

Les Ateliers de Génie Logiciel face aux enjeux de l'ingénierie simultanée

Christian VARINARD

Maître de Conférences à l'IAE, Université Jean Moulin, Lyon III

RÉSUMÉ

Alors que de nos jours, le développement de logiciels est réalisé par des développeurs travaillant en parallèle et non plus en séquentiel comme souvent auparavant, on rencontre essentiellement sous la dénomination d'Atelier de Génie Logiciel, des produits conçus pour être utilisés par un développeur unique ou par plusieurs personnes travaillant de manière indépendante. A partir des résultats d'une étude réalisée sur les versions réseau de onze AGL, cet article démontre cependant que la situation est en train d'évoluer et que les fabricants proposent de plus en plus d'Ateliers de Génie Logiciel facilitant les activités des équipes de développement de logiciels en permettant non seulement une communication implicite à travers l'application réalisée mais aussi une communication explicite grâce à l'intégration d'outils de coopération.

Mots-clés : AGL, Groupware, Intégration, Partage des informations, Coopération.

ABSTRACT

While nowadays, software development is done by developers working in the same time and not in a sequential manner like often before, it is essentially possible to find on the market, products designed for a single developer or for several developers working independently. From the results of a study concerning the network versions of eleven CASE tools, this paper shows that the situation is just evolving and that vendors are now proposing many CASE tools which facilitate the work of the software development team in permitting not only an implicit communication through the developing product but also an explicit communication about the work by the use of integrated Groupware tools.

Key-words : CASE tool, Groupware, Integration, Information sharing, Cooperation.

INTRODUCTION

Depuis toujours, le développement de logiciels est essentiellement une activité d'équipe. Une étude de De Marco et Lister (1987) révèle, en effet, que les développeurs de logiciels passent environ 70 % du temps consacré à la réalisation de projets de grande taille, à travailler en groupe. Une autre étude du même type (Jones, 1986) indique que les activités réalisées en équipe représentent 85 % des coûts des gros systèmes logiciels. Par ailleurs, il est bien connu que les équipes de développement consacrent une grande partie de leurs ressources au maintien de la communication et du contrôle entre les membres qui les composent (Sellars, 1987).

Ainsi que le signale De Terssac (1996), les activités de conception (quelle que soit la nature du produit ou du service à concevoir) ont toujours fait l'objet d'une structuration. On est cependant passé progressivement, d'une organisation séquentielle du travail des membres des équipes de conception à un nouveau mode de structuration que l'on peut qualifier d'ingénierie concurrente (ou simultanée). Midler (1996) désigne ainsi un mode de structuration né au Japon qui a émergé dans l'industrie manufacturière européenne dans la fin des années 80 et qu'il caractérise en trois points :

- l'affirmation d'un nouvel acteur, le directeur de projet, qui incarne la finalisation du processus de production et dont le rôle ne se limite pas à la coordination des expertises utilisées dans la conception. Cet acteur a notamment une responsabilité importante sur tout ce qui est susceptible d'affecter le projet et

peut éventuellement en changer l'orientation ;

- un dialogue et une négociation entre les différentes expertises tout au long du processus de conception. En effet, l'irréversibilité d'un modèle séquentiel ne permet pas aux acteurs impliqués en aval du processus de conception de peser dans les choix alors que l'ingénierie simultanée permet la prise en compte de tous les points de vue très tôt dans le processus ;
- la combinaison de deux modes de coordination : une coordination procédurale qui consiste en un contrôle des moyens et du processus de production (conformation à un mode opératoire, obligation de jouer la transparence, etc.) et une coordination par des objectifs de résultats qui correspond à des engagements des acteurs sur les buts à atteindre.

Aujourd'hui, la conception de logiciels de gestion (tout comme celle d'autres produits d'ailleurs) sous-entend donc l'intervention d'un acteur collectif composé de différents types de spécialistes correspondant à des contraintes variées. Les performances au niveau de la productivité du développement dépendent donc en partie, de la qualité de la coopération entre les acteurs concernés. Hatchuel (1996) la définit au travers de la notion de "prescription réciproque". Cette notion signifie que les différents acteurs se trouvent tour à tour en position de prescripteurs ou d'opérateurs selon les problèmes rencontrés. Cela implique donc l'existence au départ d'un ordre de la coopération qui se redéfinit au fur et à mesure de la conception. Il est cependant clair que la réversibilité

du rapport de prescription s'attène au fur et à mesure de l'avancement du projet. Cette forme de coopération repose également sur l'idée que les concepteurs peuvent partager des modélisations communes et donc que l'activité collective peut être facilitée par l'utilisation d'un référentiel commun mettant à la disposition de tous certains formalismes.

Comme un groupe de travail est un ensemble d'individus agissant de concert afin d'atteindre un objectif (Bikson, Eveland et Gutek, 1989), on peut considérer comme tel, un ensemble de développeurs qui travaillent à la réalisation d'un système complexe pendant une longue période. Afin de faciliter les activités des groupes de travail, une technologie de l'information récente (apparue officiellement en France pour la première fois au printemps 1993) est de plus en plus utilisée : il s'agit du Groupware qui peut être défini comme "l'ensemble des technologies et méthodes de travail associées qui, par l'intermédiaire de la communication électronique, permettent à un groupe engagé dans un travail collaboratif et/ou coopératif, le partage de l'information sur un support numérique" (Courbon et Tajan, 1997). Cette technologie implique donc, tout à la fois, un travail de groupe et un environnement partagé.

Les Ateliers de Génie Logiciel (AGL) sont des collections d'outils et de composants de génie logiciel regroupés autour d'un référentiel central, utilisés par les informaticiens afin d'automatiser tout ou partie du cycle de vie des logiciels. Correctement implantés et utilisés (Varinard et Reix, 1997), ils constituent une aide précieuse pour spécifier, concevoir et fabri-

quer les systèmes logiciels et peuvent, notamment, permettre d'accroître la productivité des développeurs et d'améliorer la qualité des applications produites. Jusqu'à présent, les AGL disponibles sur le marché ont presque toujours été conçus pour un utilisateur unique. Dans le domaine particulier du développement de logiciels, la capacité à coordonner une équipe est pourtant un problème critique et, ainsi que le mettent en valeur Grochow (1988) ou Brès (1994), il semble maintenant indispensable d'intégrer les principales fonctionnalités du Groupware dans les Ateliers de Génie Logiciel pour qu'ils ne demeurent pas des technologies uniquement bien adaptées à la réalisation de logiciels "par des équipes d'une personne". Une étude réalisée par Crosslin, Bergin et Stott (1993) indique d'ailleurs que la migration vers des AGL multi-utilisateurs est considérée par les experts européens et américains comme une des évolutions prioritaires de cette technologie pour les années à venir.

L'objectif de cette étude est de réaliser une investigation exploratoire au sujet des fonctionnalités du Groupware qu'il serait intéressant d'intégrer aux Ateliers de Génie Logiciel pour qu'ils deviennent ce que l'on pourrait appeler des "outils de support collaboratif" et qu'ils permettent donc de répondre aux enjeux d'une conception plus courte. Pour la réaliser, dans une première partie, nous serons amenés à aborder plusieurs questions de caractère méthodologique que ces investigations révèlent nécessairement. Notamment, nous définirons un instrument d'évaluation de l'efficacité des AGL susceptibles de supporter les "travaux de groupes". Dans une seconde

partie, nous pourrions tester cet instrument d'évaluation en l'utilisant pour déterminer l'efficacité des versions "réseaux" de onze des principaux Ateliers de Génie Logiciel du marché français.

1. ÉVALUATION DU SUPPORT FOURNI PAR UN ATELIER DE GÉNIE LOGICIEL POUR LES ACTIVITÉS DE GROUPE

Depuis qu'ils sont apparus aux Etats-Unis en 1984, les systèmes de Groupware font l'objet d'un champ d'études pluridisciplinaires appelé Computer Supported Cooperative Work (CSCW), soit littéralement, en traduction française, "travail coopératif assisté par ordinateur". Cette discipline récente étudie les différents mécanismes individuels et collectifs du travail en groupe. Les spécialistes proposent de considérer le Groupware comme une classe d'applications disposant de trois caractéristiques fondamentales puisqu'elles facilitent la communication, la collaboration et la coordination entre les individus composant le groupe (Levan et Liebman, 1994). Dans un premier paragraphe, nous proposerons un modèle d'architecture conceptuel d'un AGL supportant les activités des équipes de développement de logiciels après avoir préalablement différencié les différents types de coopération possibles entre les membres d'un groupe de travail ; dans un deuxième paragraphe, à partir du modèle proposé, nous présenterons un outil d'évaluation de l'efficacité des AGL pour soutenir les activités des équipes de développement de systèmes.

1.1. Définition d'un modèle d'architecture conceptuel d'un AGL multi-utilisateurs

Si on considère la volonté d'offrir une assistance aux activités des groupes de travail, ceux-ci sont différenciés dans la littérature consacrée au Groupware, principalement à partir de caractéristiques spatiales et temporelles. Dans ce domaine, les travaux de Johansen (1988) font référence. Cet auteur propose une matrice Moment/Lieu permettant de caractériser quatre types de collaboration (figure 1). D'un point de vue spatial, les interactions au sein du groupe peuvent avoir lieu au même endroit ou au contraire dans des lieux éloignés. Par ailleurs, si on examine le travail de groupe au niveau temporel, les individus qui composent le groupe peuvent effectuer une tâche partagée au même moment (on parlera alors d'échange synchrone) ou à des instants différents (échange asynchrone).

Cf. figure 1

Dans chacune des cases de la matrice, figurent un mode de travail conventionnel et, en dessous, un exemple correspondant d'apport de la technologie Groupware susceptible de faciliter, voire d'améliorer nettement les conditions de la collaboration.

Cette classification des types de coopération au sein d'un groupe est celle qui est la plus fréquemment retenue (voir par exemple Levan et Liebman, 1994 ; Vessey et Sravanapudi, 1995) mais on peut également, ainsi que le suggèrent Courbon et Tajan (1997), en retenir d'autres fondées, par exemple, sur la taille et la durée

<p>Même moment / Lieux différents</p> <p>TÉLÉPHONE</p> <p>↓</p> <p>Vidéo-conférence</p>	<p>Moments différents / Lieux différents</p> <p>COURRIER</p> <p>↓</p> <p>Messagerie électronique</p>
<p>Même moment / Même lieu</p> <p>RÉUNION</p> <p>↓</p> <p>Salle de décision électronique</p>	<p>Moments différents / Même lieu</p> <p>PLANNING</p> <p>↓</p> <p>Agenda partagé</p>

**Figure 1 : Matrice Moment/Lieu de travail
(adaptée de Johansen)**

du groupe ou sur ses modes de communication. Quelle que soit celle que l'on retient, le Groupware dispose d'un potentiel correspondant à chacune des formes de travail de groupe qu'il est possible d'identifier car il situe son originalité au niveau de trois concepts clés : le partage de l'information, la communication de groupe et le travail collaboratif.

D'une manière assez surprenante, les travaux de recherche proposant des relations entre les fonctionnalités du Groupware et le processus de développement des logiciels sont assez peu nombreux. Les quelques auteurs qui s'y sont intéressés font remarquer, comme Alavi (1993) ou Vessey et Sravanapudi (1995), que le ralentissement récent des ventes des AGL peut sans doute s'expliquer par le fait qu'ils ne disposent pas

d'un nombre assez important de possibilités au niveau du travail d'équipe, ce qui limite leur utilité dans les organisations où le développement des logiciels est le plus souvent réalisé en équipes. Certes, actuellement de nombreux vendeurs tentent de prendre position sur ce marché émergent mais les produits qu'ils proposent présentent des degrés très variables de support pour les activités d'ingénierie en équipe (Keyes, 1992). A l'exception notable de Vessey et Sravanapudi (1995), les auteurs n'ont pas jusqu'alors examiné l'intérêt des fonctionnalités du Groupware tout au long du processus de développement des systèmes. Ils se sont presque tous contentés d'envisager l'utilisation de la salle de décision électronique au cours de la phase de conception (voir par exemple Bostrom et Anson, 1988 ; Jarvenpaa, Rao

et Huber, 1988 ; Kozar et Zigurs, 1992 ; Hayne et Ram, 1995). Pour ce qui nous concerne, à l'image de ce qu'ont proposé Vessey et Sravanapudi, nous avons choisi dans cet article, d'examiner le support qui peut être tiré du Groupware au cours de l'ensemble du cycle de développement des applications.

Les fonctionnalités du Groupware peuvent être regroupées en cinq grandes classes (Courbon et Tajan, 1997). Ces cinq activités, susceptibles de faciliter le travail des équipes de développement au niveau de toutes les phases du processus de développement d'un logiciel, nous allons les présenter succinctement et successivement :

La communication interpersonnelle : Le Groupware donne la possibilité à des individus de communiquer entre eux, par des moyens logiciels via une communication électronique. La forme la plus habituelle de communication interpersonnelle dans le Groupware est la *messagerie électronique* qui donne la possibilité d'envoyer des messages dont les contenus sont le plus souvent des textes entrés au clavier d'un poste de travail. On peut ajouter que des fonctionnalités complémentaires de la messagerie permettent, par exemple, d'adresser le même message à plusieurs personnes ou de le coder pour qu'il ne soit accessible qu'à un ou des destinataires précis. Cette technique permet donc aux différents membres d'un groupe de travail d'avoir un dialogue élémentaire même lorsqu'ils ne travaillent pas au même endroit et au même moment. Elle ne leur permet cependant pas d'établir à distance et de manière asynchrone une discussion sur un sujet précis. Le forum

de discussion électronique est l'autre outil de communication interpersonnelle qui répond à ce besoin puisqu'il autorise les échanges entre participants sous forme de questions-réponses et permet, à tout moment, à un membre du groupe de s'insérer dans la discussion.

La coordination : Notons préalablement qu'on évoque ici aussi bien la coordination dans le temps (partage de l'agenda de chaque membre du groupe) que la coordination dans l'espace (aide à l'organisation de réunions) ou encore dans les tâches (suivi de l'avancement des tâches de chacun et consolidation du travail effectué). *L'agenda électronique partagé, le suivi d'affaires et la gestion de projets* sont les outils du Groupware qui correspondent à ces trois aspects de la coordination. Lorsque le projet est important, les membres du groupe de travail se voient attribuer des tâches respectives par le responsable du projet. Dès lors, il est indispensable qu'il ait la possibilité de suivre chacune d'entre elles et qu'il dispose de certaines possibilités pour en réaliser la consolidation. De la même manière, il est intéressant que les différents membres d'un groupe puissent avoir une idée de l'avancement du travail des collègues de l'équipe et qu'ils aient aussi la possibilité de demander leur approbation vis-à-vis de leur propre travail.

La collaboration : Lorsque la mission d'un groupe correspond seulement à la juxtaposition ou à la consolidation de plusieurs tâches individuelles, les mécanismes de coordination suffisent pour la supporter. En revanche, fréquemment, le travail d'un

groupe correspond à une imbrication de tâches individuelles avec d'autres activités simultanées dans la mesure où le déroulement d'un travail particulier dépend bien souvent du résultat obtenu par le travail de beaucoup d'autres acteurs. On est dans le cas d'un processus de collaboration dans lequel plusieurs acteurs doivent agir et prendre des décisions de manière synchrone. Le Groupware apporte à ce niveau, différents outils tels que la *réunion électronique*, la *vidéo-conférence* ou encore *l'édition conjointe* (les divers protagonistes collaborent à la réalisation d'un document partagé qui conservera les traces de l'ensemble des différents apports).

La gestion de la mémoire de groupe : Le travail de groupe suppose évidemment la mise en commun des informations apportées ou élaborées par chacun. D'une part, il faut donc que ces informations puissent être codées, saisies, validées, stockées, structurées et organisées pour leur exploitation ultérieure. D'autre part, il faut que l'on puisse y accéder pour les modifier ou effectuer sur ces données des traitements plus ou moins automatisés. Au niveau du Groupware, on parle souvent, à ce propos, de *base de documents* mais on recoupe alors nettement la notion de *dictionnaire (Repository)* qui est l'élément central des AGL. Cependant, le Groupware n'apporte pas seulement la possibilité de partager les informations. En effet, la mémoire de groupe doit aussi mémoriser les interactions des membres de l'équipe sous la forme de communications exprimant par exemple des demandes, des négociations ou encore l'historique de la collaboration au sein de l'équipe de

travail. Ces possibilités constitueraient le véritable apport du Groupware en regard de ce que proposent les principaux AGL "mono-utilisateurs" avec leur référentiel central.

L'automatisation des processus administratifs : La circulation des documents est un aspect important de la communication. Elle peut être plus ou moins structurée, c'est-à-dire suivre des chemins définis à l'avance avec plus ou moins de rigueur ou de libre arbitre de la part des acteurs impliqués. Les membres d'un groupe de travail sont parfois amenés à élaborer des documents ou des formulaires dont la diffusion est grandement facilitée par les systèmes de *Workflow* du Groupware.

Cette rapide présentation nous permet de constater que l'intégration des principales fonctionnalités du Groupware aux Ateliers de Génie Logiciel pourrait avoir de multiples avantages pour les équipes de développement. Dès lors, fort de cette constatation, nous proposons, dans cette étude, une architecture conceptuelle d'AGL multi-utilisateurs à trois niveaux organisés autour d'un référentiel central, fondée sur la classification des fonctionnalités que nous venons de présenter (figure 2). La hiérarchisation proposée est fondée sur la spécificité des tâches à accomplir :

- les outils de niveau 1 sont spécifiques à une tâche particulière ;
- les outils de niveau 2 sont nécessaires à l'accomplissement de tâches qui requièrent le partage de données ;
- les outils de niveau 3 sont indépendants des tâches à accomplir.

Niveau 3 : outils de coopération
Niveau 2 : outils de gestion de la mémoire de groupe
Niveau 1 : outils de développement

Figure 2 : Modèle d'architecture conceptuelle d'AGL multi-utilisateurs

Les outils de développement dont il est question dans le niveau 1 du modèle sont les outils classiques contenus dans les AGL (voir par exemple, pour plus de précisions, Varinard, 1997). Leur objectif est d'apporter un support automatique aux développeurs au cours des différentes phases du cycle de vie des logiciels. Il peut donc s'agir soit d'outils de conception qui offrent une assistance lors des étapes initiales du processus de développement des logiciels, soit d'outils de réalisation apportant une aide lors des phases ultimes du cycle de développement, soit encore d'outils d'aide à la gestion du développement des Systèmes

d'Information qui sont des outils annexes tels que les gestionnaires d'évolution ou les gestionnaires de projets, pouvant être utilisés à tout moment au cours du cycle de vie des logiciels. Plus précisément tous ces outils conduisent à l'automatisation d'une méthodologie de développement (notion de compagnon méthodologique). L'objectif de ces différents outils est donc d'offrir un soutien pour conduire une tâche particulière alors que s'agissant des outils des deux autres niveaux, la finalité est de permettre aux membres de l'équipe de développement d'interagir pour mener à bien la tâche envisagée.

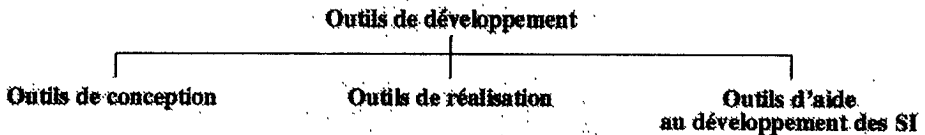


Figure 3 : Modèle d'environnement de développement

Le référentiel central (également appelé dictionnaire) d'un Atelier de Génie Logiciel supporte toutes les phases du cycle de vie couvertes par la technologie, c'est-à-dire, dans le cas d'un AGL intégré, la réalisation, la conception et la gestion du développement du système d'information. Il constitue donc le support de mémorisation

des propriétés et des entités du modèle du système d'information manipulé par le logiciel en création. Cela signifie qu'il contient non seulement la représentation des données d'une application mais aussi les traitements, les écrans, les états ou encore tous les objets relevant de l'ingénierie et de la conduite du projet.

Le dictionnaire, qui doit être fonctionnellement unique, constitue donc la mémoire de l'équipe de développement. Dans un AGL intégré, il est absolument nécessaire puisque ce qui a été fait dans la conception est utilisé au cours de la phase de réalisation du logiciel. Ainsi par exemple, si on se place dans une optique Bases de données, les descriptions des données sont-elles utilisées par les développeurs pour la génération des tables et des contraintes d'intégrité. En matière de traitement, les programmeurs utilisent les descriptions des écrans ou les enchaînements réalisés en phase de conception pour générer les écrans et les enchaînements correspondants en phase de réalisation. Il en va de même pour les règles de gestion qui sont écrites en phase de conception et qui sont récupérées et traduites en phase de réalisation. Enfin, en phase de maintenance d'une application, les développeurs doivent avoir accès aux différents composants d'un logiciel afin de réaliser les modifications ou les adaptations nécessaires.

Le référentiel central a également un rôle à jouer dans le domaine de la réutilisation des composants logiciels. Plusieurs études ont montré qu'une faible part des logiciels produits peut être considérée comme unique. Cela signifie que, très souvent, des logiciels nouveaux contiennent des fragments identiques à ceux déjà réalisés dans des logiciels anciens. Il est donc essentiel de pouvoir disposer dans le dictionnaire, d'une bibliothèque de composants logiciels (descriptions de données, descriptions de traitements, desins d'écrans, modules de traitement) disponibles pour pouvoir être réutilisés dans d'autres appli-

cations que celles pour lesquelles ils ont été élaborés.

Les outils de niveau 2 sont des outils dont l'objectif principal est de permettre le partage des informations liées au logiciel réalisé et contenues dans le référentiel central. Cette classe d'outils ne correspond donc pas à ceux qui permettent la constitution même du référentiel central (c'est-à-dire les éditeurs de modèles ou de code qui sont considérés comme des outils de développement) mais à ceux qui permettent sa gestion. Les principales fonctions remplies par ces outils sont celles rencontrées habituellement lorsque l'on parle de sécurité et de confidentialité des données au sein des systèmes de gestion des bases de données. Il s'agit tout d'abord des outils de gestion des droits d'accès aux données qui conduisent à restreindre cet accès aux seuls utilisateurs légitimes. Il s'agit ensuite des outils de contrôle nécessaires pour préserver l'intégrité des données. Ces derniers permettent tout à la fois qu'un développeur soit averti lorsqu'un changement dans la conception va affecter son travail, que les conflits éventuels soient résolus lorsque plusieurs membres de l'équipe de développement travaillent en même temps sur une même tâche et que plusieurs utilisateurs puissent travailler au même moment sur des tâches utilisant les mêmes données. Il s'agit enfin des outils de "surveillance" (monitoring) qui permettent d'une part, de contrôler l'évolution du logiciel grâce à l'enregistrement de tous les changements dans la conception et/ou dans le codage et d'autre part, de surveiller les activités des utilisateurs en enregistrant leurs connexions, leurs droits d'accès ou encore leurs temps de travail à chaque session (figure 4).

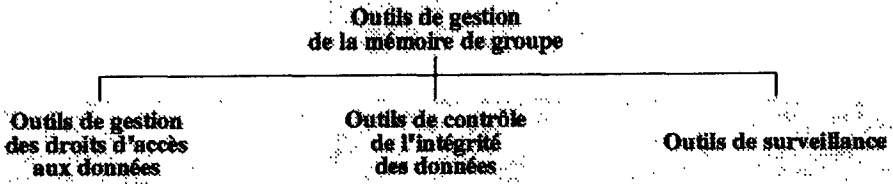


Figure 4 : Modèle d'environnement de gestion de la mémoire de groupe

Par le partage de l'information, les outils de niveau 2 permettent seulement une communication implicite des différents membres de l'équipe de développement à travers le logiciel développé. Il leur est possible de se contenter de cette seule forme de communication dès lors que le problème à résoudre peut être facilement modélisé avec les outils classiques (éditeurs de modèles par exemple) fournis par l'Atelier de Génie Logiciel. On se rapproche ici du concept de problème structuré, cher aux spécialistes des processus de prise de décision (voir par exemple Mintzberg, Rainsinghani, Theoret, 1976 ou Simon, 1983). Ces auteurs considèrent qu'un problème est structuré dès lors qu'il est totalement exprimé par un modèle calculable et qu'ainsi, le processus de prise de décision se réduit à un calcul. A partir de là, les problèmes de décision peuvent être différenciés en fonction de leur degré de structuration qui dépend de la capacité du décideur à conceptualiser le monde réel et donc de sa capacité d'abstraction et de modélisation. Ainsi que le soulignent Lévine et Pomerol (1990), la machine (par l'intermédiaire de l'Atelier de Génie Logiciel) est douée pour les problèmes naturellement structurés tandis que l'avantage reste à l'homme pour les problèmes à structuration non évidente. Ce dernier est en effet plus habile : lorsqu'un

problème est mal structuré, sa modélisation nécessite des tâtonnements, des retours en arrière, des jugements de valeur ou des recherches de solutions existantes qui aboutissent à une solution de type "politique" plutôt qu'à une solution de type "algorithmique" (Reix, 1995). Dans le domaine de la modélisation des systèmes d'information, les développeurs connaissent également des difficultés dès lors que le problème qu'ils ont à résoudre est peu ou mal structuré. Il s'agit notamment du cas où la réalisation d'une tâche particulière est dépendante de l'accomplissement préalable (voire simultané) d'un ou plusieurs autres travaux. Dans cette situation, les membres de l'équipe de développement ont besoin de communiquer de manière explicite au sujet du problème qu'ils rencontrent afin de participer à sa résolution commune.

Les outils de niveau 3 sont des outils dont l'utilité essentielle est de permettre une étroite coopération entre les membres de l'équipe de développement. Ainsi que le montre la figure 5, cette catégorie contient tout d'abord des outils permettant aux développeurs de coordonner leurs tâches respectives en facilitant la tenue de réunions classiques (outils de gestion partagée des emplois du temps des membres de l'équipe), en aidant à l'optimisation des délais et

des coûts de production des logiciels (outils de gestion de projets). D'autres outils, moins courants, permettent de suivre efficacement l'évolution du travail des différents membres de l'équipe de développement (outils de suivi d'affaire). Elle comprend ensuite des outils

autorisant une collaboration des individus en temps réel (salle de réunion électronique) ou en temps différé (édition conjointe). Enfin, elle comporte des outils permettant de communiquer de manière asynchrone (messagerie électronique essentiellement).

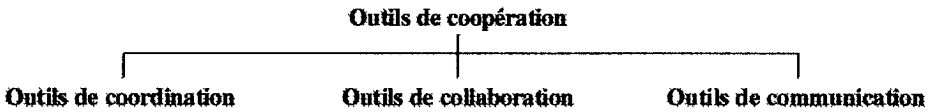


Figure 5 : Modèle d'environnement coopératif

Nous noterons, pour en terminer avec cette revue des outils de Groupware susceptibles d'être intégrés à un AGL multi-utilisateurs, que nous n'avons retenu aucun outil de la classe cinq (automatisation des processus administratifs), essentiellement parce qu'ils ne nous semblent pas présenter un apport intéressant dans le cadre des activités de développement de logiciels.

C'est en partant de ce modèle d'architecture conceptuelle d'un AGL multi-utilisateurs que nous venons de décrire, qu'il nous a été possible d'élaborer un outil d'évaluation du support amené par un AGL pour les travaux de groupes. C'est à sa description que nous consacrerons le deuxième développement de cette section.

1.2. Proposition d'un outil d'évaluation de l'efficacité d'un AGL destiné à supporter le travail d'une équipe de développement

Pour évaluer le support que peut constituer un Atelier de Génie Logiciel s'agissant du travail d'une équipe de développement de

logiciels, nous avons été amenés à élaborer un outil spécifique. Laisant volontairement de côté les outils classiques de développement que l'on trouve maintenant dans la quasi-totalité des AGL, l'outil que nous avons réalisé tente d'évaluer le support de la technologie au niveau du partage de l'information et de la coopération entre les membres de l'équipe de développement. Il correspond à un ensemble de ratios calculés en rapportant le nombre de points de fonction (ou fonctionnalités) apportés par un Atelier de Génie Logiciel au nombre de points de fonction attendus (définis ci-après). Notons que l'analyse des points de fonction (voir, pour plus de précisions sur la méthode, Habrias, 1994 ou Dreger, 1989), est à l'origine, une méthode de mesure synthétique de la taille des logiciels qui fut créée pour éliminer la subjectivité et l'ambiguïté de la méthode qui consiste à comptabiliser les lignes de code. Elle évalue la taille d'un logiciel en comptabilisant combien de fonctionnalités ont été apportées aux utilisateurs.

La littérature concernant l'utilisation des fonctionnalités du Groupware au sein d'un Atelier de

géné Logiciel étant très peu nombreuse, nous avons dû nous référer à une documentation plus spécifique (cf. bibliographie) consacrée aux AGL, au Groupware, aux bases de données ou encore aux environnements d'ingénierie des logiciels (SEE, c'est-à-dire Software Engineering Environment pour les auteurs de langue anglaise).

Les concepteurs de logiciels ne travaillent pas indépendamment les uns des autres, ni indépendamment des objets réalisés précédemment dans l'entreprise. Afin d'assurer la cohérence de la description du système d'information de l'organisation, il est important que la base d'information partagée soit fonctionnellement unique et qu'on y ait accès en temps réel. Les données partagées se présentent sous la forme de diagrammes, de structures de données, de processus ou de procédures. En nous référant au modèle proposé à la figure 4, nous déterminerons le support de l'AGL concernant le partage de l'information entre les membres de l'équipe de développement en termes de contrôle d'accès, de contrôle d'intégrité des données et de surveillance du produit et de l'utilisateur. Reprenons rapidement ces trois éléments :

Les outils de gestion des droits d'accès permettent d'assurer la confidentialité des différents objets contenus dans le référentiel central. Ceci correspond à un accès sélectif aux données pour les utilisateurs en fonction de leur qualité ou de leur rôle dans l'équipe de développement. Il existe donc, dans cette hypothèse, divers niveaux de droits d'accès et les possibilités de contrôles d'accès peuvent être plus ou moins sophistiquées. Ainsi, on peut no-

ter les solutions suivantes, d'ailleurs non limitatives :

- dans le cas où plusieurs projets seraient développés à la même période, des droits d'accès différents pour chaque projet peuvent être définis pour un développeur ;
- dans le cadre d'une équipe de développement, les droits d'accès à un projet donné peuvent être les mêmes pour tous ou différents pour chaque membre ;
- les utilisateurs peuvent être régulièrement obligés de changer leurs mots de passe ou de ne les changer que lorsqu'ils le souhaitent ;
- il peut être possible de définir pour certaines personnes, des droits d'accès en lecture seule ;
- on peut, ou non, limiter l'accès de certains membres du groupe aux seuls objets qu'ils ont créés ;
- on peut définir ou non des périodes d'accès limitées dans le temps pour certains utilisateurs.

Les outils de contrôle d'intégrité des données sont nécessaires en raison de la possibilité pour les membres de l'équipe de développement de travailler en même temps sur les différents objets concernant un projet (diagrammes, structures de données, modèles, etc.). Les problèmes essentiels se posent lorsqu'un objet particulier est utilisé sur plusieurs écrans en même temps. Dans ce cas, pour préserver l'intégrité des données, il faut alors des mécanismes permettant d'assurer leur cohérence et des mécanismes de gestion des éventuels conflits. Au niveau du contrôle de cohérence, plusieurs possibilités sont envisagées par les chercheurs :

- utilisation obligatoire d'une terminologie standard ;
- recherche des redondances dans les noms d'objets ;
- notification à un développeur de tout changement dans la conception risquant d'affecter son travail ;
- possibilité de protection contre le changement de tout ou partie d'un objet.

Les mécanismes de gestion des conflits doivent fonctionner lorsque plusieurs utilisateurs travaillent en même temps sur la même tâche ou lorsqu'ils tentent d'accéder au même objet. Ils doivent aussi permettre l'accomplissement simultané et sans problème de plusieurs tâches utilisant les mêmes données. Les solutions dans ce domaine se situent au niveau du verrouillage des diagrammes ou des structures des données dès lors qu'un utilisateur les utilise. On notera toutefois que pour éviter les problèmes de conflits, il est également envisageable d'interdire purement et simplement l'accès simultané aux entités du dictionnaire.

Les outils de surveillance (ou de "monitoring" pour utiliser le même terme que les auteurs de langue anglaise) permettent non seulement de contrôler l'évolution du produit mais aussi les activités des membres de l'équipe de développement. Au niveau du produit, cela consiste en un enregistrement de la création et de toutes les modifications réalisées sur les éléments d'un projet en les datant et en les "signant". Au niveau des développeurs, il s'agit de mémoriser les temps de travail de chacun sur un projet donné, la taille des fichiers les concernant, leurs

droits d'accès ou encore les dates, heures et durées de connexion.

Il reste maintenant à préciser les différents éléments qui nous permettront d'évaluer la manière dont l'information peut être partagée dans le cadre d'un Atelier de Génie Logiciel. Ils sont décrits dans le tableau 1 en page suivante.

En nous référant au modèle présenté à la figure 5, il nous faut essayer d'évaluer l'assistance fournie par l'AGL s'agissant de la coopération et cela en termes de coordination, de collaboration et de communication :

Les outils de coordination ont pour finalité l'ajustement du travail de chacun des membres de l'équipe de développement dont les tâches individuelles doivent être synchronisées afin que le travail prenne la forme d'un produit unique. Ainsi que nous l'avons déjà indiqué précédemment, il s'agit d'une coordination à trois niveaux : le temps, l'espace et les tâches. Pour que la ressource globale, qu'est le temps collectif, soit gérée efficacement, l'agenda électronique partagé peut être utilisé. Il optimise la gestion du temps des individus et du groupe en mémorisant les rendez-vous et les plages libres de chacun d'eux. Lorsque l'on désire organiser une réunion entre plusieurs membres de l'équipe, il détermine instantanément leurs disponibilités ainsi que les horaires auxquels les salles de réunion sont utilisables (coordination dans l'espace), ce qui améliore considérablement l'efficacité du travail. S'agissant de la synchronisation des tâches, le Groupware permet ce que l'on nomme le suivi d'affaires pour que les développeurs puissent à la fois visualiser l'évolution du tra-

I. Gestion des droits d'accès	
1-1.	Est-il possible de définir pour un individu, des mots de passe différents pour des projets différents développés à une même période ?
1-2.	Peut-on définir pour un projet déterminé, des droits d'accès différents pour les différents membres d'une même équipe de développement ?
1-3.	Est-il possible de forcer les utilisateurs de l'AGL à changer régulièrement de mots de passe ?
1-4.	Peut-on définir pour certains membres de l'équipe, des droits d'accès en lecture seule ?
1-5.	Peut-on limiter, pour un certain nombre de membres de l'équipe de développement, l'accès aux seules entités qu'ils ont créées ?
1-6.	Peut-on définir des périodes d'accès limitées dans le temps pour certains développeurs de l'équipe ?
II. Contrôle d'intégrité des données	
2-1.	Y a-t-il des standards imposés pour créer les objets de la base (modèles, diagrammes de flux, etc.) ?
2-2.	Lorsqu'un nouvel objet est créé, y a-t-il une recherche automatique des objets de même nom ou d'un nom très proche afin d'alerter l'utilisateur ?
2-3.	Un développeur est-il systématiquement prévenu lorsque son travail va être affecté par une modification voire par une suppression dans le référentiel central ?
2-4.	Est-il possible d'empêcher la modification de tout ou partie d'un objet ?
2-5.	Est-il possible d'empêcher la suppression d'un objet ?
2-6.	Plusieurs utilisateurs peuvent-ils accéder conjointement au même objet ?
2-7.	Lorsque plusieurs développeurs accèdent en même temps à un objet du dictionnaire y a-t-il un mécanisme de verrouillage automatique ?
2-8.	Dans l'hypothèse d'un accès simultané à un objet, est-il possible de savoir quel autre développeur est à l'origine du verrouillage ?
III. Surveillance de l'évolution du produit et des activités des développeurs	
3-1.	Est-il possible de retrouver qui a créé un objet donné du dictionnaire et quand il l'a été ?
3-2.	Est-il possible de retrouver des traces de toutes les modifications d'un objet donné du dictionnaire (date, auteur) ?
3-3.	Est-il possible de retrouver des informations sur les derniers accès d'un développeur aux objets d'un projet (date, heure, durée, modifications réalisées, etc.) ?

Tableau 1 : Eléments d'évaluation des possibilités d'un AGL en matière de partage de l'information

vail de leurs collègues et obtenir un feedback (éventuellement anonyme) concernant le leur.

Organisée sous forme de projet, la construction des systèmes d'information soulève les problèmes habituels rencontrés lors de la conduite de tout projet (planifica-

tion, gestion des délais, gestion des coûts, etc.). Compte tenu du fait que certaines tâches ne peuvent être réalisées que si une autre l'a été auparavant, que d'autres activités ne peuvent débuter que si plusieurs autres ont été précédemment accomplies ou que

d'autres encore doivent être accomplies simultanément, les outils de coordination de projets sont également susceptibles d'augmenter l'efficacité d'un AGL pour supporter le travail d'une équipe de développement. Ils sont le plus souvent fondés sur la méthode PERT (Program Evaluation Research Task) qui est une technique d'ordonnancement de projet visant à minimiser sa durée d'exécution. Grâce à l'utilisation de tels outils, des gains en termes de productivité des équipes de développement peuvent donc être envisagés.

Les outils de collaboration permettent quotidiennement aux membres d'une équipe de développement d'interagir entre eux. La salle de réunion électronique (parfois appelée salle de décision électronique) constitue un premier outil susceptible de faciliter la collaboration des développeurs. Elle leur donne la possibilité d'interagir grâce à des postes de travail en réseau pour des phases de réflexion ou de votes sur des décisions qu'ils doivent prendre. Les outils d'édition conjointe, proprement dits, permettent, de manière asynchrone, aux différents développeurs de collaborer en rédigeant par exemple la documentation du développement ou, plus sûrement, en élaborant ensemble des modèles de données ou de traitements ou encore des modules de programmation. Grâce à un outil de ce type, chacun des protagonistes a la possibilité d'enrichir l'élément édité, toutes les traces des différents apports étant conservées.

Les outils de communication. Pour ces derniers, nous désignons ici principalement la messagerie électronique (ou E-mail) qui cons-

titue l'application principale des réseaux de postes de travail. En fait, elle est considérée comme le niveau minimal du Groupware et permet de communiquer au travers de l'envoi de messages entre postes de travail. Elle est encore plus efficace lorsqu'elle est agrémentée de possibilités supplémentaires telles que les listes de diffusion (envoi à un ensemble de personnes), les fichiers joints (envoi avec le message de fichiers d'un type autre que "texte"), le chiffrement (pour coder et donc assurer la confidentialité des messages) ou encore la recherche assistée (pour retrouver des messages sélectivement). On ajoutera que, pour communiquer, les membres de l'équipe peuvent aussi utiliser le forum de discussion électronique qui présente l'avantage d'autoriser une communication simultanée à plusieurs individus par un système de questions-réponses dans lequel chacun peut s'insérer à tout instant.

Nous avons récapitulé dans le tableau 2 de la page suivante, les différents éléments qui nous permettrons d'évaluer l'assistance fournie par un AGL dans le domaine de la coopération.

Au total, l'instrument d'évaluation que nous avons conçu comporte 17 éléments permettant d'évaluer la manière dont l'information peut être partagée et 8 éléments correspondant à l'assistance fournie pour faciliter la coopération entre les individus. Cet outil, nous l'avons utilisé pour évaluer le support que les versions réseaux de six des principaux Ateliers de Génie Logiciel du marché français offrent aux équipes de développement engagées dans la réalisation d'un logiciel de gestion. La deuxième partie de notre article est consacrée

à la présentation des principaux résultats obtenus. Il importe d'observer qu'ils peuvent être considérés comme assez encourageants.

IV. Coordination	
4-1.	Existe-t-il un système d'agenda électronique partagé ?
4-2.	Les développeurs peuvent-ils suivre le travail de leurs collègues ?
4-3.	Les développeurs peuvent-ils obtenir un feedback, éventuellement anonyme, concernant leurs activités personnelles ?
4-4.	L'AGL inclut-il un outil de coordination des projets ?
V. Collaboration	
5-1.	Les membres de l'équipe ont-ils la possibilité de se concerter et de prendre des décisions à travers une salle de décision électronique ?
5-2.	Plusieurs développeurs ont-ils la possibilité d'éditer conjointement des documents ou des modèles ?
VI. Communication	
6-1.	L'AGL a-t-il des possibilités en matière de messagerie électronique ?
6-2.	Les développeurs disposent-ils d'un forum de discussion électronique pour communiquer de manière synchrone sur leur travail ?

Tableau 2 : Eléments d'évaluation des possibilités de l'AGL en matière de coopération

2. LE TEST DE L'INSTRUMENT D'ÉVALUATION : DES RÉSULTATS PROMETTEURS

Afin de tester notre outil de mesure et de déterminer les possibilités actuelles des AGL dans le domaine de l'assistance au travail des équipes de développement, les versions réseaux les plus récentes de onze des Ateliers de Génie Logiciel les plus répandus dans les organisations françaises (selon l'ADELI, association française de génie logiciel. Voir Adeli, 1997) ont été évaluées par leur fournisseur grâce à un questionnaire que nous avons fourni. Le tableau 3 ci-contre liste ces onze outils dont les principales caractéristiques sont par ailleurs décrites en annexe. Il est important de noter

que nous avons cherché à évaluer les efforts consentis par les fabricants d'Ateliers de Génie Logiciel pour prendre en compte les nouvelles pratiques en matière de développement de logiciels, à savoir le recours à des équipes de développeurs travaillant en parallèle et non plus en séquentiel dès lors que les projets sont de grande envergure. Il ne s'agissait donc pas d'évaluer la qualité du travail permis par tel ou tel AGL ou encore son ergonomie. Cette position explique par ailleurs que nous n'ayons pas tenté de tester "en laboratoire" chacun des outils sur des projets réels bien que nous ayons conscience du léger biais que peut induire le fait que ce sont les fournisseurs qui ont évalué eux-mêmes leurs outils (notamment pour des raisons commerciales).

	Fournisseur	Version concernée
AGL-X	Groupe PRISME	4.2
GRAPHTALK Designer	CSC Financial Services Group	3.0
MEGA	MEGA International	4.2
Paradigm Plus	PLATINUM Technology	3.52
PowerAMC	SYBASE	6.0
OBJECTPOOL	SAPIENS	3
SELECT Component Factory	Microdata Soft	5.0
SILVERRUN	SILVERRUN Technologies France	2.5
Tramis Builder	CONCIS Technologies	4.5
Visual Age Pachbase	CGI	1.6
Win'Design	CECIMA	2.2

Tableau 3 : AGL étudiés dans le cadre de l'article

Dans le premier développement, nous allons présenter et analyser les principaux résultats que nous avons obtenus en utilisant l'instrument de mesure que nous avons mis au point. Dans un deuxième, nous tenterons d'évaluer l'évolution des Ateliers de Génie Logiciel en ce qui concerne l'intégration des fonctionnalités de Groupware, notamment en comparant ces différents résultats avec ceux présentés dans une étude américaine réalisée en 1995. Deux éléments essentiels peuvent être

dégagés que nous présenterons successivement.

2.1. Une prédominance des outils de gestion de la mémoire de groupe par rapport aux outils de coopération

Le tableau 4 présente les résultats de notre évaluation. Ils correspondent au ratio moyen d'outils de chaque catégorie qui sont

intégrés aux Ateliers de Génie Logiciel. Son examen permet de constater, qu'en moyenne, 72 % environ des fonctionnalités de gestion de la mémoire de groupe

sont implantés dans les AGL étudiés contre 34 % seulement des fonctionnalités facilitant la coopération entre les membres des équipes de développement.

	<i>% moyen de fonctionnalités intégrées par AGL</i>
Outils de gestion des droits d'accès	60,61 %
Outils de contrôle d'intégrité	75,00 %
Outils de surveillance	84,00 %
Outils de gestion de la mémoire de groupe	71,86 %
Outils de coordination	40,91 %
Outils de collaboration	40,91 %
Outils de communication	13,64 %
Outils de coopération	34,09 %
Total	59,64 %

Tableau 4 : Résultats de l'évaluation des fonctionnalités "Groupware" intégrées au sein des AGL

D'un point de vue global, on constate que les onze produits étudiés intègrent en moyenne 60 % des 25 caractéristiques proposées. Notons à ce sujet que le nombre de fonctionnalités comprises par AGL varie de 9 (pour Win'Design) à 23 (pour Object-pool). L'analyse de la figure 6 (représentant le % de fonctionnalités retenues dans notre modèle, fournies par chaque AGL) permet d'ailleurs de constater les différences notables existant entre les différents produits analysés en ce qui concerne le nombre d'outils de chacune des deux catégories qu'ils contiennent.

Les différences sont beaucoup moins importantes pour les outils de gestion du référentiel central (cela va de 47 % environ pour

Win'Design à 94 % pour Object-pool) que pour les outils permettant la coopération entre les développeurs. Certains produits n'intègrent en effet aucune fonctionnalité de cette catégorie ou presque (c'est par exemple le cas de Tramis Builder, Power AMC ou Win'Design) alors, qu'en revanche, Sapiens Objectpool en contient 7 sur les 8 possibles.

Si l'on s'intéresse, ensuite, aux outils permettant le partage des informations, on constate que l'on trouve plus fréquemment des outils de surveillance (85 %) ou des outils de contrôle d'intégrité des données (75 %) que des outils de gestion des droits d'accès (61 % environ) au sein des AGL multi-utilisateurs.

Cf. figure 6

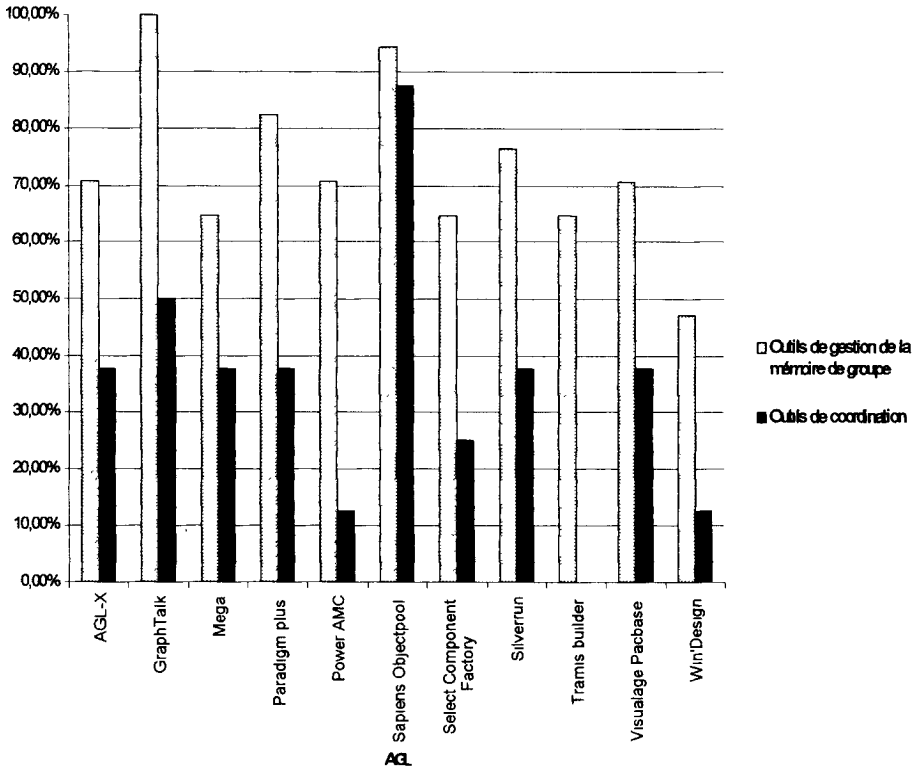


Figure 6

Résultats par AGL

La figure 7 présentée en page suivante permet de visualiser les outils de gestion de la mémoire de groupe, les plus fréquemment intégrés aux AGL. Dans le domaine des outils de gestion des droits d'accès, des fonctionnalités telles que les possibilités de définir des mots de passe différents pour des projets différents, de prévoir des mots de passe différents pour les membres d'une même équipe ou encore de définir des droits d'accès en lecture seule sont très courantes (82 %). En revanche, seul un des onze produits retenus (Sapiens Objectpool) permet de prévoir des périodes d'accès aux

objets, limitées dans le temps pour certains développeurs.

Si nous nous intéressons aux outils de contrôle d'intégrité des données, nous constatons que tous les AGL que nous avons étudiés prévoient le verrouillage automatique d'un objet lorsqu'un accès simultané est tenté. Plus de 80 % d'entre eux permettent d'empêcher la modification ou la suppression d'un objet et prévoient la recherche automatique des noms d'objets identiques ou proches. En revanche, moins de la moitié des produits (45 %) impose l'utilisation d'une terminologie standard pour la création des objets.

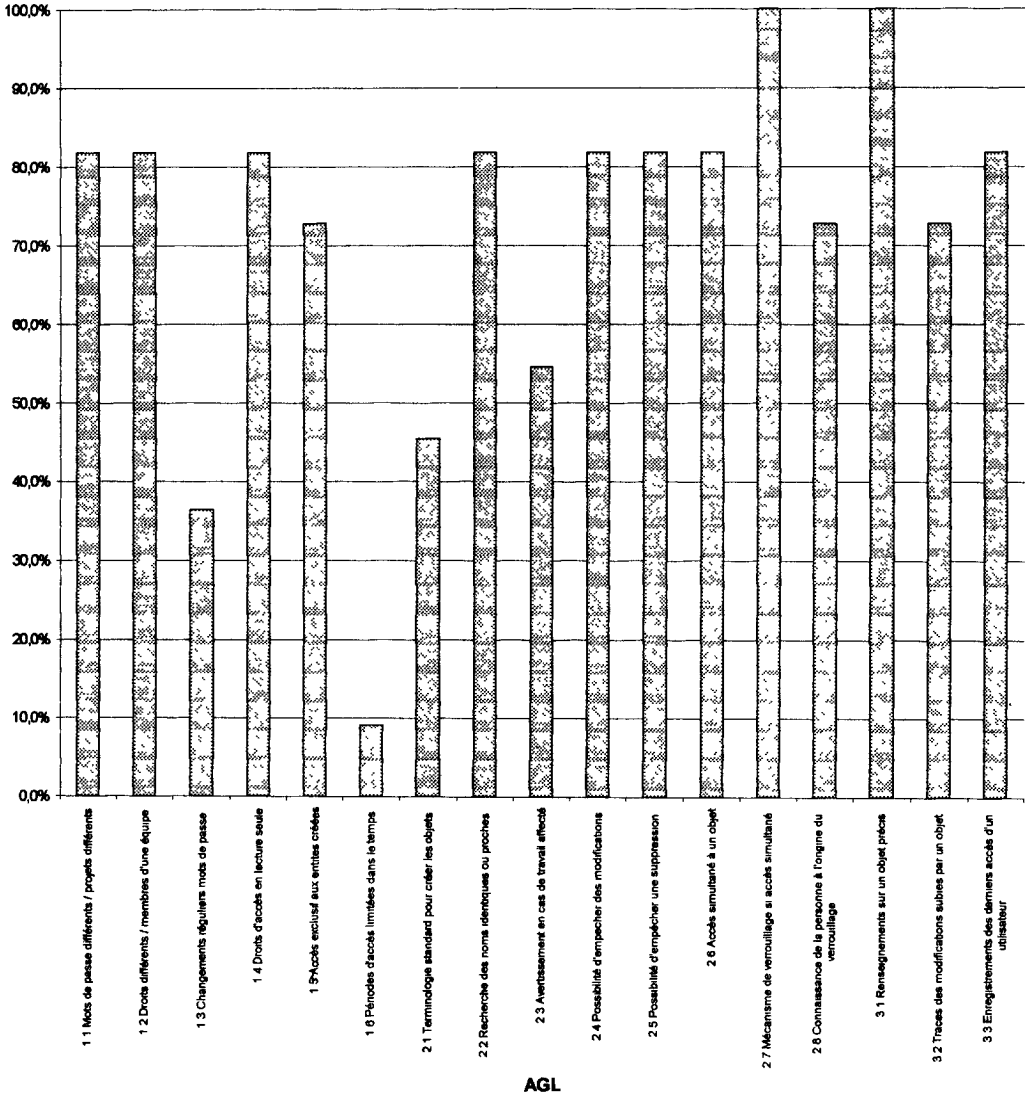


Figure 7

Fréquence d'intégration des fonctionnalités de gestion de la mémoire de groupe dans les AGL

Ainsi que le montre la figure 7, les outils de "surveillance", que ce soit de l'évolution du produit ou du travail des développeurs, sont très souvent intégrés aux AGL multi-utilisateurs. Ils sont en effet tous présents dans huit AGL sur onze.

Ainsi que nous l'avons déjà observé, on trouve actuellement dans les Ateliers de Génie Logiciel, infiniment plus d'outils visant à faciliter le partage des données entre les développeurs que d'outils facilitant leur coopération. Excepté Objectpool, aucun des AGL examinés ne dispose de plus de 50 % des fonctionnalités proposées dans notre modèle au plan de la coopération entre les développeurs d'une équipe. Notons d'ailleurs que quatre d'entre eux ne dépassent même pas 25 % de cet ensemble (Power AMC, Select Component factory, Tramis Builder et Win'Design).

De plus, s'ils intègrent parfois des outils de coordination et de collaboration (voir tableau 4), notre étude montre également que les Ateliers de Génie Logiciel actuels ne disposent quasiment jamais d'outils de communication (deux sur onze seulement).

Ajoutons, enfin, que les fonctionnalités de coopération les plus répandues sont "le suivi de travail" et l'édition conjointe puisqu'on les rencontre dans plus de 70 % des AGL. En revanche, ainsi que la figure 8 permet de le visualiser, certains outils restent encore très marginaux au sein de cette technologie puisqu'on ne les trouve (au

maximum) que dans deux produits sur onze. Il s'agit pourtant d'outils de Groupware très classiques, tels que la messagerie électronique, la salle de décision électronique ou encore le forum de discussion.

En définitive, les résultats les plus significatifs obtenus dans cette étude mettent en évidence la très grande diversité des produits que l'on peut trouver sous la dénomination générique d'Atelier de Génie Logiciel multi-utilisateurs. Il est certes difficile de fixer un seuil (c'est-à-dire un pourcentage de fonctionnalités de Groupware intégrées) au-delà duquel on peut considérer qu'un AGL est un véritable "outil de support collaboratif" en regard du modèle d'architecture conceptuel que nous avons défini mais, compte tenu du nombre assez réduit de fonctionnalités que certains des outils examinés contiennent, il nous paraît difficile de les classer dans cette catégorie. Il en est ainsi pour Win'Design et Tramis Builder par exemple qui n'intègrent respectivement que 36 % et 44 % des fonctionnalités. D'autres, en revanche, disposent de toutes les caractéristiques nécessaires pour être considérés comme de véritables technologies multi-utilisateurs. On citera sous cette rubrique AGL-X, GraphTalk, VisualAge Pacbase, Silverrun, Paradigm Plus et surtout Objectpool dont les scores sont d'au moins 60 % d'outils de support des activités des équipes de développement intégrés à la technologie.

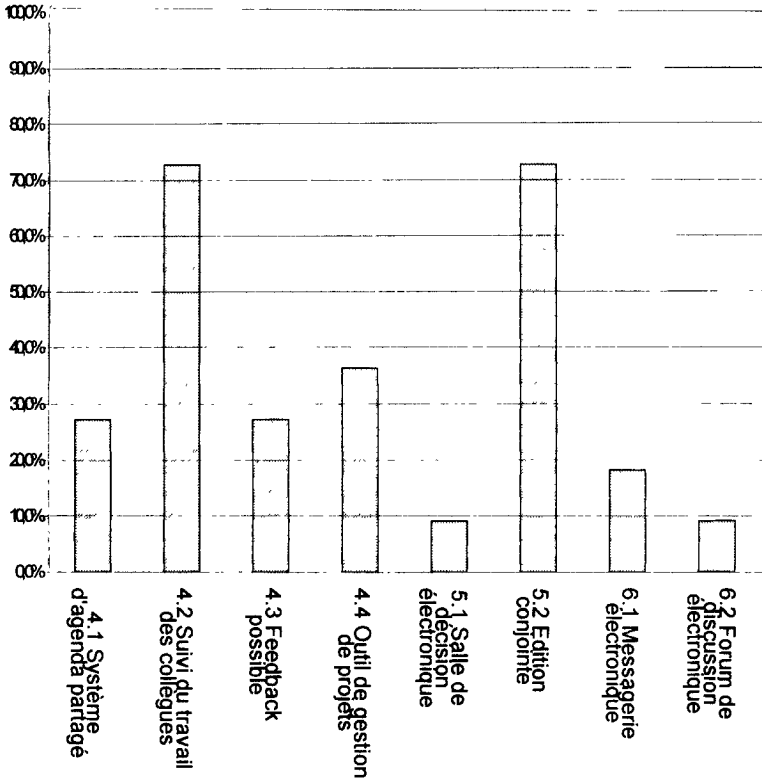


Figure 8

Fréquence d'intégration des fonctionnalités de coopération au sein des AGL

Il reste que si la plupart des produits de génie logiciel examinés peuvent encore être améliorés quant au soutien qu'ils apportent pour supporter les travaux des équipes de développement, il n'en demeure pas moins que l'évolution dans ce domaine est importante ainsi que le démontre l'analyse comparative réalisée entre notre étude et une étude américaine de 1995, présentée dans le paragraphe qui suit.

2.2. Une évolution sensible vers de véritables environnements de développement multi-utilisateurs

En 1995, deux chercheurs américains (Vessey et Sravanapudi) ont également défini un outil de mesure de l'efficacité des AGL pour favoriser les activités de groupe. Même si cet instrument

ne s'intéresse pas exactement aux mêmes fonctionnalités que le nôtre, il les réunit en deux classes comparables à celles que nous avons proposées puisque les deux auteurs ont regroupé, d'une part, les outils de gestion de la mémoire de groupe dans une catégorie dénommée "Teamware" et d'autre part, les outils favorisant la coopération, réunis sous la dénomination de "Groupware". Le tableau 5 qui suit, permet de com-

parer les 2 instruments de mesure au niveau du nombre de fonctionnalités de chaque catégorie. Il montre essentiellement que l'outil de mesure proposé favorise les fonctionnalités de gestion de la mémoire de groupe par rapport à celles concernant la coopération. Pour cette dernière raison, nous n'avons pas considéré pour nos comparaisons, de sous-classe dans la catégorie des outils de coopération.

	<i>Instrument de Vessey et Sravanapudi</i>	<i>Notre instrument</i>
Gestion des droits d'accès	7	6
Contrôle d'intégrité des données	17	8
Surveillance (monitoring)	8	3
Gestion de la mémoire de groupe	32	17
Coopération	3	8
Total	35	25

Tableau 5 : répartition des fonctionnalités retenues dans chaque instrument de mesure

Les deux auteurs américains ont réalisé leurs mesures sur les versions réseaux de seulement quatre outils présents sur le marché américain (DEFT 4.0, ICONIX 4.0, System Architect 2 et Visible Analyst 2.0). A ce titre et ainsi qu'ils le soulignent eux-mêmes, leurs résultats sont sans doute moins significatifs que ne peuvent l'être ceux de notre étude. Cependant, ils nous permettent de tirer quelques enseignements intéressants s'agissant de l'évolution depuis 2 ans, du support fourni par les AGL au niveau des activités des équipes de développement de logiciels.

La lecture du tableau 6 nous permet tout d'abord de constater, qu'en moyenne, les AGL de la génération actuelle contiennent beaucoup plus d'outils destinés à

favoriser les activités d'équipe que ceux qui étaient disponibles, il y a seulement deux ans. On observe en effet que les produits que nous avons examinés intègrent en moyenne 60 % des fonctionnalités décrites dans notre modèle contre moins de 40 % en 1995. Ensuite, il semble que les fabricants d'AGL ont intégré en priorité à leurs produits des outils permettant la surveillance tout à la fois de l'évolution du logiciel fabriqué et des activités des développeurs (+ 60 points environ). Enfin, on remarque que les nouveaux Ateliers de Génie Logiciel intègrent plus d'outils facilitant la coopération entre les individus (+ 17 points) même si ce sont toujours ceux qui permettent de gérer la mémoire de groupe qui sont les plus nombreux.

	<i>Etude de 1995</i>	<i>Etude de 1997</i>
Outils de gestion des droits d'accès	28,57 %	60,61 %
Outils de contrôle d'intégrité des données	54,41 %	75,00 %
Outils de surveillance	25,00 %	84,85 %
Outils de gestion de la mémoire de groupe	32,81 %	71,66 %
Outils de coopération	16,70 %	34,09 %
Total	39,30 %	59,64 %

Tableau 6 : Pourcentage moyen de fonctionnalités de chaque classe intégrées à un AGL

En définitive, cette analyse comparative des résultats de deux études réalisées à deux ans d'intervalle, démontre une évolution des orientations des fournisseurs d'Ateliers de Génie Logiciel vers la production d'outils plus efficaces pour supporter le travail des équipes de développement. Les nouveaux produits de génie logiciel intègrent en effet de plus en plus d'outils permettant aux développeurs de travailler facilement ensemble et donc de communiquer. On notera néanmoins qu'il s'agit toujours beaucoup plus d'une communication implicite, à travers les informations liées au logiciel fabriqué, qu'une véritable communication explicite visant à résoudre des problèmes particuliers. Nonobstant, il est certain que l'on trouve de plus en plus de fonctionnalités de coopération au sein des AGL.

CONCLUSION

Dans le cadre de cet article, nous avons proposé un modèle d'outil de support collaboratif, composé de trois niveaux (outils de développement, outils de gestion de la mémoire de groupe, ou-

tils de coopération). En faisant référence à ce modèle, nous avons défini un instrument de mesure de l'efficacité d'un Atelier de Génie Logiciel pour supporter les activités d'une équipe de développement de logiciels. Cet instrument a été utilisé pour évaluer les versions réseaux de onze produits présents sur le marché français. La dimension exploratoire de cette étude nous interdit toute conclusion définitive. Elle montre cependant que la technologie AGL est en train d'évoluer puisque l'on trouve de plus en plus de produits sur le marché permettant de soutenir efficacement le travail d'une équipe de développement de logiciels en optimisant les contrôles et les échanges entre concepteurs. Elle suggère toutefois que, pour que de telles technologies puissent être considérées comme de véritables environnements de développement multi-utilisateurs, il est nécessaire d'aller encore plus loin que la plupart des fournisseurs ne l'ont fait jusqu'à présent, en intégrant non seulement des outils favorisant le partage des informations mais aussi des outils permettant aux membres d'une équipe de développement de coopérer efficacement. Elle conduit évidemment à un

certain nombre de nouvelles pistes de recherche complémentaires, d'autant qu'il n'existe qu'un nombre très réduit de travaux sur la manière dont les individus travaillent ensemble au sein des équipes de développement de logiciels (voir éventuellement, Bendifallah et Scacchi, 1989 ; Mantel, 1981). Il serait certainement intéressant d'étudier précisément comment et quand les développeurs communiquent au cours du processus de développement afin de faire évoluer en ce sens les Ateliers de Génie Logiciel et les composants qu'ils intègrent. Dans un autre domaine, une étude qui traiterait des effets sur la productivité de l'utilisation d'AGL supportant les travaux de groupe semble être également un problème de recherche intéressant pour l'avenir. Comme on peut le constater, l'utilisation des Ateliers de Génie Logiciel reste un domaine particulièrement fécond pour les chercheurs en sciences de gestion.

BIBLIOGRAPHIE

- Adeli, Direction Paul Theron (1990), *Génie logiciel et productivité des études*, Afnor technique, Paris-la-Défense.
- Adeli, (1991), *Les ateliers de conception. Comprendre, comparer, mettre en œuvre (tome 3)*, Publications ADELI, Paris.
- Adeli, (1992), *Les ateliers de conception. Comprendre, comparer, mettre en œuvre (Tome 4)*, Publications ADELI, Paris.
- Adeli, (1997), *AGLoscope 97*, Les éditions de l'ADELI, Paris.
- Alavi, M. (1993), « Making CASE an Organizational Reality », *Information Systems Management*, Vol. 10, n° 2, p. 15-20, Spring.
- Bartram, P., Youett, C. (1993), *Groupware report '93*, Business Intelligence.
- Bénassy, J. (1991), « Du dictionnaire de données au référentiel », *L'informatique professionnelle*, n° 93, Avril.
- Bendifallah, S. Scacchi, W. (1989), « Work Structures and Shifts : an Empirical Analysis of Software Specification Teamwork », *Conference on Computer Human Interaction*, p. 260-270.
- Bikson, T.K., Eveland, J.D., Gutek, B.A. (1989), « Flexible Interactive Technologies for Multi-person Tasks : Current Problems and Future Prospects », *Technological Support for Workgroup Collaboration*, M.H. Olson (Ed.), Lawrence Erlbaum Associates, p. 89-112.
- Brès, J. (1994), *Ateliers de Génie Logiciel - Réalités et tendances*, Masson, Paris.
- Bostrom, R., Anson, R. (1988), « Using Computerized Collaborative Work Support Systems to improve the Logical Systems Design Process », *Proceedings of the SIGCPR Conference*, ACM press, New York, p. 88-108.
- Charrette, (1986), *Software Engineering Environments Concepts and Technology*, McGraw-Hill, New York.
- Coleman, D. (1992), *Groupware '92 Conference Proceedings*, Morgan Kaufmann Publishers.
- Coleman, D. (1993), *Groupware '93 Conference Proceedings*, The Conference Group.
- Coleman, D., Khanna, R. éditeurs (1995), *Groupware : Technology and Applications*, Prentice Hall.
- Courbon, J.-C., Tajan, S. (1997), *Groupware et Intranet - Application avec Notes et Domino*, Masson, Paris.
- Crosslin, R.L., Bergin, Thomas, J., Stott, Jack, W. (1993), « Critical Factors Influencing the Future of Computer-Aided Software Engineering » in *Computer-Aided Software Engineering. Issues and Trends for the 1990s and Beyond*, Thomas J. Bergin (ed.), Idea Group Publishing, Harrisburg (USA) et Londres (UK), p. 616-637.
- De Marco, T., Lister, T. (1987), *Peopleware*, Dorset House, New York.
- De Sanctis, G., Gallupe, B. (1987), « A Foundation for the Study of

Group Decision Making », *Management Science*, Vol. 33, n° 5, mai.

De Terssac, G. (1996), « Le travail de conception : de quoi parle-t-on ? » dans *Coopération et conception*, Gilbert De Terssac et Erhard Friedberg (eds.), Octares éditions, Toulouse, p. 1-22.

Dreger, B.J. (1989), *Function Point Analysis*, Prentice-Hall.

Grochow, J.M. (1988), « Designer Workstations in a multi-person environment », *Advance Papers, Second International Workshop on Computer-Aided Software Engineering*, Vol. 2, E. Chikofsky (Ed.), Index Technology Corporation, Cambridge mass.

Grudin, J. (1994), « Computer-Supported Cooperative Work : History and Focus », *IEEE Computer*, p. 19-26, May.

Habrias, H. (1994), *La mesure du logiciel*, Teknea, Toulouse.

Hatchuel, A. (1994), « Apprentissages collectifs et activités de conception », *Revue française de gestion*, n° 99, p. 109-120.

Hatchuel, A. (1996), « Coopération et conception collective. Variété et crises des rapports de prescription », dans *Coopération et conception*, Gilbert De Terssac et Erhard Friedberg (eds.), Octares éditions, Toulouse, p. 101-121.

Hayne, S., Ram, S. (1995), « Group Data Base Design : Addressing the View Modeling Problem », *Journal of Systems and Software*, Vol. 28, n° 2, p. 97-116, February.

Jarvenpaa, S.L., Rao, V.S., Huber, G.P. (1988), « Computer Support for Meetings of Groups Working on Unstructured Problems : a Field Experiment », *MIS Quarterly*, Vol. 4, n° 2, p. 645-666.

Johansen, R. (1988), *Groupware : Computer Support for Business Teams*, The Free Press, McMillan, New York.

Jones, T.C., (1986), *Programming Productivity*, McGraw-Hill, New York.

Keyes, J. (1992), « Team Tools Targeting Development Process », *Software Magazine*, Vol. 12, n° 15, p. 45-56, November.

Korth, Silberschatz (1987), *Database System Concepts*, McGraw-Hill, New York.

Kozar, K, Zigurs, I. (1992), « Human and machine Roles in Team Product Reviews. A Prescription for Change », *Information and Management*, Vol. 23, p. 149-157.

Levan, S.K., Liebman, A. (1994), *Le Groupware - Informatique, management et organisation*, Hermes, Paris.

Lévine, P., Pomerol, J.-C. (1990), *Systèmes interactifs d'aide à la décision et systèmes experts*, Hermes, Paris.

Liou, Y.I., Chen, M. (1994), « Using Group Support Systems and Joint Application Development for Requirements Specification », *Journal of Management Information Systems*, Vol. 10, n° 3, p. 25-41, Winter.

Mace, S. (1993), « Knowledgeware toolset to offer team development features », *Infoworld*, Vol. 15, n° 42, p. 30, October 18.

Malone, T.W., Lai, K., Fry, C. (1992), « Experiments with OVAL : a Radically Tailorable Tool for Cooperative Work », *Proceedings of the ACM Conference on CSCW*, Toronto, November.

Mantel, M. (1981), « The Effect of Programming Team Structures on Programming Tasks », *Communications of the ACM*, Vol. 24, n° 3, p. 106-113, March.

Midler, C. (1996), « Modèles gestionnaires et régulations économiques de la conception », dans *Coopération et conception*, Gilbert De Terssac et Erhard Friedberg (eds.), Octares éditions, Toulouse, p. 63-85.

Mintzberg, Rainsinghani, Theoret (1976), « The Structure of Unstructured Decision Process », *Administrative Science Quarterly*, Vol. 21, June.

Paoli, F.D., Tisato, F. (1994), « CSDL : a Language for Cooperative Systems Design », *IEEE transactions on Software Engineering*, Vol. 8, n° 20, August.

Reinhard, W., Schweitzer, J., Volksten, G. (1994), « CSCW Tools : Concepts and Architectures », *IEEE Computer*, p. 28-36, May.

Reix, R. (1995), *Systèmes d'information et management des organisations*, Vuibert, Paris, Septembre.

Sellars, P. (1987), « IPSEs in Support of Team » in *Automating systems Development*, D. Benyon, S. Skidmore (eds.), Plenum Press, New York.

Simon, H. (1983), *Administration et processus de décision*, Economica, Paris.

Varinard, C. (1997), *Impact de l'utilisation des Ateliers de Génie Logiciel en informatique de gestion (étude*

exploratoire), Thèse, Université Montpellier II, Mai.

Varinard, C., Reix, R. (1997), « Une tentative d'identification des facteurs clés du succès de l'introduction d'un Atelier de Génie Logiciel dans une organisation », *Systèmes d'Information et management*, Vol. 2, n° 3.

Vessey, I., Sravanapudi, A.P. (1995), « CASE tools as Collaborative Support Technologies », *Communications of the ACM*, Vol. 38, n° 1, p. 83-95, January.

Cf. ANNEXE

ANNEXE

Tableaux synoptiques décrivant
les dix Ateliers de génie Logiciel étudiés

1. Informations générales

	Environnement de travail utilisateur	Nombre de licences (France)	Nombre de licences (Europe)	Passage en réalisation Données	Passage en réalisation Traitements
AGL-X	Windows / AS400	90	90	Intégré	Intégré
GRAPHTALK DESIGNER	Windows	1 500	2 000	Intégré	Intégré
MEGA	Windows / OS2	10 000	10 500	Intégré	Interfacé
OBJECTPOOL	Windows / UNIX / AS400	20	150	Intégré	Interfacé
PARADIGM PLUS	Windows / OS2 / UNIX	1 000	Non communiqué	Intégré	Interfacé
POWER AMC	Windows	7 500	9 000	Intégré	Intégré
SELECT COMPONENT FACTORY	Windows	800	18 000	Intégré	Intégré
SILVERRUN	Windows / OS2 / UNIX / Macintosh	33 000	15 000	Intégré	Intégré
TRAMIS BUILDER	Windows	1 650	1 650	Intégré	Intégré
VISUALAGE PACBASE	Windows / OS2	1 750	2 250	Intégré	Intégré
WIN'DESIGN	Windows	200	Non communiqué	Intégré	Interfacé

2. Méthodes de modélisation supportées

	Méthodes structurées	Méthodes objet
AGL-X	MERISE / SA	
GRAPHTALK DESIGNER		OMT / UML
MEGA	MERISE / IE / MEGA	UML / OOSE
OBJECTPOOL	Object Modeler	Object Modeler
PARADIGM PLUS		OMT / OOA / OOAD / OOM / UML / OOCL
POWER AMC	MERISE	OMT / Gane & Sarsson / Yourdon & De Marco / SSADM
SELECT COMPONENT FACTORY		OMT / UML / Business Process Modeler
SILVERRUN	MERISE / IE	OMT / Classe-Relation / UML
TRAMIS BUILDER	MERISE / IE / SA	OMT / OOM / UML / Méthode des objets naturels
VISUALAGE PACBASE	MERISE / YSM	WSDDM - BIRD
WIN'DESIGN	MERISE	

3. Fonctionnalités particulières

	Documentation générée	Norme du référentiel	Réutilisation	Rétro-conception des données	Rétro-conception des traitements
AGL-X	Oui / Format RTF	IRDS	Oui	Oui	Non
GRAPHTALK DESIGNER	Oui / Format RTF	Système de gestion objets	Oui	Oui	Non
MEGA	Oui / WORD / WORDPERFECT	IRDS	Oui	Oui	Non
OBJECTPOOL	Oui / WORD	AD / CYCLE	Oui	Oui	Non
PARADIGM PLUS	Oui / WORD	Non communiquée	Oui	Oui	Oui
POWER AMC	Oui / WORD / WORDPERFECT	Non communiquée	Non	Oui	Non
SELECT COMPONENT FACTORY	Oui / WORD	Btrieve / OLE	Oui	Oui	Oui
SILVERRUN	Oui / WORD	Non communiquée	Oui	Oui	Oui
TRAMIS BUILDER	Oui / Format RTF	Non communiquée	Oui	Oui	Oui
VISUALAGE PACBASE	Oui / Propre traitement de texte	PCTE / IRDS	Oui	Oui	Oui
WIN'DESIGN	Oui / Format RTF	Système propriétaire	Non	Oui	Non