

Un essai comparatif de deux méthodes de conception des systèmes d'information (Merise et OMT) : acquisition et utilisation

Jacques RUIZ

Enseignant chercheur, CREGO, IAE, ISIM, Montpellier II

RESUME

Cet article se propose d'apporter quelques résultats d'expérimentation en comparant au plan de l'apprentissage que de la mise en œuvre, une des méthodes de conception orientée objet, OMT, avec une des méthodes systémiques parmi les plus représentatives, MERISE. Les résultats de cette recherche tendent à nuancer le discours habituel consistant, à partir de l'intérêt des concepts utilisés dans les méthodes objet, à extrapoler sur leurs éventuels gains de productivité dans le processus de conception-développement.

Mots-clés : Méthodes de conception des systèmes d'information, méthodes d'analyse orientée objet, efficacité du développement des applications, maintenance des programmes.

ABSTRACT

The purpose of this research is to present some results on the comparison between two information system analysis methods (MERISE and OMT) concerning their learning and operating aspect. The results of this study show that oriented-object analysis methods have not always and automatically better efficiency than other methods about learning and use in a real project. On these points these results are in opposition with what is currently written. A discussion has to be opened on the necessary conditions for the effective use of these methods in the firms.

Key words: Information system analysis methods, object-oriented analysis methods, programs efficiency, programs maintenance.

Les méthodes de conception des systèmes d'information évoluent rapidement depuis leur apparition et leur utilisation dans les organisations à partir des années 70. Après une absence totale de principe méthodologique, liée à la nouveauté de la technologie informatique, on est passé à des méthodes souvent partielles, principalement axées sur des démarches (découpage en étapes, sous-étapes...), assez pauvres au plan des concepts : MINOS, CORIG en Europe, SSA aux Etats-Unis sont des exemples typiques de cette première génération de méthodes. Dans les années 80 sont apparues des méthodes plus globales, privilégiant les niveaux conceptuels dans la représentation de l'organisation et de son système d'information : le courant structuré (SADT, HIPO, DEMARCO...) et le courant systématique (MERISE, AXIAL, REMORA...) représentent ces méthodes de deuxième génération. L'apparition de méthodes utilisant les principaux concepts de ces deux courants (JACKSON, IE, le rapprochement entre SSADM et MERISE2) montre la tendance à la normalisation et à la convergence de ces familles de méthodologies.

Aujourd'hui apparaît une troisième génération de méthodes, "orientée objet", voulant intégrer les concepts de la programmation objet (héritage, composition, polymorphisme...) au monde de l'analyse et de la conception. Depuis les travaux de G. Booch (Booch,1986) des méthodes de conception et d'analyse apparaissent : OOA (Coad et Yourdon, 1990), OMT (Rumbaugh,1991), etc. Encore peu utilisées dans les entreprises, elles sont élaborées dans un souci de **réutilisation** d'un certain nombre de résultats de conception antérieurs, dans une même entreprise ou dans des entreprises différentes travaillant sur des **motifs** de spécification identiques. Ces méthodes, très prometteuses au plan des concepts, paraissent très

séduisantes a priori quant à leur impact potentiel sur la productivité de la conception ou sur la maintenance des applications. Qu'en est-il dans la réalité ? Vont-elles révolutionner le monde de la conception ? Ne vont-elles pas au contraire complexifier la démarche de conception pour un résultat peu en rapport avec ceux escomptés ?

Les expériences du passé, lors de l'implantation des méthodologies de deuxième génération (MERISE en France par exemple) commencent à être analysées voire, décriées. Certains (Ruiz,1993) se demandent si ces méthodologies si prometteuses elles aussi en leur temps, sont réellement adaptées à toutes les structures organisationnelles. Certains autres (Weigiao,1992) vont jusqu'à remettre en cause leur intérêt.

Cet article se propose, à la lumière des expériences du passé, d'apporter quelques résultats d'expérimentation sur la pratique des méthodes de conception objet par rapport aux méthodes en vigueur actuellement dans les entreprises. Ces résultats n'ont pour seule ambition que de soulever un certain nombre d'interrogations et de susciter quelques remises en question sur l'idéologie dominante en la matière, fondée plus sur la prédominance des concepts utilisés que sur la praticabilité de la démarche.

1. UNE RECHERCHE DIFFICILE À CONDUIRE

La recherche que nous présentons a pour but de comparer l'utilisation de deux méthodologies différentes - d'une part MERISE (Tardieu et al.,1983), méthode la plus utilisée dans les grandes organisations en France, d'autre part OMT représentant l'une des méthodes orientées objet parmi les plus citées à l'heure actuelle dans la littérature.

Une comparaison véritable s'avère souvent impossible dans la réalité, tant les conditions à respecter pour que cette comparaison soit scientifiquement satisfaisante, sont difficiles à obtenir. En effet, pour évaluer réellement les apports de tel outil par rapport à tel autre il faut pouvoir appliquer ces outils sur le même ouvrage ce qui, en matière de conception de systèmes d'information, s'avère difficilement réalisable pour les raisons suivantes :

- Les tâches de conception d'un projet d'informatisation sont souvent trop lourdes et trop coûteuses pour qu'on puisse les réitérer avec des outils différents dans le seul but de comparer ces outils.

Dans le monde de l'entreprise, les sommes allouées à la recherche-développement en termes de méthodologie de conception sont plus axées sur la formation à une méthode, sa diffusion, son support que sur son choix. Celle-ci s'effectue souvent sur des critères subjectifs (la mode, la concurrence, la connaissance du responsable méthodes en la matière (Ruiz, 1991). Sur cinq grandes entreprises contactées pour ce type d'expérience, entreprises caractérisées par leur effort important (plus de 4 % du CA) dans le développement de leur système d'information, aucune n'a répondu favorablement à notre demande : toutes ont reconnu l'intérêt de la recherche mais ont refusé d'y participer, à cause du coût, tant direct qu'indirect, qu'une telle expérience représentait.

- Les interrelations entre la méthode utilisée et le concepteur sont très difficiles à gérer. Les résultats obtenus dans une même conception avec des méthodes différentes sont trop fortement corrélés aux phénomènes d'apprentissage.

Pour que les résultats d'une comparaison entre l'utilisation de deux méthodes de conception différentes aient réellement un sens, ces deux

méthodes doivent être appliquées dans un même environnement. Celui-ci est fortement dépendant du système à concevoir (complexité, difficulté...) et de l'utilisateur de la méthode (compétence, expérience...). Il faudrait pouvoir faire développer le même sujet par le même acteur pour respecter cette contrainte d'environnement et s'approcher du "toute chose étant égale par ailleurs" rituel. On comprend aisément que cette contrainte est impossible à respecter dans la réalité car l'acteur ou le groupe d'acteurs, même s'il s'agit physiquement du même, va obligatoirement évoluer dans le temps (apprentissage du système à concevoir) au cours de l'expérience.

En outre, tout concepteur a été formé à une méthode même si ses connaissances sur d'autres méthodes peuvent être variées. Une meilleure maîtrise de cette méthode la lui fera utiliser plus efficacement qu'une autre. Pour comparer l'efficacité de deux méthodes, comment déterminer alors quelle est la part de la méthode elle-même et celle du concepteur qui utilise plus ou moins bien celle-ci ?

Ces arguments rendent toute recherche comparative sur les méthodes de conception des systèmes d'information très délicate à mener sur le terrain de leur utilisation "normale", c'est-à-dire dans les organisations. Ils nous ont conduit à bâtir un protocole de recherche expérimentale proche de ceux utilisés en laboratoire, où les paramètres d'environnement peuvent être, sinon totalement contrôlés, du moins minimisés quant à leur influence sur les résultats obtenus.

2. UNE METHODOLOGIE DE CIRCONSTANCE

2.1 Les prérequis

L'analyse des difficultés inhérentes à une recherche de ce type conduit à

s'interroger sur une hiérarchisation des problèmes à résoudre pour contourner ces obstacles méthodologiques et proposer des approximations satisfaisantes. Ainsi du protocole "faire faire la **même conception** (sur le même problème d'informatisation) par le **même acteur** utilisant tour à tour deux méthodologies de conception différentes qu'il connaît identiquement et évaluer dans les deux cas les résultats de la conception" nous sommes passés au protocole "faire faire la **même conception** au même moment par deux acteurs différents, chacun utilisant une **méthodologie de conception différente** et évaluer les deux résultats ainsi obtenus".

Pour que cette approximation permettant de lever le problème de l'apprentissage de l'acteur dans le temps soit admissible, il faut deux acteurs assez proches en termes de compétence et de formation. Cette condition passe par :

- une évaluation des compétences *a priori* ;
- une formation des individus, de même niveau, jusqu'à une maîtrise à peu près équivalente des deux méthodes de conception que l'on veut comparer.

a) Protocole d'évaluation des compétences *a priori*

Dans deux formations différentes, une école d'ingénieurs en informatique de gestion, un mastère en conception des S.I., nous avons sélectionné deux groupes d'étudiants présentant des caractéristiques homogènes. Pour ces quatre groupes de trois étudiants chacun nous avons choisi des candidats sur :

- Leurs capacités intellectuelles

Dans la première école recrutant à bac+2, les étudiants ont tous obtenu un DEUG de mathématiques avec la mention AB. Dans la deuxième école,

les candidats sont tous ingénieurs de niveau équivalent. Les candidats ont, de plus, subi un test de sélection (test d'aptitude IBM à l'embauche des cadres en informatique) portant sur la logique, le raisonnement, la rigueur, l'analyse et la synthèse. Tous les étudiants retenus ont obtenu une note comprise entre 13 et 15 à cette épreuve.

- Leur non formation aux méthodologies de conception :

Nous nous sommes assurés que tous les candidats sélectionnés n'avaient en informatique que des connaissances limitées à la programmation de base (BASIC, PASCAL, ...). Ce critère de sélection avait pour objectif, dans notre expérience, de maîtriser la formation des acteurs sur les méthodes que nous voulions comparer.

- Leur motivation à l'analyse :

Un test de profil de carrière (utilisé dans une grande SSCI française) a été administré montrant une ambition forte ou très forte des candidats sélectionnés pour les activités de conception et de conduite de projet.

- leur capacité de modélisation :

Au cours de leur cursus les étudiants ont été testés sur leur capacité à modéliser des problèmes mathématiques. Une homogénéité dans les résultats obtenus lors de ces évaluations a été recherchée.

b) Protocole de formation à la méthodologie de conception

La formation à la méthodologie a été effectuée de manière identique et par le même enseignant (le chercheur lui-même) dans chacune des écoles et pour chacun des groupes (nous appellerons OMT1 et MERISE1 les groupes ayant suivi respectivement les formations OMT et MERISE dans la première école, OMT2 et MERISE2 les groupes de la deuxième école). Cette formation s'est déroulée de la manière suivante :

- un cours magistral (20 h) sur les concepts principaux c'est-à-dire les modèles de représentation proposés par la méthode enseignée. Pour MERISE il s'agissait de présenter et d'expliquer les modèles conceptuels des données (modèle entité-association) et des traitements, les modèles logique et organisationnel des données et des traitements puis les modèles physiques. On insistait en outre sur leur place et leur rôle dans la démarche en fonction de leur niveau d'invariant. Pour OMT, ce cours consistait à présenter d'abord les concepts utilisés - classe, héritage, composition, polymorphisme ... - puis les modèles objet, dynamique et fonctionnel en insistant sur les interactions entre ces modélisations et sur leur complémentarité ;
- un cours magistral (20 h) sur la démarche de conception : ce cours dispensé après le précédent permettait d'exposer les différentes phases et étapes de la méthode tant au niveau de l'analyse du problème que de la conception de la solution. Il s'avère que les deux démarches sont assez identiques en dehors des concepts manipulés. Ce cours s'accompagnait de la présentation d'un exemple et de sa résolution. Le même exemple était présenté dans les deux formations : il s'agissait de l'informatisation de la gestion d'une quincaillerie en gros (Hertz et al. 1988) ;
- une étude de cas (40 h), tirée d'une étude réelle (gestion des approvisionnements chez MOULINEX), simplifiée pour être conciliable avec les contraintes de temps de formation, à conduire en groupe jusqu'à la conception détaillée de la solution ;
- une évaluation, dans un temps limité, sur les acquisitions de connaissances, portant essentiellement sur la modélisation.

2.2 L'expérimentation

Les conditions de l'expérience une fois atteintes, l'expérience elle-même, s'est déroulée en même temps sur les deux échantillons de la façon suivante :

- Sélection d'un cas réel répondant aux critères suivants :
 - bonne connaissance du chercheur sur le sujet proposé ;
 - aucune connaissance du secteur d'activité concerné et du projet à traiter par les participants à l'expérience ;
 - mission permettant, à l'exception des phases de planification, de recouvrir la totalité du cycle de vie d'un projet (analyse préalable \Rightarrow mise en place) et d'utiliser les méthodes dans leur globalité.

Le projet retenu concernait le suivi des interventions sur les câbles téléphoniques pressurisés du centre de construction des lignes de France Telecom. Certains câbles téléphoniques enterrés, contenant plusieurs centaines de lignes, sont pressurisés de façon à prévenir les infiltrations d'eau. Si des réparations doivent être entreprises sur ces câbles, le centre confie les travaux à des sous-traitants agréés, gère leur intervention, dépressurise puis repressurise les câbles avant et après intervention, avertit les clients d'éventuelles coupures, teste les pressions totales ou par secteur sur les câbles après intervention pour déterminer si les travaux n'en ont pas affecté l'imperméabilité. L'ensemble de la procédure entre le centre, l'équipe de coupure, l'agent de constatation, les sous-traitants, les clients, devait être analysée et améliorée (vitesse, délais, erreurs...).

- Exposé du sujet, en même temps, à tous les groupes de projet et précisions apportées en fonction des ques-

tions posées. Un délai maximum de quatre mois (charge de 12 mois/homme) a été imposé pour la réalisation complète de ce projet à chacun des quatre groupes. Les conditions de conduite de projet ont été établies comme suit :

- un seul interlocuteur, en l'occurrence le chercheur, pour tous les groupes ;
- aucune communication entre les groupes pendant toute la durée de vie du projet ;
- un compte-rendu d'activité hebdomadaire écrit précisant la nature des activités effectuées et le temps passé par activité et par acteur ;
- un exposé bi-hebdomadaire de chaque groupe sur l'avancement du projet ;
- autant de réunions de travail entre le groupe de projet et le chercheur que nécessaires, à la demande du groupe, pour la compréhension du sujet ;
- une présentation générale des résultats atteints en fin de projet.

• La conduite de projet a été laissée à l'initiative de chaque groupe avec, comme nous l'avons déjà dit, un contrôle du chercheur sur le rythme de travail. Il est à noter que cette activité donnait lieu à une évaluation très importante pour le cursus de chacun des participants, gage de sérieux et de motivation de leur part. La conception et la réalisation devaient s'effectuer sans l'aide d'ateliers de génie logiciel dont les différences auraient pu influencer sur la productivité des études.

Nous avons enregistré toutes les réunions avec chaque groupe ce qui a constitué, avec les comptes-rendus d'activité et les états d'avancement, les principales données recueillies au cours de cette expérience. Les résultats que nous présentons ci-après résultent de l'analyse de ces données brutes et des observations directes effectuées tout au long de l'expérience.

3. LES RESULTATS DE L'EXPERIENCE

3.1 Un constat sur la formation

Les premiers résultats émergent avant même que la phase de conduite de projet ne soit atteinte. C'est au cours de l'enseignement des méthodologies que nous avons pu constater que :

Les concepts utilisés dans les méthodes objet semblent plus difficiles à acquérir que ceux des méthodes différenciant données et traitements.

D'une façon générale, les groupes formés à la méthode OMT ont obtenu des résultats moins satisfaisants, en termes de modélisation que les deux autres groupes, après la phase de formation initiale. Il a même fallu pour que l'expérience puisse se poursuivre, dans des conditions acceptables d'équivalence d'expertise entre les groupes, accroître la formation des groupes OMT. L'analyse détaillée des difficultés relevées lors de l'acquisition des concepts, pour chacune des méthodes, montre que :

• Pour MERISE, les concepts d'entité et d'association dans le modèle de données ont été rapidement acquis, après quelques représentations malheureuses de supports d'information (bons de livraison, commandes...) en tant qu'entités. Le concept de cardinalité des associations ne pose aucun problème tant que les associations sont binaires (entre deux entités). Pour des associations n-aires (entre plusieurs entités) le concept de cardinalité est plus difficile à appréhender. L'analogie avec les notions de fichier, de lien entre fichiers et d'enregistrements, notions assez pragmatiques et tangibles dans l'esprit des étudiants, permet de surmonter assez aisément cette difficulté.

Concernant les modèles de traitement, la notion d'événement est la plus délicate à comprendre. Ce concept que l'on retrouve d'ailleurs dans OMT et qui pose le même problème, semble recouvrir, dans l'esprit des étudiants, la notion d'opération que cet événement déclenche. La prise en compte d'événements temporels (fin de mois, passage à une date d'échéance...) et d'événements indépendants du processus de gestion étudié (détérioration d'un stock...) permet de lever ces ambiguïtés. Pour les autres concepts utilisés dans la représentation des traitements - opérations, résultats, synchronisation, poste... - leur acquisition pose peu de problèmes.

- Pour OMT, les concepts d'objet et de classe avec le regroupement dans la définition de l'objet ou de la classe de ses caractéristiques informationnelles (attributs) et instrumentales (services) ne présentent, au départ, aucune difficulté d'acceptation. Ces concepts sont, de l'avis des étudiants, assez comparables à ceux de "collection d'information" et de traitements effectués sur des "objets de gestion". Le principe d'encapsulation, d'interface publique ou privée, apparaît par contre comme extrêmement théorique, difficilement assimilable quant à son utilisation. La traduction physique de cette notion en programmation objet semble être le passage obligé pour la compréhension de ces concepts en analyse.

Les notions de spécialisation-généralisation et de composé-composants dans le concept plus large d'héritage (liens PART-OF, MEMBER-OF, IS-A) soulèvent peu d'interrogations, *a priori*, quand on les présente. Ils s'avèrent pourtant difficiles à mettre en œuvre ensuite dans les études de cas : leur mauvaise assimilation génère des ambiguïtés, des décompositions arbitraires et (ou) trop détaillées par rap-

port à la vision de l'utilisateur. Dans les études menées, les étudiants parvenaient souvent à des modèles de représentation beaucoup trop parcelaires (donc compliqués). Un effort d'agrégation, par un processus inverse de généralisation, devait alors être effectué (perte de temps) pour que la mise en œuvre des modèles de représentation soit acceptable.

Les concepts utilisés dans le modèle dynamique de OMT - événement, état, activité, action - ont soulevé de multiples interrogations et de réelles difficultés de compréhension : tout d'abord des mises en œuvre assez approximatives du concept d'événement (idem dans MERISE) ; les notions d'états, de transition d'états, d'opérations exécutées par des objets pendant un état (activité), d'opérations exécutées pendant une transition d'état (action) n'ont pas été acquises facilement. Ce modèle dynamique qui représente en fait les envois de messages entre objets (événements), les différents états pendant lesquels les objets ont une activité et les réponses de ces objets à ces demandes (polymorphisme), a dû être illustré à plusieurs reprises et sur plusieurs exemples par le formateur. C'est principalement sur ces points qu'une formation complémentaire a dû être assurée pour OMT (15 h supplémentaires environ) pour parvenir à une maîtrise équivalente des groupes vis-à-vis de leur méthode respective.

En ce qui concerne le modèle fonctionnel OMT, les principaux concepts - processus, flux de données... - ont été compris assez rapidement. Les concepts d'acteur et d'objets de stockage ont toutefois provoqué quelques demandes d'éclaircissement.

Les premiers résultats de l'expérience montrent donc qu'une formation plus longue semble nécessaire pour l'apprentissage d'une méthode d'analyse et de conception orientée objet

que pour celui d'une méthode "classique" de type MERISE. Si l'on cumule la totalité des temps consacrés à l'enseignement initial, aux explications complémentaires postérieures à celui-ci, tant lors de la phase de formation que lors des phases de mise en œuvre (étude de cas puis projet), on peut estimer l'augmentation de la durée de formation entre 40 % et 50 %.

Ce résultat n'est certes pas étonnant : les concepts utilisés en analyse orientée objet sont en effet plus rigoureux, plus précis, plus abstraits (exemple : le modèle de la dynamique de OMT est beaucoup plus représentatif du réel que le MCT de MERISE, assez pauvre avec sa représentation en processus et séquençement) et nécessitent donc un apprentissage plus long. Ce résultat corrobore les constats effectués (Ruiz, 1993) lors du passage des méthodes CORIG ou SSA aux méthodes MERISE ou SADT (Ross, 1977) dans les années 80. S'il n'est en rien étonnant, il mérite tout de même d'être signalé tant les conditions de l'applicabilité des différentes méthodes dans les organisations peuvent en être affectées, question rarement abordée dans la littérature quoique très souvent soulevée par les responsables sur le terrain.

3.2 Des résultats sur le processus de conception et de développement

Les autres résultats sont directement issus de l'expérience elle-même. Ils portent sur les phases de conception et de réalisation du projet proposé, conduites à partir de méthodologies différentes.

Les données relevées au cours des deux expériences ne sont pas comparables directement, étant donné le niveau de qualification différent des individus des deux échantillons, mais elles convergent fortement sur les différences observées dans les deux cas.

Ce sont ces différences relevées phase par phase au cours du déroulement du projet que nous présentons ici :

- Les phases d'analyse, comprenant l'élaboration des modèles de représentation ont duré de 30 % à 50 % de plus chez les groupes OMT que chez les groupes MERISE.

Des différences sont apparues dès l'élaboration des modèles objet d'une part et des modèles de données d'autre part. Elles représentent des augmentations de durée de l'ordre de 20 % dans les deux expériences. Ces écarts dans la conception se sont accrus lors de la prise en compte de la dynamique du système (modèle dynamique de OMT et MCT de MERISE) pour atteindre des facteurs de l'ordre de 60 % à 80 %. Les différences de temps passé pour l'élaboration des modèles fonctionnel (OMT) et logique (MERISE) sont assez peu significatives mais toujours en faveur de MERISE (1,05 et 1,10 pour les groupes OMT dans les deux expériences).

Un temps très important a été consacré par les groupes OMT à vérifier la cohérence des modèles entre eux. Pour les groupes MERISE, la vérification de cette cohérence (MCT-MCD, MLD-MOT, cohérence des modèles externes) a été beaucoup plus rapide. A travers la vérification de cette cohérence, les groupes OMT ont été amenés à modifier (réitération du processus d'élaboration des modèles) en moyenne trois à quatre fois leurs modèles. Les groupes MERISE ont par contre atteint un degré de cohérence satisfaisant après deux itérations.

- Les phases de conception détaillée qui consistaient principalement dans ce cas à estimer les ressources nécessaires, à optimiser les modes d'accès aux données, à implémenter les associations, à choisir le mode de stockage des données, à déterminer l'architecture du système de contrôle, ont été plus longues à réaliser par les groupes

OMT (de l'ordre de 20 % en moyenne). La principale difficulté rencontrée avec OMT s'est située dans la définition de la base de données. Seules les bases de données relationnelles étaient connues des étudiants ce qui peut expliquer en partie cela. Sur l'architecture du système de contrôle, les groupes OMT ont passé moins de temps que les groupes MERISE (10 % de moins en moyenne sur les deux expériences). On aurait pu croire, *a priori*, que ce facteur aurait été plus important grâce à l'utilisation des pré et post-conditions dans le modèle objet. Il s'est avéré que ces conditions établies en amont dans l'analyse devaient souvent être reprises et affinées, à ce niveau de la conception, en fonction de l'implémentation des associations et du regroupement des classes et des associations dans les modules.

La décomposition du processus de conception est certes arbitraire surtout pour une comparaison entre deux méthodes dont les démarches et les modèles ne se recouvrent que partiellement. Ces résultats, détaillés par point de fonction n'ont pour but que d'illustrer un certain nombre de difficultés rencontrées. Les résultats à retenir concernent les processus d'analyse et de conception dans leur globalité comme indiqué dans les tableaux 1 et 2.

- Concernant la réalisation, les groupes MERISE ont utilisé le langage de type impératif C, les groupes OMT utilisant le langage orienté objet c++. Tous les groupes ont bénéficié des mêmes outils de développement : les utilitaires liés au langage et un formateur d'écran aussi bien manipulable sous C que sous C++.

L'objectif de l'expérience ne consistait pas *a priori* à évaluer les temps et charges de réalisation, phases non directement prises en compte dans les méthodes d'analyse et de conception que nous voulions comparer. Cette phase a été conduite pour tenter

d'évaluer, d'une part la qualité de l'analyse, d'autre part l'influence des résultats des analyses sur les phases aval dans le cycle de vie du projet.

Nous pouvons considérer, au vu des résultats de la programmation et de la mise en œuvre des programmes que, tant pour les groupes OMT que MERISE, les résultats de l'analyse et de la conception ont été satisfaisants : ils n'ont pas été remis en cause, à quelques détails près, en phase de programmation.

Pour les temps de réalisation et de mise en œuvre, on peut remarquer qu'un avantage, certes minime de l'ordre de 8 % à 10 %, se dégage, en terme de productivité, au bénéfice des équipes MERISE (tableaux 1 et 2). Le nombre de lignes écrites explique peut-être ce résultat, puisque, en moyenne 12 000 lignes ont été écrites par les équipes MERISE-C, les équipes OMT-C++ en écrivant quant à elles 14 000.

Il n'apparaît donc pas, contrairement à ce que l'on aurait pu supposer *a priori*, de gains en termes de réalisation d'une spécification par rapport à une autre, les qualités de programmation pouvant être jugées comme équivalentes par rapport aux résultats obtenus (interface utilisateur, temps de réponse, sécurité).

3.3 Des indications sur la maintenance des applications

La qualité du système mis en place dépend non seulement de son adéquation aux besoins informationnels exprimés et de ses performances intrinsèques mais aussi de son évolutivité potentielle, de la facilité de sa maintenance.

C'est pour apprécier l'impact de la nature des spécifications sur ces critères, dont l'importance n'est plus à démontrer, qu'une dernière étape de

l'expérience a consisté à demander des modifications à chacun des groupes par rapport à ce qui avait été défini au préalable. Ces modifications consistaient, pour résumer, à pouvoir faire appel à plusieurs entreprises sur le même chantier, à pouvoir exécuter les travaux par intermittence, à planifier plus strictement les interventions pour que les usagers soient avertis plus tôt de ces interventions et de leurs conséquences, à pouvoir suivre plus précisément l'évolution des pressions sur les câbles.

Elles imposaient des modifications de l'analyse, de la conception, puis de la réalisation. Nous avons évalué de manière globale l'impact de ces nouvelles demandes sur les phases d'analyse-conception d'une part, sur la programmation d'autre part. Il en ressort que :

- sur la partie analyse-conception, les groupes MERISE et OMT ont réussi à prendre en compte les modifications demandées dans un temps sensiblement équivalent. Pourtant, sur les critères de complétude et d'exhaustivité, il apparaît que la qualité des spécifications modifiées est plus faible pour les groupes OMT. Ceux-ci ont dû reprendre leur dossier d'analyse afin de le finaliser correctement et complètement, ce qui a représenté en moyenne une charge de travail supplémentaire de 4 jours/homme environ ;
- sur la partie réalisation, les résultats des deux expériences sont opposés : le groupe OMT1 a mis moins de temps que le groupe MERISE1 (- 1 jour), le groupe MERISE2 étant plus rapide que le groupe OMT2 (- 4 jours). Une analyse plus fine (tableaux 3 et 4) des modifications apportées dans les applicatifs montre que :
 - pour l'expérience 1, regroupant des individus plus confirmés en programmation tant objet qu'impérative, le nombre de lignes de

code à modifier est moindre en C++ (groupe OMT) qu'en C (groupe MERISE). Cet écart peut être estimé à 12 %. La productivité des développeurs en C, estimée sur le temps nécessaire pour modifier correctement une ligne de code, est par contre plus importante que celle du groupe C++ (environ + 8 %),

- pour l'expérience 2, le nombre de lignes de code à modifier a été équivalent dans les deux groupes (-3 % en C++), la productivité restant, comme dans l'expérience 1, plus importante en programmation C (+ 23 %).

3.4 Synthèse et portée des résultats

Le constat que nous pouvons tirer de ces observations sur la mise en œuvre des méthodes orientées objet dans la conduite d'un projet d'informatisation se révèle assez surprenant puisqu'en opposition avec les arguments souvent avancés par les tenants ou les apôtres de ces "nouvelles" méthodes. Ces arguments reposent sur les potentialités des nouveaux concepts proposés pour la simplification de la démarche d'analyse (héritage) et pour la diminution de la maintenance (mise en œuvre de l'encapsulation et de la communication par messages).

Les deux expériences que nous avons conduites convergent plutôt sur un accroissement (en temps et en charge) des phases d'analyse, de conception et de réalisation, conduites sur ces nouveaux principes conceptuels. Leur mise en œuvre s'avère, dans les conditions de l'expérience, poser quelques problèmes aux concepteurs, tant dans la représentation du système à étudier que dans la cohérence du système à concevoir et à mettre en place.

Dans les phases de maintenance, il semblerait que l'impact de l'utilisation des méthodes objet soit un peu plus tangible mais dépende fortement de la qualité de la conception et de la réalisation. L'analyse de la supériorité très relative des approches orientées objet en matière de maintenance peut être un peu biaisée par rapport à ce que l'on pourrait constater sur le terrain et mériterait d'être approfondie : le protocole d'un seul changement significatif n'est sûrement pas suffisant pour mettre en évidence des phénomènes de courbe d'apprentissage ; de plus, le fait que la maintenance soit faite par les créateurs de l'application ne permet de tirer aucune conclusion sur l'un des aspects les plus importants de ce processus qui réside dans la reprise du dossier par d'autres développeurs.

L'ensemble de ces résultats doit être nuancé sur trois autres points :

- ils ne portent que sur des situations de conception précises : projet court, concepteurs débutants, absence des utilisateurs, remplacés par un professionnel de la conception, etc.

Ce n'est que par la multiplication d'expériences de ce type dans différents environnements sur des situations de plus en plus proches de la réalité du terrain que notre connaissance sur l'utilisation des méthodes de conception s'accroîtra. De ce point de vue, cette recherche ne doit être considérée que comme une contribution à cette problématique ;

- ils n'intègrent pas l'aspect outil de génie logiciel puisque les méthodes ont été mises en œuvre "à la main". On peut penser que ces résultats se trouvent, de ce fait, éloignés de la situation réelle des entreprises, de plus en plus nombreuses, qui utilisent un outil informatisé pour la mise en œuvre de leur méthode de conception : c'est, dans certains cas d'ailleurs, l'acquisition par l'entreprise d'un outil informatisé d'aide à la conception qui constitue un puissant

facteur pour l'impulsion de l'emploi d'une méthode.

L'utilisation d'ateliers de génie logiciel a sciemment été exclu lors de l'expérience afin d'éviter que les outils ne viennent biaiser la comparaison des qualités intrinsèques des méthodes : on peut en effet aisément supposer que des outils supportant une méthode vieille de quinze ans comme MERISE soient supérieurs (finition, complétude, ergonomie, fiabilité) à ceux basés sur une méthode plus récente ; *a posteriori*, on peut dire que ces différences supposées n'auraient donc fait qu'amplifier les résultats obtenus. De plus, nous avons estimé que l'utilisation d'outils d'aide à la conception, pour être efficace, passait par une bonne connaissance préalable de la méthodologie utilisée et de sa pratique par le concepteur ; sur les aspects "élaboration des principaux modèles d'analyse" qui étaient au cœur de l'expérience les ateliers de génie logiciel sont d'ailleurs considérés comme peu performants par les praticiens ;

- ils ne peuvent être que partiels puisque portant sur un seul projet, excluant de ce fait toute vérification du principe de réutilisation potentielle des résultats de l'analyse, à travers les concepts mis en œuvre dans les méthodes objet.

Un prolongement de la recherche, en cours actuellement, devrait permettre d'obtenir quelques résultats sur ce point précis et compléter les observations déjà effectuées. Bien que partiels et non encore suffisamment confirmés pour en faire état, les premiers relevés tendraient à montrer que la réutilisation est très faible voire inexistante. Celle-ci semblerait se limiter à une partie d'un objet pour les groupes OMT ; la réutilisation se révèle quasi identique en conception systémique et programmation impérative à travers la récupération de deux sous-

programmes initialement élaborés. Ces premières constatations sembleraient confirmer les résultats d'une analyse (non publiée à ce jour) que nous avons entreprise en 1993 chez MATRA MARCONI SPACE : sur trois cents modules développés dans cette organisation à partir de la méthode de conception HOOD en programmation ADA ou C++, seuls trois objets ont été réutilisés, deux d'entre eux ne représentant que des bibliothèques mathématiques classiques.

CONCLUSION

Les résultats de cette recherche, aussi partiels et contestables soient-ils quant à leur généralisation, tendent à nuancer le discours habituellement tenu dans la plupart des articles sur les méthodologies de conception des systèmes d'information (Castellani, 1991) (Rochfeld, 1991) (Bailin, 1989) (Ehrich et al., 1990) (Kappel et Schrefl, 1990) (Wand et Weber, 1989). Ces derniers insistent sur les avantages des méthodes orientées objet par rapport aux méthodes "classiques". Les deux expériences que nous avons conduites ne font pas ressortir, en termes de productivité de conception, de réalisation ou de maintenance, de gains réels liés à l'utilisation d'une méthode orientée objet. Il semblerait que, dans un certain nombre de situations, ce soit même le contraire.

Loin de remettre en cause l'intérêt des méthodes objet qui sur le plan conceptuel représentent un réel apport, cette recherche suggère deux réflexions complémentaires qui peuvent sembler triviales mais que l'on a trop tendance à oublier :

- **les "nouvelles" méthodes tendent à complexifier le processus de conception** : les modèles proposés sont de plus en plus abstraits, de

plus en plus rigoureux, de plus en plus complets pour tenter de rendre compte de la complexité des problèmes à représenter et à résoudre dans des environnements mouvants et évolutifs. Ils deviennent ainsi plus difficiles à comprendre et à manipuler par les acteurs concernés. La cohérence entre les différents modèles de représentation devient de plus en plus difficile à contrôler sans l'aide d'outils informatiques spécifiques ;

- **des prérequis sont nécessaires pour que l'utilisation de ces "nouvelles" méthodes soit efficace** : une formation accrue des concepteurs et des réalisateurs, une plus grande expertise passant par un apprentissage plus long et plus important, sont des conditions obligatoires pour que les avantages potentiels liés à l'utilisation de ces méthodes deviennent réels et tangibles. Il semblerait même que leur efficacité passe par un état d'esprit que les acteurs de l'organisation, concepteurs, réalisateurs, utilisateurs, décideurs, doivent acquérir et faire partager.

A travers les interrogations que cette recherche a pu soulever, nous avons voulu montrer que toute implantation d'une méthode de conception des systèmes d'information dans les organisations devait s'évaluer sur de multiples dimensions. Les qualités intrinsèques d'une méthode ou d'une classe de méthodes par rapport à une autre méritent certes d'être discutées et démontrées, mais les conditions de son applicabilité au sein de l'organisation sont, elles aussi, importantes à définir et à évaluer. Notre connaissance du terrain nous incite à penser que, pour le moment dans la grande majorité des organisations, les conditions d'une bonne utilisation des méthodes de conception orientée objet ne sont pas rassemblées pour que leur efficacité puisse réellement émerger dans

les faits. Ces conditions reposent essentiellement sur deux points :

- la "stabilité" des méthodes de conception orientée objet qui passe, à notre avis, par l'unification ou la convergence des concepts et des modèles présentés dans les différentes méthodes de ce courant méthodologique. Il semble que nous nous trouvons aujourd'hui à l'aube de cette maturité et qu'il est encore un peu tôt pour une entreprise de se lancer dans cette voie sans pouvoir compter sur des structures extérieures (SSCI par exemple) suffisamment expérimentées et fiables ;
- la "qualité" de la ressource humaine en conception orientée objet des systèmes d'information, disponible sur le marché. Cette compétence, encore rare de nos jours, passe par une for-

mation approfondie à ces nouveaux concepts : celle-ci n'est assurée que depuis peu de temps dans les cursus spécialisés et ne portera ses fruits que dans deux ou trois ans au maximum. Il faudra encore que les entreprises prennent conscience que cette compétence engendrera un coût qu'elles devront savoir supporter sous peine de seulement satisfaire à une mode qui, à travers une mauvaise utilisation des concepts véhiculés, pourrait s'avérer en définitive néfaste pour le développement rationnel des systèmes d'information : essayons de ne pas oublier, à ce sujet, les expériences malheureuses du passé (Ruiz, 1993) lors de l'introduction de MERISE dans certaines organisations peu préparées et ne disposant pas des ressources humaines nécessaires.

BIBLIOGRAPHIE

Bailin, S.C. (1989), An "Object-Oriented Requirements Specification Method", *Comm. ACM*, vol. 32, n° 5, Sept.

Booch, G. (1986), "Object-Oriented Developpement", *IEEE transactions on software engineering*, vol SE-12, Fev.

Castellani, X. (1991), *Le modèle de la méthode MCO d'analyse et de conception des systèmes d'objets*, in Actes du Congrès Inforsid, Paris, Juin.

Coad, P., Yourdon, E. (1990), *Object Oriented Analysis*, Yourdon Press, Computing series, Prentice hall, NJ.

Ehrich, H.D., Sernadas, A., Sernadas, C. (1990), "From Data Types to Object Types", *Journal of Information Processing and Cybernetics*, EIK 26.

Hertz, B., Moine, C., Sabatier, R. (1988), *Conception des systèmes d'information : six études de cas d'analyse informatique en entreprise*, Dunod infor, Paris.

Kappel G., Schrefl M. (1990), "Using an object-oriented diagram technique for the design of information systems", in Proceedings of the Int. Working Conf. on modelling of Information Systems, Noordwijkerhout, The Netherlands, Avr.

Rochfeld A. (1991), *Modèle externe de données et modèle externe objet*, 4^e congrès Génie Logiciel, Toulouse, Dec.

Ross D.T. (1977), "Structured analysis: a language for communicating ideas", *IEEE transactions on software engineering*, Jan .

Ruiz J. (1991), *Les pratiques du contrôle des systèmes d'information dans les organisations*, Thèse de doctorat, Montpellier, Oct.

Ruiz J., Magnier J.P. (1993) *Compétitivité des PME et maîtrise du système d'information*, Congrès International Francophone de la PME, Carthage, Oct 1993

Rumbaugh J. (1991), *Object oriented modeling and design*, Englewood cliffs, Prentice hall.

Tardieu H., Rochfeld A., Colletti R. (1983), *La Méthode Merise: Principes et outils*, Editions d'organisation, Paris, 1983

Wand Y., Weber R. (1989), An ontological Evaluation of Systems Analysis and Design Methods, In Proceedings of the IFIP TC8/WG8.1 Working Conference on the Object-Oriented Approach in Information System, Namur.

Weigiao H. (1992), "L'inadéquation des méthodes", *Bouhot et Legendre Publications*, Paris, Dec.

ANNEXES

Tableau 1 : Expérience 1 charges en jours/homme et écarts entre le groupe OMT et le groupe MERISE

	MERISE G1	OMT G1	OMT/MERISE
ANALYSE	30 j/h	45	1,5
CONCEPTION	20 j/h	24	1,2
TOTAL ANAL/CONC	50 j/h	69	1,38
REALISATION	100 j/h	110	1,1
TOTAL	150 j/h	179	1,3

Tableau 2 : Expérience 2 charges en jours/homme et écarts entre le groupe OMT et le groupe MERISE

	MERISE G2	OMT G2	OMT/MERISE
ANALYSE	40 j/h	52	1,3
CONCEPTION	25 j/h	30	1,2
TOTAL ANAL/CONC	65 j/h	82	1,26
REALISATION	130 j/h	140	1,08
TOTAL	195 j/h	222	1,14

Tableau 3 : Expérience 1 : résultats sur la maintenance des logiciels

	MERISE G1	OMT G1	OMT/MERISE
NOMBRE DE LIGNES ECRITES	11000	13000	+ 18 %
NOMBRE DE LIGNES MODIFIEES	2100	1730	- 18 %
CHARGE POUR MODIFIER	27 j/h	23 j/h	- 15 %
PRODUCTIVITE	78 l/h	74 l/j	+ 8 %

Tableau 4 : Expérience 2 : résultats sur la maintenance des logiciels

	MERISE G2	OMT G2	OMT/MERISE
NOMBRE DE LIGNES ECRITES	13000	15000	+ 15 %
NOMBRE DE LIGNES MODIFIEES	2500	2400	- 3 %
CHARGE POUR MODIFIER	32 j/h	40 j/h	+ 20 %
PRODUCTIVITE	78 l/h	60 l/j	+ 23 %