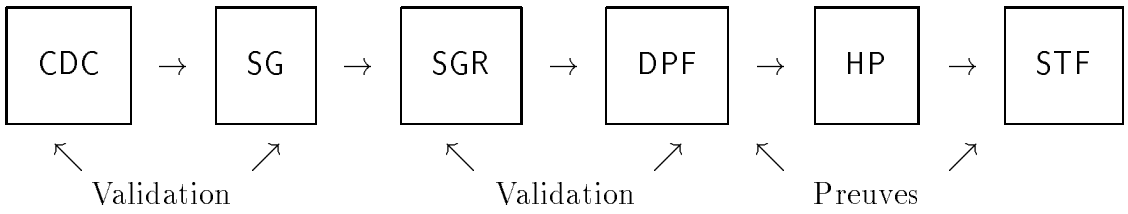
Résumé

Nous avons expliqué comment les propriétés composant le SGR (extrait des textes de référence de la SG) étaient

1. transcrites en petits modèles mathématiques relativement indépendants,
2. hiérarchisées les unes par rapport aux autres dans un rapport d'abstraction,
3. incorporées petit à petit dans la Spécification Technique Formelle construite par raffinements successifs et décomposition.
4. prouvées quant à leur cohérence mutuelle.

Bilan

Comme on le voit sur le schéma ci-dessous, quatre phases intermédiaires se sont glissées entre le CDC et la STF. À noter, comme nous l'avons déjà signalé, que chacune de ces phases peut être amenée à *rétro-agir* sur celles qui la précède. Ceci montre que la vérification et la validation du Cahier des Charges, notre but initial, sont en réalité réalisées au moyen d'un *processus complexe* qui nécéssite un effort important. Il n'y a pas de méthode-miracle, on s'en serait douté. Il n'y a qu'un patient travail, qui nous amène, à l'aide d'un cadre relativement précis, à *beaucoup réfléchir* au futur système que nous voulons construire.



Cahier des Charges	CDC
Spécification Générale	SG
Spécification Générale de Référence	SGR
Document des Propriétés Formalisées	DPF
Hiérarchisation des Propriétés	HP
Spécification Technique Formelle	STF

References

[1] J.-R. Abrial. *The B-Book: Assigning Programs to Meanings*. Cambridge University Press (1996)

[2] J.-R. Abrial. *Exemple de Rédaction d'un Cahier des Charges*. Non publié (1998)

[3] D. Bert (Ed). *B'98: Recent Advances in the Development and Use of the B Method*. LNCS 1393 Springer (1998)

[4] N. Lopez. *Construction de la Spécification Formelle d'un Système Complexe*. Mémoire d'ingénieur CNAM (1996)

[5] I. Sommerville *Requirements Engineering. A Good Practice Guide*. Wiley (1997)

[6] Steria. *Atelier B Version 3.3*. (1997)