

Quand les clients organisent la coopération/coordination entre entreprises innovantes : le cas de l'industrie logicielle.

François Scheid.

EM LYON,

23 avenue Guy de Collongue, 69132 Ecully Cedex, France,
+33 4 78 33 77 78, Fax: +33 4 78 33 78 84, scheid@em-lyon.com

&

Florence Charue-Duboc

Ecole Polytechnique-CRG,

1 rue Descartes, 75005 Paris, France.

- mots clés : développement de produit nouveau ; innovation radicale ; lead user ; modularité; logiciel.

Résumé :

La prise en compte des besoins des clients a été soulignée comme un facteur clé de succès des processus d'innovation dès les travaux de Freeman (1982). Von Hippel et Katz (2002) considèrent même que les clients jouent un rôle déterminant dans l'émergence et l'orientation de l'innovation. Dans l'innovation logicielle, la prise en compte des besoins des utilisateurs apparaît particulièrement critique (Cusumano 2004). Fichman et Kemerer (1997) suggèrent d'impliquer les utilisateurs dans les phases amont de conception et de développement de ces innovations afin de prendre en compte leurs besoins. Dans quelle mesure le concepteur de l'innovation doit-il satisfaire ces besoins alors qu'il cherche à développer un produit susceptible d'intéresser un nombre significatif de clients ? Nous nous appuyons principalement sur les travaux relatifs à la modularité et à l'innovation par les utilisateurs. Nous analysons la littérature portant sur la modularité qui souligne que les architectures modulaires favorisent l'innovation. Nous présentons ensuite les travaux sur l'innovation par les utilisateurs et discutons leur pertinence dans le cas de l'innovation logicielle radicale.

Nous étudions le cas d'une petite entreprise innovante, SoftCo. Elle édite des progiciels de *text mining*, qui permettent l'extraction d'informations contenues dans un corpus de textes (articles de journaux, textes de lois, brevets...). Nous nous focalisons sur une succession de projets visant à déployer des applications mobilisant une même brique logicielle innovante chez différents clients. Nous comparons l'offre innovante et l'organisation de ces projets, pour repérer des effets d'apprentissage et d'irréversibilité créés par les premiers produits conçus pour les premiers clients par rapport aux produits de la même famille conçus pour les clients suivants. Nous avons suivi de manière longitudinale deux projets pour deux grandes entreprises (groupe de presse et éditeur), basés sur des logiciels d'extraction, de catégorisation et d'archivage d'information. Ces logiciels sont développés principalement par SoftCo et KnowCo (éditeur d'un système de gestion de base de connaissance). Nous avons interrogé les employés de SoftCo qui sont en relation avec le client final dans le cadre d'entretiens semi-

directifs, ainsi que les responsables de projet chez KnowCo et chez les clients, en tout trente-et-un entretiens, entre juin 2004 et juin 2006 :

Notre recherche met en évidence un *lead-user* dont la compétence est d'abord sur les usages et dont la contribution à la conception de logiciel innovant porte sur l'architecture d'une offre complète. L'architecture est conçue par le premier client pour ses besoins spécifiques, et elle est ensuite reprise par les entreprises de logiciel dans leur proposition au second client. Le second domaine que notre travail éclaire a trait à l'élaboration des *design rules* qui permettent une modularisation des systèmes. Dans le cas des ordinateurs, ces *design rules* ont été constituées *ex-ante* par un acteur dominant du secteur. Notre recherche met en évidence un autre processus de construction des *design rules* (Baldwin et Clark, 1997). En effet, les *design rules* émergent ici progressivement lorsque plusieurs éditeurs de logiciel coopèrent pour construire une offre globale satisfaisant les besoins d'un client.

Quand les clients organisent la coopération/coordination entre entreprises innovantes : le cas de l'industrie logicielle.

Résumé :

La prise en compte des besoins des clients a été soulignée comme un facteur clé de succès des processus d'innovation dès les travaux de Freeman (1982). Von Hippel et Katz (2002) considèrent même que les clients jouent un rôle déterminant dans l'émergence et l'orientation de l'innovation. Dans l'innovation logicielle, la prise en compte des besoins des utilisateurs apparaît particulièrement critique (Cusumano 2004). Fichman et Kemerer (1997) suggèrent d'impliquer les utilisateurs dans les phases amont de conception et de développement de ces innovations afin de prendre en compte leurs besoins. Dans quelle mesure le concepteur de l'innovation doit-il satisfaire ces besoins alors qu'il cherche à développer un produit susceptible d'intéresser un nombre significatif de clients ? Nous nous appuyons principalement sur les travaux relatifs à la modularité et à l'innovation par les utilisateurs. Nous analysons la littérature portant sur la modularité qui souligne que les architectures modulaires favorisent l'innovation. Nous présentons ensuite les travaux sur l'innovation par les utilisateurs et discutons leur pertinence dans le cas de l'innovation logicielle radicale.

Nous étudions le cas d'une petite entreprise innovante, SoftCo. Elle édite des progiciels de *text mining*, qui permettent l'extraction d'informations contenues dans un corpus de textes (articles de journaux, textes de lois, brevets...). Nous nous focalisons sur une succession de projets visant à déployer des applications mobilisant une même brique logicielle innovante chez différents clients. Nous comparons l'offre innovante et l'organisation de ces projets, pour repérer des effets d'apprentissage et d'irréversibilité créés par les premiers produits conçus pour les premiers clients par rapport aux produits de la même famille conçus pour les clients suivants. Nous avons suivi de manière longitudinale deux projets pour deux grandes entreprises (groupe de presse et éditeur), basés sur des logiciels d'extraction, de catégorisation et d'archivage d'information. Ces logiciels sont développés principalement par SoftCo et KnowCo (éditeur d'un système de gestion de base de connaissance). Nous avons interrogé les employés de SoftCo qui sont en relation avec le client final dans le cadre d'entretiens semi-directifs, ainsi que les responsables de projet chez KnowCo et chez les clients, en tout trente-et-un entretiens, entre juin 2004 et juin 2006.

Notre recherche met en évidence un *lead-user* dont la compétence est d'abord sur les usages et dont la contribution à la conception de logiciel innovant porte sur l'architecture d'une offre complète. L'architecture est conçue par le premier client pour ses besoins spécifiques, et elle est ensuite reprise par les entreprises de logiciel dans leur proposition au second client. Le second domaine que notre travail éclaire a trait à l'élaboration des *design rules* qui permettent une modularisation des systèmes. Dans le cas des ordinateurs, ces *design rules* ont été constituées *ex-ante* par un acteur dominant du secteur. Notre recherche met en évidence un autre processus de construction des *design rules* (Baldwin et Clark, 1997). En effet, les *design rules* émergent ici progressivement lorsque plusieurs éditeurs de logiciel coopèrent pour construire une offre globale satisfaisant les besoins d'un client.

I Introduction :

Notre recherche porte sur la relation fournisseur/client dans le développement d'innovations de rupture dans le logiciel. La prise en compte des besoins des clients a été soulignée comme un facteur clé de succès des processus d'innovation dès les travaux de Freeman (1982). Von Hippel et Katz (2002) considèrent même que les clients jouent un rôle déterminant dans l'émergence et l'orientation de l'innovation. Plus spécifiquement, dans l'innovation logicielle, c'est la prise en compte des besoins des utilisateurs qui apparaît critique (Cusumano 2004). Fichman et Kemerer (1997) suggèrent d'impliquer les utilisateurs dans les phases amont de conception, développement de ces innovations afin de prendre en compte leurs besoins. Cependant les premiers clients risquent d'orienter le processus d'innovation vers leurs besoins propres qui peuvent être très spécifiques. Dans quelle mesure le concepteur de l'innovation doit-il satisfaire ces besoins alors qu'il cherche à développer un produit susceptible d'intéresser un nombre significatif de clients ? Telle est la question sur laquelle porte notre recherche. Nous avons choisi pour apporter des éléments de réponse à ces questions d'étudier le cas d'une entreprise dans le domaine du logiciel destiné aux entreprises. Nous caractérisons les modalités d'une intégration des premiers clients, dans le processus d'innovation. Nous montrons les implications de cette intégration sur l'organisation du projet d'innovation et sur l'architecture de l'innovation elle-même.

II Revue de littérature et choix du cadre théorique :

Nous nous appuyerons d'une part sur les travaux de Von Hippel (1986) qui ayant mis en évidence le rôle des utilisateurs dans les processus d'innovation a étudié différentes configurations permettant de les associer plus systématiquement. Nous présenterons ensuite des travaux sur la modularité qui soulignent l'apport de cette architecture pour favoriser l'innovation et pour satisfaire des besoins variés tout en bénéficiant d'économie d'échelle. Ces deux courants de travaux ne sont pas spécifiques au domaine du logiciel. Nous positionnerons enfin des analyses des processus de développement logiciel qui insistent sur la flexibilité nécessaire compte tenu de la difficulté à expliciter et anticiper les besoins des clients.

II-1. Les notions de lead user, de toolkit et de communauté d'utilisateurs :

Von Hippel (1986) suggère aux entreprises de générer des concepts d'innovation en partenariat avec des *lead users*. La notion de *lead user* est définie comme suit. Il s'agit d'utilisateurs (individus ou organisations) qui ressentent des besoins qui deviendront ultérieurement ceux d'un grand nombre d'utilisateurs, et qui espèrent retirer des bénéfices significatifs de la satisfaction de ces besoins. Comme il est généralement difficile pour un utilisateur de définir des besoins relatifs à de nouveaux produits, une approche possible consiste à sélectionner des clients *lead users* puis à leur faire tester des prototypes, pour récupérer des informations exploitables. Mais, face à l'accélération du rythme d'apparition des innovations dans chaque industrie, il peut être profitable de laisser certains utilisateurs développer eux-mêmes leur innovation, comme c'est le cas pour 80% des produits du secteur de l'instrumentation scientifique (von Hippel, 1994). Ainsi, certaines entreprises vont encore plus loin en choisissant d'équiper leurs clients d'outils permettant de concevoir et de développer eux-mêmes les produits qui leur conviennent. Von Hippel (2001) propose une approche dans laquelle l'utilisateur reçoit les outils (« user toolkit » ou boîte-à-outils de l'utilisateur) permettant de s'approprier complètement la conception du produit, l'éventuelle industrialisation restant à la charge du producteur. Le but est, ensuite, d'incorporer certaines solutions développées par ces utilisateurs *lead-users* dans des produits standard qui intéresseront ultérieurement un plus grand nombre d'utilisateurs (Thomke et von Hippel, 2002). Les boîtes-à-outils permettent aussi aux clients qui le souhaitent de développer une solution exactement adaptée à leurs besoins. L'approche *toolkit* appliquée à la création de produit innovant passe par le découpage du processus d'innovation, en sous-tâches dévolues soit à l'utilisateur soit au producteur (von Hippel et Katz, 2002). Une telle répartition des tâches peut impliquer des changements radicaux quant à l'architecture d'un produit, et entraîne généralement le développement d'une architecture modulaire, telle qu'on peut l'observer dans les logiciels libres (von Hippel et von Krogh, 2003).

Pour von Hippel (2001), il est fondamental qu'il existe une *communauté d'utilisateurs* de l'innovation, partageant librement les connaissances acquises relatives à cette innovation, ainsi que les améliorations qu'ils lui apportent (exemple du logiciel libre ou *open source*). Une *communauté d'utilisateurs* ne peut fonctionner qu'à certaines conditions, la première étant qu'un nombre significatif d'utilisateurs, suffisamment compétents soit motivés par l'innovation. La deuxième condition est que de multiples sources d'innovation puissent être activées, sources dont chaque membre de la communauté puisse bénéficier (sinon, chaque

utilisateur devrait développer la totalité de sa solution ou de ses perfectionnements). Le partage de l'innovation résulte d'incitations telles que l'accroissement de la réputation de l'innovateur et la création d'obligation envers celui qui a innové. De plus, le partage de la connaissance accroissant la diffusion de la solution peut pousser les fabricants à l'incorporer dans la future conception du produit (Harhoff *et al.*, 2000). Dans le cas d'innovations devant être incorporées dans des produits physiques (qu'il faut fabriquer et distribuer physiquement) les utilisateurs « lead-user » peuvent assurer un travail significatif de développement et de tests de prototype, mais production et diffusion restent assurées par le fabricant. Dans le cas de produits intangibles, par contre, la communauté peut selon von Hippel (2001) assurer l'ensemble des fonctions des processus d'innovation, et l'innovation développée par l'utilisateur devrait être en concurrence avec l'innovation des firmes commerciales.

La question des compétences requises pour participer à la conception d'une innovation et donc être lead user à part entière est peu abordée par ces travaux. N'est ce pas déterminant du rôle que peuvent tenir les utilisateurs dans des contextes technologiques variés et dans les innovations de rupture technologique ?

II-2. La modularité au cœur de la littérature sur l'architecture informatique :

La *modularité* est au centre des travaux de recherche sur les processus d'innovation dans le domaine informatique. « Différentes sociétés peuvent concevoir et produire indépendamment des composants tels que des lecteurs de disques durs ou des systèmes d'exploitation, qui sont autant de modules qui fonctionneront ensemble dans un produit complexe, car les concepteurs de ces modules suivent un ensemble de règles de développement, *design rules*. » (Baldwin & Clark, 1997). La *modularisation* est la décomposition d'un système complexe en sous-systèmes quasi autonomes, pouvant être conçus de manière indépendante (Baldwin et Clark, 1997 ; Aoki, 2002). Le problème est alors d'élaborer un système complexe en intégrant ces sous-systèmes. La *modularisation* peut ainsi être vue comme une stratégie de spécialisation et de division du travail, permettant de maîtriser la complexité. C'est d'ailleurs ce qu'on observe dans l'industrie du logiciel, où les innovations radicales sont généralement développées par de petits éditeurs de logiciel indépendants, qui se focalisent par exemple sur de nouveaux types d'application (Horn, 2000). Pour Aoki (2002), l'intérêt de cette stratégie s'est renforcée parce que les systèmes sont devenus tellement complexes que la *modularisation* s'est étendue aux modules eux-mêmes. La *modularisation* peut également être considérée comme un mode de gestion de l'innovation, car elle laisse chaque responsable de module libre d'innover, en respectant les

règles édictées par l'architecte. Cependant la tendance croissante à la *modularisation* des technologies et à la désintégration des systèmes implique la production de nouveaux types de connaissances (Steinmueller, 2002), telles que normes, standards, savoir d'intégration (Shapiro et Varian, 1999). Ces nouveaux types de connaissance sont nécessaires à la coordination, c'est-à-dire à l'intégration d'ensembles faiblement couplés. La connaissance est alors produite par deux types d'activités, la recherche et la coordination, cette dernière en pleine extension déterminant des processus d'innovation originaux (Pavitt, 2002). Comment ces connaissances émergent-elles ? Qui les construit ? Comment sont élaborées les design rules ? Ces questions restent ouvertes.

II-3. La flexibilité des processus de développement :

Différentes études ont exploré des modèles de développement de produit nouveau flexibles, caractérisés par le recouvrement des étapes de développement (Krishnan et alii 1997). Ces modèles sont fondés sur un processus qui met l'accent sur la capacité à générer et à prendre en compte de nouvelles informations aussi longtemps que possible pendant le développement de l'innovation (McCormack, Verganti & Iansiti, 2001). Plutôt qu'un processus séquentiel de type « stage-gate », le développement devient un processus « évolutionnaire » d'apprentissage et d'adaptation (Callon & Latour, 1991 ; Tushman et O'Reilly, 1997). Les activités procèdent d'une manière itérative, le retour d'information ou « feedback » obtenu lors d'un cycle d'expérimentation étant employé pour guider les activités du cycle suivant (Eisenhardt et Tabrizi 1995). C'est dans le secteur du logiciel que les modèles de développement flexibles ont été le plus largement préconisés (par exemple, le modèle « incrementale » Wong 1984 ; le modèle « spirale » Boehm 1988 ; ou le modèle « prototypage rapide » Connell et Shafer 1989). De tels modèles ont été élaborés pour résoudre les problèmes rencontrés par le modèle de développement traditionnel « waterfall ». Ce modèle de type « stage-gate » est le résultat des efforts entrepris par les entreprises pour parvenir à contrôler la gestion de grands projets de développement logiciel (Royce 1970). Les modèles flexibles de développement logiciel reposent sur un processus itératif, fondé sur la construction d'une série de prototypes qui doivent permettre de recueillir des *feedbacks* relatifs à l'adéquation entre le concept et les exigences des clients. La littérature observe qu'un certain nombre de sociétés commerciales utilisent ces approches sous des formes variées (Cusumano & Selby 1995, Iansiti & McCormack 1997). McCormack, Verganti et Iansiti (2001) ont travaillé sur la corrélation entre la flexibilité d'un modèle de développement logiciel et le niveau de performance des projets. Ainsi, ces auteurs étudiant le développement

de logiciels Internet, montrent que le niveau de *flexibilité du processus de développement* est effectivement corrélé avec son niveau de performance. Ce processus flexible se caractérise par la capacité à générer et répondre à de nouvelles informations sur une plus longue période du cycle de développement. Les construits qui soutiennent un tel processus sont en premier lieu de plus grands investissements dans la phase de conception de l'architecture du logiciel et des *feedbacks* du marché plus précoces sur les performances du produit. On observe que type d'analyse des processus de conception a été mis en évidence dans d'autres secteurs comme l'automobile ou la chimie (Midler, 1993 ; Charue-Duboc et Midler, 2002).

II.4 Formulation d'hypothèses :

En conclusion, nous chercherons à analyser la façon avec laquelle une innovation logicielle radicale, portée par différents éditeurs de logiciels, passe du stade de concept à celui de solution commercialisable. Nous faisons l'hypothèse que certains ces premiers clients, possédant les caractéristiques de *lead user*, jouent un rôle fondamentale dans le processus de conception d'une telle innovation. Nous faisons aussi l'hypothèse que la modularité de la solution favorise la flexibilité de ce processus, mais qu'elle pose le problème de l'intégration de la solution innovante, intégration que nous chercherons à analyser.

III Méthode :

Nous avons choisi, pour apporter des réponses à ces questions, d'étudier le cas d'une entreprise innovante, dans le domaine du logiciel. Nous nous sommes focalisé sur une succession de projets visant à déployer des applications mobilisant une même brique logicielle innovante chez différents clients. Nous comparons l'offre innovante et l'organisation de ces projets. Cette approche comparative vise à repérer des effets d'apprentissage et d'irréversibilité créés par les premiers produits innovants conçus pour les premiers clients par rapport aux produits de la même famille conçus pour les clients suivants.

III-1. Terrain de recherche :

L'entreprise SoftCo où nous avons conduit notre étude, édite des progiciels innovants et notamment des logiciels de *text mining*, qui permettent l'extraction, la catégorisation et la cartographie d'informations contenues dans un corpus de textes quelconque (articles de journaux, textes de lois, brevets...). Les logiciels de *text mining* de la société SoftCo reposent sur des fondements technologiques forts : SoftCo, en plus de ses

propres brevets, s'appuie sur des brevets développés au centre de recherche européen de Xerox, dont il a acquis les droits d'exploitation, et qui représentent des dizaines d'années homme de recherche. SoftCo dispose d'une gamme de logiciel dont les principaux sont le moteur d'extraction d'information ESoft (extraction terminologique), KSoft (catégorisation d'un document), CSoft (*clustering*, regroupement de documents présentant des similitudes) et MServer (application fédérant les différents logiciels de SoftCo). Pour chaque client, un développement personnalisé, qui complète ESoft, doit être réalisé pour définir exactement les termes et concepts que celui-ci veut extraire : ce développement est appelé par SoftCo *cartouche de connaissance*. Chaque vente suppose également des développements personnalisés pour intégrer ces logiciels, eux-mêmes en évolution, au système d'information du client final.

Cette famille d'innovations peut être considérée comme une innovation de rupture. D'une part elle s'appuie sur une combinaison d'algorithmes innovants d'analyse sémantique et d'analyse statistique. D'autre part, elle introduit une rupture dans les usages. En effet, l'offre que propose SoftCo à ses clients modifie les habitudes de travail des services qui l'adoptent, en permettant l'automatisation de l'analyse d'un texte : l'indexation et la catégorisation automatique (ou assistée par ordinateur), effectuées jusqu'alors manuellement, permettent ensuite un stockage électronique des documents (enrichis de méta-données, comme la date de parution), et de la connaissance qu'ils contiennent (comme le thème principal d'un article). Les utilisateurs peuvent ensuite effectuer des recherches parmi les documents et connaissances ainsi stockées.

III-2. Collecte des données

Nous avons suivi de manière longitudinale et en temps réel deux projets pour deux grandes entreprises différentes (groupe de presse et éditeur) visant chacun à développer une solution de création et de gestion de base de connaissance, basée sur des logiciels d'extraction, de catégorisation et d'archivage d'information. Ces différents logiciels ont été développés principalement par SoftCo (qui proposent les logiciels de *text mining* proprement dit) et une autre entreprise française de type start-up KnowCo (qui développe un système de gestion de base de connaissance). Il s'agissait dans ces deux projets de développer une application de ces logiciels destinée aux groupes de presse et éditeurs.

Le premier projet concerne la création d'une base de connaissance pour le groupe de presse PressPro. Il s'agissait de faciliter la création de dossiers de presse thématiques destinés aux journalistes. Il s'est déroulé de 2003 à mars 2006. Le deuxième projet porte sur la

création d'une base de connaissance pour l'éditeur EditPro. Il avait pour but la rédaction de notes de synthèses juridiques vendues aux professionnels du droit. Il a commencé en 2005, il doit s'achever par le déploiement de la solution début 2007.

Nous avons interviewé régulièrement, en moyenne une fois tous les deux mois, tout au long des projets, les personnes (chefs de projet, responsables techniques, *account managers*) travaillant pour SoftCo qui sont en relation avec le client final dans le cadre d'entretiens semi-directifs, ainsi que les responsables de projet chez PressPro et EditPro. Nous avons aussi assisté à des réunions de projet internes, participant en tout à trente-et-une réunions et entretiens, entre juin 2004 et juin 2006 :

- 15 entretiens avec des personnes impliquées, chez SoftCo, dans le projet PressPro,
- 12 entretiens avec des personnes impliquées, chez Softco, dans le projet Edit Pro,
- 2 réunions internes chez SoftCo, une sur chaque projet,
- 2 entretiens avec les chefs de projet chez les clients.

Le suivi longitudinal visait à limiter les biais liés à la rationalisation a posteriori (Girin, 1990 ; Yin, 1994 ; Dumez, 2004) et à constituer un matériau dans des entreprises où les documents formels sont peu nombreux.

Nous avons cherché à recueillir trois types d'informations sur la gestion du projet en interne chez SoftCo, sur la relation fournisseur/client, et sur le rôle du client dans le processus d'innovation. Nous avons posé des questions sur les choix d'organisation, les hypothèses initiales, les évolutions en cours de projet, l'influence du client sur ces évolutions, la nature des interactions mises en place entre le client et le concepteur des logiciels, et les difficultés rencontrées et ce pendant les trois phases de ces projets : conception, développement et déploiement des logiciels.

IV Analyse du cas :

Nous allons maintenant présenter les projets que nous avons suivis, ensuite nous reviendrons sur la participation des premiers clients à la définition de l'offre et de son architecture, puis nous décrirons la flexibilité du processus de développement, et enfin nous traiterons de la personnalisation de l'offre au travers du module logiciel appelé « cartouche de connaissance ».

IV-1. Présentation des projets :

A l'origine, le projet PressPro a été imaginé, en juin 2003, par la responsable de la documentation, récemment nommée. Son objectif principal touche à l'automatisation d'une

tâche d'indexation d'articles de presse « people », qui était jusqu'alors effectué manuellement par les documentalistes en charge de préparer des dossiers destinés aux journalistes de ce groupe de presse. Cette responsable a sollicité KnowCo et SoftCo, afin qu'ils s'associent pour créer une solution d'indexation d'articles de presse et de stockage de connaissance assistée par ordinateur. Ils ont ainsi rédigé une proposition chiffrée, qui a été acceptée par PressPro, sans qu'aucun prototype n'ait été développé. Toutefois, la responsable de la documentation de PressPro a demandé, aux deux éditeurs de logiciel de s'associer à XMLCo, un spécialiste du stockage de données au format XML, sa compétence semblant indispensable à PressPro. Aucun intégrateur n'a été sollicité pour ce projet, et au départ PressPro devait assurer le développement de l'application fédératrice, « appelant » les différentes briques logicielles. Mais face aux problèmes rencontrés par PressPro, cette tâche a fini par revenir à XMLCo et KnowCo. Enfin, le chef de projet de XMLCo a été nommé chef de projet de la solution logicielle, constituée des différentes briques de SoftCo, KnowCo et XMLCo, en tant que telle.

Pour ce qui est du projet EditPro, un membre de la Direction des Systèmes d'Information de cet éditeur, qui avait déjà travaillé avec des linguistes, a demandé à KnowCo, en avril 2004, de bâtir une solution permettant l'extraction et le stockage de la connaissance contenue dans des textes juridiques, en remplacement d'une solution ne permettant que d'extraire et de stocker des chaînes de caractères. KnowCo a alors proposé son offre couplée (grâce au projet PressPro) avec SoftCo. Dans un premier temps, lors de la phase de développement du « démonstrateur » (un prototype de la solution), le projet a été dirigé par l'initiateur du projet chez EditPro. Ensuite, cette tâche a été attribuée au chef de projet de SoftCo. Un intégrateur spécialisé, ConceptCo, a été choisi, chargé à la fois du développement de l'application fédératrice, et de l'intégration de la solution elle-même chez EditPro. Un autre intégrateur, généraliste celui-là, a même fait initialement partie du projet en tant que « superviseur », mais son rôle s'avérant inexistant, il en est rapidement sorti.

Les deux projets étudiés ont donc impliqué différents types d'acteurs :

- plusieurs éditeurs de logiciel (dont SoftCo et KnowCo dans les deux cas),
- le client final
- et dans le projet EditPro, un intégrateur ou *service provider*, c'est-à-dire une société de services informatiques chargée de faire en sorte que la solution logicielle intègre le système d'information du client (en l'occurrence ConceptCo).

- **Le premier projet, PressPro :**

<i>Début / Fin</i>	Octobre 2003 / Mars 2006 (projet poursuivi, devant aboutir à une « version 2 » de la solution).		
<i>Intégrateur de la solution logicielle</i>	Aucun (rôle assumé par les éditeurs de logiciel eux-mêmes)		
<i>Chef de projet logiciel</i>	XMLCo		
<i>Editeurs de logiciels participant au projet</i>	SoftCo	KnowCo	XMLCo
<i>Logiciels standards utilisés</i>	1. ESoft (extraction terminologique) 2. KSoft (catégorisation des documents)	ASL (Système de gestion de base de connaissance).	XMLCo Server (système d'archivage de documents au format XML) .
<i>Logiciels spécifiques développés pour le projet</i>	1. Cartouche « people » 2. Passerelle entre ESoft et ASL de KnowCo.	1. Application fédérant les logiciels de SoftCo, KnowCo et XMLCo 2. Thésaurus « people », 3. Couplage avec ESoft de SoftCo.	1. Filtre permettant la récupération des archives de PressPro, dans XMLCo server, 2. Passerelle entre ESoft et XMLCo Server.

- **Le deuxième projet, EditPro :**

<i>Début / Fin</i>	Mai 2005 / Mars 2007	
<i>Intégrateur de la solution logicielle</i>	ConceptCo	
<i>Chef de projet logiciel</i>	EditPro (pendant la phase de développement du démonstrateur), puis SoftCo.	
<i>Editeurs de logiciels participant au projet</i>	SoftCo	KnowCo
<i>Logiciels standards utilisés</i>	ESoft, KSoft	ASL
<i>Logiciels spécifiques développés pour le projet</i>	1. Cartouche « juridique », 2. Passerelle entre ESoft et ASL de KnowCo (adaptation de la passerelle « PressPro »)	1. Thésaurus « juridique », 2. Couplage avec le logiciel ESoft de SoftCo (adaptation du couplage « PressPro »)

Le suivi longitudinal que nous avons effectué sur plus de deux ans, nous a permis de voir émerger une offre innovante unique : le couplage d'un logiciel d'extraction (ESoft) et de

catégorisation de l'information (KSoft) avec un logiciel de gestion de base de connaissance (ASL), destiné au secteur du *publishing*.

IV-2. Le rôle des premiers clients dans la définition de l'offre et son architecture :

Dans le premier projet PressPro, on peut souligner trois moments dans lesquels le client a eu un rôle structurant dans le processus d'innovation.

Le rôle de PressPro a concerné tout d'abord la constitution d'un partenariat entre éditeurs de logiciel. A l'origine du projet PressPro, se trouve la nouvelle responsable du service documentation, qui sollicite SoftCo et KnowCo pour qu'ils associent leurs logiciels d'extraction terminologique et de gestion de base de connaissance. Ensuite, c'est encore PressPro qui suggère d'associer XMLCo, spécialiste de l'archivage de documents au format XML, en début de projet. Ainsi, le client va largement contribuer à définir l'architecture de la solution et à lui donner son caractère modulaire. La solution développée pour PressPro se compose donc de deux briques logicielles de SoftCo, une de KnowCo, une de XMLCo, auxquelles il faut ajouter *l'application documentaire* qui les fédèrent, et la *cartouche de connaissance* spécifique: en tout six modules logiciels distincts, plus les passerelles logicielles entre ces modules.

Dans un second temps, au cours de la phase de développement de la solution elle-même, les réunions entre groupes d'utilisateurs et chefs de projet logiciel, ont permis de « transférer » chez les éditeurs de logiciel l'information nécessaire concernant le métier de documentaliste et le contexte d'utilisation de la future solution. Les réunions quasi hebdomadaires des chefs de projet logiciel avec les *user groups* qui ont eu lieu pendant 12 mois montrent l'étendue des efforts à fournir pour y parvenir.

Enfin, pendant la phase de « pré-production », c'est-à-dire pendant la phase préalable au déploiement définitif, les documentalistes de PressPro ont formulé plusieurs demandes explicites de fonctionnalités (relatives à la constitution de dossiers et à la recherche d'information) auxquelles les trois éditeurs n'avaient pas pensé.

Le second projet a été lancé une fois le partenariat entre SoftCo et KnowCo établi et l'a développé. Le deuxième client a contribué à la structuration de l'offre, et à la réalisation du projet mais d'une manière plus « active » que le premier. Là aussi nous insisterons sur trois moments : lors de la structuration du projet, dans le cours du développement et en fin de projet pour son déploiement. Le client EditPro a proposé un intégrateur, de petite taille, ConceptCo, spécialiste de la gestion des connaissances ayant déjà travaillé pour lui.

ConceptCo a ensuite conclu un accord de partenariat avec SoftCo (comme « intégrateur référencé » ou *value added reseller*). La présence de ConceptCo n'a pas empêché qu'en parallèle SoftCo et KnowCo interagissent fortement avec EditPro dans le cadre du projet. La question de la participation d'un intégrateur s'était posée également dans le premier projet mais le client n'avait pas souhaité donner suite à cette demande des éditeurs de logiciel qui ont assumé la tâche d'intégration eux même. EditPro a aussi proposé un intégrateur « superviseur » du projet, mais qui a rapidement disparu du projet, son rôle s'avérant inexistant.

Ensuite, EditPro a développé lui-même une base de connaissance fictive pour réaliser des tests logiciels, ce qui l'a amené à réclamer l'amélioration de certaines performances du système comme les temps d'accès à la base. Ceci s'est avéré très utile aux dires des chefs de projet de SoftCo et KnowCo, car cela a aussi permis l'amélioration de l'interfaçage entre logiciels. (Ces deux évolutions ont même pu être intégrées au projet PressPro, qui n'était pas encore terminé.)

Enfin, le client EditPro a accepté et même réclamé que SoftCo forme ses linguistes sur sa *cartouche de connaissance* personnalisée, afin qu'il puisse dans un premier temps en assurer la maintenance, et à terme réaliser ses propres solutions.

Sur le premier projet, de nombreuses incertitudes entouraient le début du processus de développement. Tout d'abord, le client n'a pas pu tester de prototype de la solution. En effet, l'élaboration rapide d'un prototype n'était pas envisageable car l'interface entre les logiciels de SoftCo, de KnowCo et de XMLCo n'existait pas. En l'absence de prototype, la définition de spécifications de la solution a été plus difficile. Les spécifications sont restées floues quant au niveau de performance attendu (pas de quantification de la qualité de l'extraction) et à la nature des documents, que le système devait être capable de traiter. La rédaction de spécifications précises relatives à l'extraction terminologique n'était pas évidente car le client ne savait pas exactement ce qu'il désirait sur ce point, si ce n'est « la meilleure solution possible ». Les interactions entre éditeurs de logiciel et utilisateurs ont été nombreuses et régulières, sauf entre SoftCo et PressPro, à propos de la brique logicielle d'extraction d'information : cette interaction n'a eu lieu que tout à la fin du processus, car la responsable de la documentation de PressPro, pensait que le travail sur les briques logicielles de KnowCo et de XMLCo était prioritaire. En effet, ces briques permettant le stockage et la recherche d'information et d'articles, sont indispensables aux documentalistes, mais elles sont moins radicalement innovantes que la brique d'extraction d'information de SoftCo. L'implication

des utilisateurs dans le développement de cette brique n'a pu se faire qu'en fin de processus. Des difficultés sont alors apparues, alors qu'il restait peu de temps pour achever le projet. Il s'est avéré alors, que les attentes implicites de PressPro par rapport à l'extraction automatique d'information étaient trop élevées par rapport aux résultats fournis à ce moment-là par la cartouche de connaissance couplée au logiciel IDE. En effet, les articles de presse que PressPro voulait analyser, sont rédigés d'une manière plus littéraire et moins purement factuelle que des textes de lois ou de brevets, par exemple, ce qui rend l'extraction terminologique plus complexe. Ainsi, le projet PressPro s'est achevé d'une manière mitigée, puisque la solution mise en place ne donnait pas entière satisfaction, pour la partie incombant à SoftCo.

Mais, sans le travail effectué sur le projet PressPro, le projet EditPro n'aurait pu être déployé de la même manière. Les principales modifications dans la façon de gérer le projet, qui ont été prises en compte dans le projet EditPro sont les suivantes :

- un développement rapide d'un prototype testable de la solution, qui a permis une meilleure compréhension des possibilités du futur logiciel par le client.
- une rédaction beaucoup plus précise du cahier des charges et des spécifications, qui ont ensuite permis une meilleure compréhension et une meilleure appropriation du projet par le client, mais aussi un meilleur cadrage du travail des équipes-projet de SoftCo, KnowCo et EditPro.

On note aussi des différences majeures, relatives aux compétences de l'intégrateur et du client, entre les deux projets, qui aux dires des membres de l'équipe-projet de SoftCo, ont contribué à faciliter le déroulement dudit projet :

- l'équipe-projet du client EditPro dispose d'une compétence relative à son métier, mais aussi aux technologies d'extraction terminologique du *text mining* et aux systèmes de gestion des bases de connaissance. Ainsi, le chef de projet EditPro connaît à la fois le *text mining* et la gestion des bases de connaissance, tandis que deux futurs utilisateurs du système, membres de l'équipe-projet EditPro, possèdent une compétence en linguistique (utile pour aborder l'extraction terminologique) en plus de leur compétence métier. Cela leur a permis, selon les membres de l'équipe-projet de SoftCo, de mieux comprendre ce qu'ils pouvaient attendre du futur système.
- l'intégrateur ConceptCo a su développer, dans le cadre du projet EditPro, l'application fédérant les différentes briques logicielles, en s'appuyant à la fois sur ses compétences

en intégration logicielle et en gestion de base de connaissance. Il a permis à SoftCo et KnowCo de se concentrer sur leurs briques logicielles, en interaction avec le client.

- Une méthodologie a été développée, qui définit le rôle de chaque acteur, leurs interactions, et les livrables qu'ils doivent remettre au fur et à mesure du processus de développement d'une solution de création automatisée de base de connaissance. Elle définit en particulier le rôle d'un « intégrateur de couplage », qui est en charge du couplage entre les logiciels de SoftCo et de KnowCo. Cette méthodologie a pu être appliquée avec succès dans la deuxième moitié du projet EditPro.

En synthèse, le travail effectué dans le cadre de ce premier projet a donc permis à SoftCo et KnowCo de faire l'apprentissage, notamment grâce au client PressPro, de la création d'une offre logicielle innovante destinée au secteur du *Publishing*. La comparaison des deux projets successifs nous incite à souligner des acquis du premier projet sur lesquels le second projet s'est appuyé :

- SoftCo et KnowCo ont défini lors du premier projet une offre standard conjointe regroupant plusieurs de leurs logiciels, et les passerelles logicielles nécessaires à leur interfaçage,
- les chefs de projet de SoftCo et de KnowCo ont défini une méthodologie, des modalités de collaboration, pour leurs futurs projets communs,
- les difficultés rencontrées sur le projet PressPro, ont conduit SoftCo et KnowCo à modifier leur façon d'appréhender le projet EditPro, dans le but de permettre une meilleure compréhension et une meilleure appropriation du projet par le client (compréhension et appropriation qui sont, par ailleurs, facilitées par la nature et l'étendue des compétences du client).

IV-3. Un processus complexe mais flexible :

Par ailleurs, on observe une grande flexibilité dans l'organisation du projet PressPro, sans que cela entraîne de gros problèmes de délais (il n'y aura « que » quelques mois de retard par rapport au planning initial). Cette flexibilité concerne surtout les tâches à effectuer et les responsabilités relatives à ces tâches : ainsi le nombre d'éditeurs de logiciel impliqués dans le développement de la solution logicielle change au cours du temps (un puis deux, puis trois éditeurs de logiciel en relation avec le client) ; on observe aussi que la responsabilité du développement de *l'application documentaire*, qui doit fédérer les différentes briques logicielles est transférée après plusieurs mois de PressPro à XMLCo, dont ce n'est d'ailleurs

pas le métier ; de même, la récupération des articles de presse déjà stockés par PressCo, qui devait être réalisée par XMLCo, l'est finalement par EditCo ; par ailleurs, les groupes d'utilisateurs finissent par refuser de participer aux réunions de projet à partir de début 2005, mais cela ne bloque pas le processus qui se poursuit avec un seul groupe d'utilisateurs. On note enfin que le chef de projet « théorique » de la solution globale, qui est aussi le chef de projet de la partie « XMLCo », n'exerce qu'un contrôle très restreint, ce qui contribue aussi à la grande souplesse du projet global, tout en générant un flou qui gêne les équipes-projet de SoftCo et de KnowCo. Il faut aussi souligner la flexibilité dont font preuve les éditeurs de logiciel. Leur taille réduite est favorable à une telle flexibilité mais leur motivation quant à l'aboutissement de projets, qui peuvent devenir autant de références et dont ils pensent qu'ils constituent un moyen unique d'apprentissage et de concrétisation de leur offre innovante est déterminante. Ainsi par exemple, lorsque XMLCo tarde à développer le filtre permettant de récupérer les archives stockées dans l'ancienne base de connaissance de PressPro, ce sont des développeurs de SoftCo qui effectuent cette tâche afin de ne pas ralentir leur partie de projet, et ce sans contrepartie de la part de XMLCo. De même, lorsque les ressources humaines viennent à manquer, SoftCo n'hésite pas à détourner de leur tâche des développeurs chargés des logiciels standards ou *core products* de SoftCo.

IV- 4. Cartouches de connaissance et personnalisation de l'offre

Pour pouvoir exploiter les logiciels de SoftCo, il est nécessaire de développer ce que SoftCo appelle une *cartouche de connaissance*, qui doit contenir la terminologie propre au secteur étudié (par exemple l'intelligence économique dans le secteur pétrolier, le pré-traitement des CV reçus par la DRH d'une banque). Initialement, SoftCo pensait que chaque client pourrait développer sa propre *cartouche de connaissance* après avoir défini les concepts et les termes qu'ils souhaitaient extraire automatiquement. SoftCo fondait beaucoup d'espoir sur sa boîte-à-outils appelée *SDK*, environnement de développement et ensemble d'outils destinés à aider les programmeurs dans leur travail et à faciliter la création de la partie personnalisée de la solution, par le client lui-même. Cet outil était donc fourni aux clients. Dans le projet PressCo cependant, SoftCo dût assurer lui-même le développement de ces cartouches. Rapidement, SoftCo a dû constater que ses clients n'utilisaient pas cette boîte-à-outils, peu convivial. Les SSII contactées ne se sont pas montrées intéressées par le développement de ces cartouches, estimant que le marché du *text mining* n'était encore que balbutiant, et rechignant à s'y investir. SoftCo s'est ainsi résolu, dans un premier temps, à prendre en charge ces développements et n'utiliser cette boîte-à-outils qu'en interne.

Néanmoins, le *SDK* n'a pas été abandonné, et un ingénieur continue d'y travailler régulièrement, en parallèle des projets : la stratégie de SoftCo consiste à ce que des partenaires *var's*, intégrateurs, développent un jour des *cartouches de connaissance* pour leur client. Pour qu'à moyen terme, ce soit le client final qui développe lui-même sa *cartouche de connaissance*, il faudrait que l'interface homme machine et l'interface de navigation soit améliorées, qu'il existe une documentation, et que les développeurs de SoftCo aient du recul sur son utilisation. Dans l'esprit des responsables de SoftCo, le *SDK* doit constituer la dernière étape de création de l'offre SoftCo, celle qui doit permettre à cet éditeur de limiter ces activités de services pour se concentrer sur la vente de logiciel *sur étagère* ou *off the shelf* censée être la plus rentable. De plus, le développement personnalisé effectué par l'intégrateur ou le client final, résoudrait un problème critique pour SoftCo, constitué par le suivi et la maintenance des logiciels personnalisés ainsi développés.

V Discussion

V-1. Lead-user ?

On observe donc que les clients étudiés présentent des caractéristiques de *lead users*, et qu'ils jouent un rôle clef dans le processus d'innovation, confirmant ainsi notre première hypothèse. Tout d'abord, les clients recherchaient des solutions à leurs problèmes spécifiques. L'innovation en satisfaisant leur besoin devait leur apporter des gains : gain de productivité et enrichissement des tâches des utilisateurs pour le projet PressPro, offre enrichie pour le projet EditPro porteuse de valeur ajoutée pour les clients de l'entreprise. Ils sont allés jusqu'à définir les contours de l'offre de « création de base de connaissance assistée par ordinateur » qui leur convenait, en retenant plusieurs éditeurs de logiciel et en leur demandant de s'associer pour l'occasion. Après la phase de sélection des éditeurs de logiciel, une relation étroite s'est instaurée entre le client, tant au niveau des utilisateurs que de son service informatique, et les éditeurs de logiciel. Ces derniers mettent en place une organisation qui leur permet de travailler de manière organisée et régulière avec des groupes de futurs utilisateurs, ainsi qu'avec la direction des systèmes d'information du client. Pour ces raisons, on peut considérer que ces clients ont participé au processus de conception de cette solution innovante.

De plus, on peut souligner que les besoins de ces utilisateurs sont pointus mais aussi à l'avant garde d'un marché que l'on peut anticiper concernant l'analyse automatisée de texte libre. En effet, le volume croissant d'information auquel organisations et individus peuvent avoir accès, laisse présager un besoin croissant d'analyse automatisée de texte libre dont la croissance

pourrait être à l'image de celle du marché du *data mining* qui concerne l'analyse et le traitement d'informations formatées stockées dans des bases de données.

Cependant, on constate que le client ne possède pas la maîtrise de la technologie portée par l'innovation (analyses linguistique et statistiques combinées), ni la capacité d'intégrer un projet incluant plusieurs éditeurs de logiciels devant fournir une offre commune. Le *text mining* est une technologie nouvelle, et même un concept nouveau. A ce titre il est peu connu des entreprises, et il semble normal que les premiers clients manquent de repère quant à son potentiel, ses limites et à la façon de la mettre en oeuvre. Ceci contribue naturellement à limiter leur capacité d'appropriation. L'interaction directe entre éditeurs de logiciel innovant et utilisateurs, a d'ailleurs aussi pour but d'« évangéliser » ces derniers, c'est-à-dire de les convaincre de la pertinence du concept, tout en essayant de comprendre comment ils le perçoivent. Certes PressPro, le premier client, a eu l'idée d'associer des logiciels de différents éditeurs, mais il n'a pu réellement contrôler le projet, ni en percevoir les limites. Le client PressPro ne va pas au bout du processus d'innovation, il n'est pas capable de bâtir une solution complète satisfaisant ses besoins propres, mais son action contribue à l'élaboration d'une offre finalisée. Par contre, le client EditPro participe activement au projet, et est en train de se former, pour pouvoir aussi rapidement que possible développer ses propres *cartouches de connaissance*.

Les *communautés d'utilisateurs* constituent pour von Hippel (2001) un moyen unique de partage d'information et de création d'une innovation. Elles sont particulièrement soulignées dans l'industrie du logiciel, open source comme propriétaire. Dans les cas étudiés, on ne peut parler de *communauté d'utilisateurs* car il existe naturellement peu d'utilisateurs, et car ceux-ci ne possèdent pas encore l'expertise nécessaire à la personnalisation du logiciel. Néanmoins, le nombre d'utilisateurs et leur capacité à réaliser leurs propres solutions augmentant, cette situation peut évoluer. Comme nous l'avons dit, le *text mining* est un concept nouveau, et à ce titre difficile à appréhender par les entreprises. Lancer une offre de *text mining* passe par la diffusion de ce nouveau concept, la constitution de compétences et la construction d'un nouveau marché.

V-2. Modularité par construction et création de l'offre

Les entreprises développant des logiciels innovants sont souvent de taille réduite, cela va de pair avec un processus de spécialisation (Horn, 1999). Ceci entraîne un morcellement de l'offre et rend plus complexe la constitution de solutions globales. Ainsi, SoftCo, KnowCo et

XMLCo se connaissaient, mais avant la demande expresse d'un client, n'avaient pas pensé à s'associer pour proposer une offre commune. Ils considéraient même que dans une certaine mesure, ils pouvaient se passer des compétences des deux autres éditeurs, s'estimant eux-mêmes suffisamment experts dans leurs domaines privilégiés. Ce projet a débouché sur une association formalisée, et une offre logicielle conjointe entre SoftCo et KnowCo, de création et de gestion de base de connaissance, utilisant les technologies de *text mining*. Cette modularité par construction va permettre aux utilisateurs de tester et de faire des retours d'information par module : ainsi, dans le cas PressPro, les documentalistes ont pu tester une solution ne comprenant que les modules de KnowCo et de XMLCo, avant que les modules de SoftCo ne soient finalisés.

Les cas étudiés suggèrent que la spécialisation des éditeurs de logiciels innovants, conduit à l'intervention auprès des clients de plusieurs acteurs à la fois. Ceci complique la tâche du client, aspiré dans un processus d'interaction avec plusieurs acteurs en parallèle, et celle des éditeurs de logiciel qui doivent coordonner leurs travaux et la collaboration avec leur client. Au départ, dans le projet PressCo, ni le client, ni les éditeurs de logiciel concernés ne souhaitaient fonctionner de la sorte. KnowCo devait initialement être le seul interlocuteur de PressPro. Ce fut impossible, car les compétences de KnowCo étaient insuffisantes en termes d'extraction terminologique. Les deux éditeurs ont été amenés à travailler directement avec leur client, et ont même du faire appel au troisième, XMLCo, pour la solution d'archivage des articles de presse. L'intervention d'une société de services, en charge uniquement de l'intégration de la solution comme c'était le cas sur le deuxième projet n'a pas conduit à supprimer ces multiples interactions.

Cette structuration modulaire multi-acteurs de l'offre entraîne une multiplication des interactions entre le client et les éditeurs de logiciels, et entre les éditeurs de logiciels eux-mêmes. Ces interactions permanentes, créent une dynamique favorisant l'évolution des briques logicielles et de leurs associations, pour aboutir à la solution recherchée. Elles permettent aussi, à chaque éditeur de logiciel, de « vendre » le nouveau concept aux utilisateurs, après l'avoir testé et reformulé, dans une démarche dialectique. Ces interactions doivent être aussi précoces que possible, ce qui passe notamment par le développement rapide d'un prototype (ou « démonstrateur ») disposant de fonctionnalités limitées, mais permettant l'établissement d'un dialogue avec les utilisateurs, ce qui a pu être fait dans le projet EditPro. Le premier projet ayant permis l'émergence d'une architecture fondée sur le couplage de différents logiciels, il a ensuite été possible à SoftCo, KnowCo et XMLCo de développer un

démonstrateur pour le deuxième projet, Edit Pro. Cela a largement contribué à la bonne intégration des informaticiens et utilisateurs d'EditPro dans le processus d'innovation.

Cette modularité permet la parallélisation du travail sur les différentes tâches, même si certaines d'entre elles (définition des interfaces bien sûr mais pas uniquement), nécessitent un travail commun de plusieurs éditeurs. Ce dernier point est un des facteurs principaux qui contribuent à flexibiliser le processus d'innovation, et donc à améliorer ses performances (McCormack, Verganti et Iansiti, 2001). La structure naturellement modulaire d'une offre constituée de logiciels développées par des sociétés indépendantes les unes des autres, permet de la faire évoluer sans remettre en cause son architecture, ni chacun des modules qui la composent (Baldwin et Clark, 1997).

Ainsi l'hypothèse selon laquelle la modularité de l'offre innovante favorise la flexibilité du processus est-elle vérifiée, et nous observons qu'elle favorise aussi son appropriation par le client, en ce qu'elle permet de multiples interactions entre celui-ci et chaque éditeur de module logiciel, et en ce qu'elle facilite les tests et feedbacks portant sur des sous-parties de la solution. Par contre, elle complexifie la vision globale du projet, et le travail d'intégration de la solution, ce qui corrobore aussi notre hypothèse. Le problème de l'intégration a été résolu d'une part en intégrant au process un intégrateur spécialisé en charge de l'intégration globale de la solution, et d'autre part en définissant le rôle d' « intégrateur de couplage », en charge de l'interfaçage personnalisé entre le moteur d'extraction de SoftCo et la base de connaissance de KnowCo. Cette définition s'inscrit dans le cadre de la méthodologie développée par KnowCo et SoftCo, qui vise à définir les rôles de chaque acteur, leurs interactions et les livrables qu'ils doivent fournir. Ainsi, ce dispositif permettra d'organiser la collaboration entre acteurs dans les projets ultérieurs et de résoudre le problème de l'intégration d'une solution modulaire multi-acteurs.

V-3. Les *toolkits*, compétences requises au niveau des clients et obstacles :

Le concept de *toolkit* (ou boîte-à-outils) permettant au client de développer sa propre solution, s'il peut être pertinent à terme, ne semble pas opérationnalisable dans la phase de lancement de l'offre. Ceci paraît contredire les résultats de certains travaux (von Hippel, 2002). Nous nous proposons donc d'étudier plus finement ce point dans la suite de notre travail. De plus, SoftCo, capitalisant sur des projets antérieurs peut proposer dans sa gamme des *cartouches de connaissance* génériques : par exemple, le développement d'une

cartouche de connaissance d'intelligence économique dans l'industrie pétrolière pour un client particulier, a permis d'ajouter à la gamme des logiciels standards de SoftCo, une *cartouche de connaissance* générique destinée à l'intelligence économique, adaptable à chaque secteur économique. Nous observerons ainsi l'élaboration d'une offre logicielle innovante, permettant la création de bases de connaissance destinées à des fournisseurs de contenus (textes) numériques.

VI Conclusion :

Notre recherche met en évidence un lead-user dont la compétence porte d'abord sur les usages et dont la contribution à la conception de logiciel innovant concerne l'architecture d'une offre complète. Ce résultat dans le domaine du logiciel propriétaire se distingue assez nettement des travaux sur le logiciel libre qui soulignent le rôle des lead-user dans l'écriture du code des logiciels innovants. Dans notre cas, le client sélectionne les « briques » logicielles de différentes entreprises à associer afin de satisfaire son besoin propre et met en place entre ces entreprises des contrats afin d'obtenir la réalisation d'une application intégrée. Cela conduit chacune des entreprises à modifier sa « brique » et à créer les passerelles permettant à l'ensemble de l'application de fonctionner. Les entreprises écrivent également le code nécessaire à l'intégration de cette nouvelle application dans le système informatique du client. L'architecture est conçue par le premier client pour ses besoins spécifiques, et elle est ensuite reprise par les entreprises de logiciel dans leur proposition au second client. Nous montrons alors comment les éditeurs de logiciel étudiés s'approprient l'apport du premier client à la conception du logiciel. Il est difficile dans notre cas de mettre en évidence une communauté d'utilisateurs mais l'effet d'apprentissage que souligne par von Hippel (2001) se retrouve. Il se situe dans l'enchaînement des projets et repose sur le fait que les mêmes entreprises logicielles sont amenées à coopérer de manière récurrente pour des offres similaires.

Le second domaine que notre travail éclaire a trait à l'élaboration des *design rules* qui permettent une modularisation des systèmes et de leur conception. Dans le cas des ordinateurs, ces *design rules* ont été constituées ex-ante par un acteur dominant du secteur, par exemple Intel. Notre recherche met en évidence un autre processus de construction des *design rules* (Baldwin & Clark, 1997). La structure modulaire résulte de l'existence d'une pluralité de sociétés logicielles spécialisées qui ont développé des briques spécifiques. Les *design rules* émergent progressivement lorsque ces entreprises sont amenées à coopérer pour construire une offre globale satisfaisant les besoins d'un client et nécessitant l'association de

ces briques. Alors que la dynamique dans l'industrie informatique a été vers une désintégration verticale s'accompagnant d'une modularisation, nous analysons un mouvement inverse de combinaison d'offres spécialisées en une offre globale intégrée. Nous mettons alors l'accent sur le travail nécessaire pour rendre les modules interconnectables, et la conception logicielle qui ne peut se faire qu'en situation de coopération. Nous montrons alors que l'élaboration de *design rules* est répartie entre deux acteurs : les *lead users* qui exigent de plusieurs entreprises de coopérer et ces mêmes entreprises qui adaptent leur modules, conçoivent les interfaces manquantes afin de les rendre interconnectables, et élaborent une méthodologie qui définit les rôles, interactions et livrables pour chaque acteur. Ce faisant, ils contribuent à résoudre le problème de l'intégration de la solution, problème dont nous avons pu vérifié l'acuité dans le cas d'une solution modulaire multi-acteurs.

Le troisième point sur lequel nous reviendrons a trait aux *toolkits* ou boîtes-à-outils. Les entreprises de logiciels ont conçu leur offre avec des « cartouches » personnalisables permettant l'adaptation du logiciel aux besoins spécifiques des clients. Ces boîtes-à-outils sont apparus de peu d'utilité pour impliquer les clients à ce stade de maturité de l'innovation où la priorité est la construction d'un premier prototype. L'implication demandée au client était à la fois trop exigeante et décalée par rapport à son souci de construction d'une offre globale satisfaisant son besoin de recherche documentaire. A ce stade, la conception avec cartouche personnalisable est apparue cependant intéressante pour faciliter les effets d'apprentissage inter-projet de l'entreprise logiciel et limiter les coûts d'adaptation à chaque client et non pour impliquer le client dans la conception logicielle.

L'évolution constatée entre le premier et le deuxième projet, nous incitent à faire l'hypothèse d'une maturation progressive de l'offre par l'enrichissement des différents projets, parallèle à une maturation de l'interaction entre l'éditeur de logiciel innovant et les premiers utilisateurs. Les premiers projets traversent une phase pendant laquelle des clients, motivés mais ne maîtrisant pas l'innovation, contribuent à faire émerger une offre, en collaborant avec des éditeurs de logiciel qui eux doivent bâtir une offre modulaire, et la contextualiser, pour la rendre opérationnelle. Ensuite, une fois cette offre stabilisée, les conditions sont réunis pour qu'une boîte-à-outils utilisable par les clients ou les sociétés de service puisse être utilisé par ceux-ci, et qu'ensuite une communauté d'utilisateurs émerge.

Cette hypothèse d'une reprise en main du développement d'une solution personnalisé par le client, dans une phase ultérieure, et ensuite de l'émergence d'une communauté d'utilisateurs, dans le cas d'innovation radicale serait intéressante à tester dans des recherches futures.

Bibliographie :

- Aoki M. (2002), "Modularity : Its Relevance to Industrial Architecture", Le Centre de recherche économique de Saint-Gobain.
- Baldwin, C. Y. and K. Clark (1997), "Managing in the Age of Modularity", *Harvard Business Review*.
- Charue-Duboc F. et Midler C. (2002), "L'activité d'ingénierie et le modèle de projet concourant", *Sociologie du travail*, vol 44, pp 401-417
- Cusumano, M. (2004), *The Business of Software*, Free Press, New York.
- Dumez H. (2004), « Elaborer la théorie à partir des données », *Revue Sciences de Gestion*, N°44, pp. 139-155.
- Fichman, R.G. and Kemerer, C.F. (1997) "The Assimilation of Software Process Innovations: An Organizational Learning Perspective", *Management Science*, vol. 43(1), pp. 1345-1363.
- Girin J. (1990). L'analyse empirique des situations de gestion : éléments de théorie et de méthode. In (s/d) Martinet, *Épistémologies et sciences de gestion*. Paris: Éditions Economica.
- Harhoff D., Henkel J. & von Hippel E. (2000), "Profiting from voluntary information spillovers: How users benefit by freely revealing their innovations", MIT Sloan School of Management.
- Horn F.(1999), "Diversité des informations traitées par des moyens informatiques, standardisation optimale et acteurs du processus de standardisation", *Communications et stratégies* n°33, pp. 85-117.
- Horn F. (2000), « L'économie du logiciel », Thèse de doctorat d'Economie Industrielle, Université de Lille I.
- MacCormack A., Verganti R. & Iansiti M. (2001), "Developing Products on "Internet Time" : The Anatomy of a Flexible Development Process", *Management Science*, vol. 47, N°1, pp. 133-150.
- Midler C., *Twingo, l'auto qui n'existait pas*, InterEdition, Paris, 1993.

- Pavitt, K. (2002) “System integrators as “post-industrial” firms”, DRUID Conference, Copenhagen Business School.
- Raymond E. (2001), *The Cathedral and the Bazaar (revised edition)*, O'Reilly, 2001.
- Shapiro, C. & Varian, H. R. (1999) “The art of standards war”, *California management review*, 41 (2): pp. 8-32.
- Steinmueller, W.E. (2002) “Collaborative innovation: rationale, indicators and significance.” In: J. de la Mothe and A.N. Link (eds), *Networks, Alliances and Partnerships in the Innovation Process*; Boston: Kluwer, pp. 29-43.
- Thomke S. & von Hippel E., (2002). « Customers as Innovators , a New Way to Create Value », *Harvard Business Review*, April, pp. 74-81.
- von Hippel, E. (1986). “Lead Users: A Source of Novel Product Concepts”, *Management Science*, vol. 32, no. 7: pp. 791–805.
- von Hippel, E. (1994) “Sticky Information and the Locus of Problem Solving : Implications for Innovation”, *Management Science*, vol. 40,4 : pp. 429-439.
- von Hippel, E. (2001). “Perspective: User Toolkits for Innovation”, *Journal of Product Innovation Management*, vol.18, pp. 247–257.
- von Hippel, E., & R. Katz. (2002). “Shifting Innovation to Users via Toolkits.” *Management Science*, vol. 48, no. 7: pp. 821–833.
- von Krogh, G. & von Hippel, E. (2003). « Special Issue on Open Source Software development», Editorial in *Research Policy* 32/7 : pp. 1146 - 1157.
- Yin, R. (1994). *Case study research: Design and methods* (2nd ed.). Beverly Hills, CA: Sage Publishing.