

Numéro de bibliothèque

--	--	--	--	--	--	--	--	--	--

THÈSE
PRESENTÉE À
L'UNIVERSITÉ DE PAU
ET
DES PAYS DE L'ADOUR

**ÉCOLE DOCTORALE DES SCIENCES EXACTES ET DE LEURS
APPLICATIONS**

PAR

Raphaël MICHEL

POUR OBTENIR LE GRADE DE

DOCTEUR

Spécialité :

INFORMATIQUE

**SYSTÈME D'INFORMATION POUR L'ÉVALUATION DES
COMPOSANTS LOGICIELS SUR ÉTAGÈRE
(COTS COMPONENTS)**

Soutenue le 4 décembre 2006

Après avis des Rapporteurs :

Mr Bernard COULETTE, Professeur à l'université de Toulouse le Mirail,

M^{me} Colette ROLLAND, Professeur à l'université de Paris1-Panthéon-Sorbonne.

Devant la commission d'examen formée par :

Le président,

Mr Pascal WEIL, Directeur de recherche au LaBRI.

Les examinateurs,

Mr Franck BARBIER, Directeur de thèse, Professeur à l'UPPA,

Mr Philippe ROOSE, Encadrant, Maître de conférences à l'UPPA.

Remerciements

Je remercie vivement Franck BABIER, directeur de recherche à l'UPPA, pour m'avoir accueilli dans son laboratoire et m'avoir permis de réaliser cette thèse.

J'exprime aussi ma grande reconnaissance à Philippe ROOSE, maître de conférences à l'UPPA, pour m'avoir encadré mais surtout soutenu dans les moments difficiles et de doutes. Ses conseils et ses commentaires ont été très importants pour la réalisation de ce travail. Je tiens à le remercier pour l'énorme effort qu'il a fait pour vérifier et corriger le texte de ce mémoire et pour son aide lors de la rédaction de ce manuscrit

Je remercie très chaleureusement Pascal WEIL d'avoir accepté de présider le jury de cette thèse, ainsi que les membres du jury B. Coulette et C. Rolland pour leurs commentaires, leurs remarques pertinentes m'ayant permis d'améliorer le manuscrit.

Le travail de recherche n'exclut pas des moments difficiles surtout lorsque celui-ci est mené en parallèle avec un travail de consultant. Je remercie l'équipe du CRT Estia Innovation et à mes collègues de travail qui m'ont soutenus dans les moments difficiles et de doutes entre ma thèse et le consulting.

J'exprime toute ma sympathie à l'ensemble du personnel de l'IUT de Bayonne pour leurs sympathies et à tous ceux qui m'ont soutenu dans ce travail

Je ne saurai terminer mes remerciements sans remercier ma famille, pour son soutien moral tout au long de ma thèse. Je tiens donc à témoigner toute ma reconnaissance à ma femme Valérie et à mes deux puces Lucie et Alizée et ma gratitude à mes parents à qui je dédie ce travail.

Résumé

Les utilisateurs d'applications informatiques sont aujourd'hui demandeurs de nouvelles fonctionnalités tout en exigeant une amélioration des performances. Tous les logiciels sont concernés par ce phénomène. Les progrès du matériel informatique (processeur, bus de données, mémoire, traitement vidéo...) permettent d'envisager des améliorations dans les rendus aussi bien que dans les fonctionnalités que les développeurs peuvent proposer à leurs clients. L'intégration du multimédia, des effets graphiques 3D ainsi que des outils de calculs performants sont présents dans la plupart des produits. De ce fait, les développeurs d'applications se spécialisent en fonction des technologies employées et des métiers. Ils proposent de plus en plus des produits logiciels sur étagères ou « COTS component ». Ainsi ils mettent à disposition des développeurs des produits intégrant des fonctionnalités difficiles à mettre en œuvre ou spécifiques à un métier donné.

En effet, la taille des applications informatiques est de plus en plus importante. L'accroissement des services offerts et leur complexité rendent les développements de plus en plus difficiles à réaliser. Donc, l'utilisation de ces « COTS component » semble être la solution à ces problèmes. La complexité d'une partie de l'application est ainsi sous-traitée à un fournisseur spécialisé qui est en charge de l'amélioration de ses produits au niveau de la qualité, des performances ou encore de l'accroissement des fonctionnalités. De ce fait, l'idée même de pouvoir développer une application à l'aide de plusieurs produits du marché est accueillie favorablement par les développeurs. Ils espèrent faire des gains de temps et de coûts de réalisation, mais aussi pensent pouvoir s'assurer d'une qualité logicielle qu'ils auraient du mal d'obtenir et/ou à maintenir.

Si la réalisation d'applications à l'aide de ces « COTS component » est de notre point de vue une bonne solution qui se généralisera à l'avenir, il ne faut pas masquer les réelles difficultés d'intégrations de ces logiciels aujourd'hui. Les efforts nécessaires pour découvrir et sélectionner les COTS, pour les intégrer dans l'application à base de COTS et enfin pour les mettre à jour ne doivent pas être minimisés. Pour aider les développeurs, des processus spécifiques à ce type de développement sont présents dans la littérature. Mais l'une des grandes difficultés lors de l'intégration se situe sur le recueil et l'obtention des informations permettant l'évaluation des produits nécessaires au processus de

développement. Pour les obtenir, les développeurs réalisent des prototypes à partir desquels les COTS seront évalués. Ils permettront le recueil des informations si difficiles à obtenir de la part des vendeurs. Il semble aujourd'hui que c'est la meilleure des solutions pour les acquérir.

Pourtant, le développement d'un prototype nécessite lui aussi des connaissances sur la manière d'intégrer les COTS. Or, ces informations étant elles aussi, difficiles à obtenir, nous avons décidé de proposer aux développeurs un moyen facilitant le recueil d'informations par l'intermédiaire d'une relation client/fournisseur, l'intégration des COTS en utilisant une base de connaissances, la capitalisation de celles-ci et la détection des conflits ou des difficultés connues lors de l'intégration.

Notre proposition est basée sur l'utilisation d'un Système d'Information. Il contient les cartes d'identités des différents produits proposés sur étagères par des fournisseurs de COTS. Les développeurs peuvent par exemple identifier l'ensemble des services proposés par chaque logiciel, les configurations logicielles requises et la liste des dysfonctionnements, notamment avec d'autres logiciels. Ce dernier cas permet au Système d'Information de lister les difficultés prévisibles ou les dysfonctionnements connus pour l'ensemble des COTS faisant partie de l'application à réaliser. On détecte ainsi les produits à risque ou nécessitant une évaluation particulière afin de maîtriser les risques et les coûts d'intégration. C'est donc une aide indispensable lors des phases de sélection et de tri de ces composants car le développeur peut traiter au plus tôt dans le processus de développement, ces COTS identifiés comme à risque. Le gain de temps et de qualité de l'évaluation est immédiat, d'autant plus que le développeur peut décider de rechercher d'autres produits à évaluer pour remplacer ceux détectés à risque par notre Système d'Information.

Celui-ci est conçu à partir d'une base de données informatique. Nous avons réalisé un démonstrateur opérationnel permettant de valider notre approche. A travers lui, les fournisseurs de COTS peuvent diffuser de l'information utile pour l'intégration de leur produit. Celle-ci permet, au développeur de l'application à base de COTS, de réaliser un prototype et de recueillir plus facilement qu'aujourd'hui les informations utiles aux processus d'évaluation et de sélection des COTS. Nous favorisons ainsi une meilleure maîtrise des différentes étapes du processus de développement et donc on garantit au développeur un choix optimal des COTS ainsi qu'une maîtrise des coûts et des risques.

Mots clés

- Composants COTS (logiciels sur étagères),
- Evaluation,
- Intégration,
- Système d'Information.

Table des Matières

REMERCIEMENTS.....	5
RESUME.....	7
MOTS CLES.....	9
TABLE DES MATIERES	11
LISTE DES FIGURES	17
CHAPITRE 1 : INTRODUCTION.....	21
I PROBLEMATIQUE.....	21
II DEMARCHE ADOPTEE	23
III ORGANISATION DE LA THESE	25
CHAPITRE 2 : ETAT DE L'ART	29
I INTRODUCTION.....	29
II LES COTS	30
II - 1 DEFINITION DES COTS.....	30
II - 2 CLASSIFICATION DES COTS	31

II - 3	LES PROCESSUS DE DEVELOPPEMENT D'UNE APPLICATION A BASE DE COTS.....	34
II - 4	DES DIFFICULTES POUR L'INTEGRATION DES COTS	36
II - 5	LES COTS DANS UNE APPLICATION	39
III	UTILISATION DES COTS	39
III - 1	PRINCIPE D'UTILISATION	39
III - 2	LA SELECTION DES COTS.....	40
III - 3	L'ASSEMBLAGE DES COTS	41
III - 4	GAINS ATTENDUS	42
III - 5	LES DIFFICULTES DE MISE EN ŒUVRE.....	42
III - 6	LA NECESSITE D'UN PROCESSUS DE DEVELOPPEMENT	45
IV	LE PROCESSUS DE DEVELOPPEMENT D'UNE APPLICATION A BASE DE COTS	45
IV - 1	CONTEXTE	45
IV - 2	LA PHASE DE SPECIFICATION DES BESOINS.....	46
IV - 3	LA PHASE D'ACQUISITION ET DE SELECTION	48
IV - 4	LA PHASE DE QUALIFICATION.....	51
IV - 5	LA PHASE D'ADAPTATION DES COTS DANS L'APPLICATION BASEE COTS	53
IV - 6	LA PHASE D'ASSEMBLAGE.....	55
IV - 7	LA PHASE DE MISE A JOUR	56
IV - 8	BILAN	57
V	L'UTILISATION D'UN PROTOTYPE POUR LE RECUEIL D'INFORMATIONS	58
V - 1	LE ROLE DU PROTOTYPE.....	58
V - 2	A TRAVERS LES PHASES DU PROCESSUS DE DEVELOPPEMENT D'APPLICATION A BASE DE COTS	59
V - 2.1	Contexte.....	59
V - 2.2	A partir des besoins	59
V - 2.3	Pour sélectionner des COTS.....	61
V - 2.4	Pour trier des COTS	62
V - 2.5	Pour évaluer et Qualifier des COTS	64
V - 2.6	Pour étudier l'adaptation des COTS	64
V - 3	BILAN	65
VI	CONCLUSION.....	66

CHAPITRE 3 : LES INFORMATIONS POUR L'UTILISATION DES COTS 67

I INTRODUCTION.....	67
II LES INFORMATIONS NECESSAIRES POUR LE PROCESSUS DE DEVELOPPEMENT.....	68
II - 1 INTRODUCTION	68
II - 2 LES DONNEES DISPONIBLES	69
II - 3 LES DONNEES NECESSAIRES	72
II - 3.1 Pour la veille technologique et la définition des besoins	73
II - 3.2 Pour l'acquisition et la sélection des COTS.....	76
II - 3.3 Pour la qualification des COTS	79
II - 3.4 Pour l'adaptation des COTS	80
II - 3.5 Pour l'assemblage des COTS.....	82
II - 3.6 Pour la mise à jour de l'application à base de COTS.....	84
II - 4 LA SYNTHESE SUR LES INFORMATIONS RECUEILLIES	85
II - 5 BILAN	86
III LES INFORMATIONS NECESSAIRES POUR L'INTEGRATION D'UN COTS DANS UN PROTOTYPE	88
III - 1 INTRODUCTION	88
III - 2 LE SCENARIO D'INTEGRATION.....	90
III - 2.1 Un environnement d'exécution du COTS	90
III - 2.2 Un environnement pour l'intégration du COTS	91
III - 2.3 Un environnement d'exécution des outils pour l'intégration	92
III - 2.4 Les connaissances techniques pour réaliser l'intégration	93
III - 3 LES SCENARIOS D'INCOMPATIBILITE.....	94
III - 4 UN RAPPORT DE SYNTHESE.....	95
III - 4.1 Introduction	95
III - 4.2 Un rapport sur les COTS	96
III - 4.3 L'estimation des coûts et des risques	98
III - 4.4 Les comparatifs	100
III - 4.5 Bilan sur le rapport de synthèse.....	100
IV CONCLUSION.....	101

CHAPITRE 4 : LE SYSTEME D'INFORMATION POUR COTS.....	103
I INTRODUCTION.....	103
II LE ROLE DU SYSTEME D'INFORMATION	104
II - 1 DEFINITION DU SYSTEME D'INFORMATION.....	104
II - 2 RELIER LES FOURNISSEURS D'INFORMATIONS ET LES CLIENTS.....	105
II - 3 FAVORISER LE RECUEIL ET LE TRAITEMENT DES INFORMATIONS	106
III LES ACTEURS ET LES INFORMATIONS DISPONIBLES POUR LE SYSTEME D'INFORMATION (SI)	107
III - 1 LES ACTEURS DU SYSTEME D'INFORMATION.....	107
III - 1.1 Le fournisseur et le concepteur de COTS (FourCo).....	107
III - 1.2 L'utilisateur de COTS	108
III - 1.3 La communauté de COTS	108
III - 1.4 La liste des acteurs.....	109
III - 2 LES INFORMATIONS DISPONIBLES.....	110
III - 3 L'AIDE A L'INTEGRATION DES COTS	120
IV LES INFORMATIONS MANIPULEES PAR LE SI.....	122
IV - 1 LA RELATION CLIENT/FOURNISSEUR	122
IV - 2 STRUCTURE D'INFORMATION POUR LA REALISATION DE PROTOTYPES	124
IV - 3 STRUCTURE D'INFORMATION POUR L'INTEGRATION D'UN COTS	126
IV - 4 LA NECESSITE DE CERTIFIER LES INFORMATIONS.....	129
V L'UTILISATION DU SI.....	130
V - 1 PRESENTATION DU SYSTEME D'INFORMATION	130
V - 2 LES FOURNISSEURS D'INFORMATIONS SUR LES COTS	131
V - 3 L'UTILISATEUR DE COTS.....	133
V - 3.1 Dans la phase de veille technologique et de spécification.....	133
V - 3.2 Dans la phase de sélection et d'acquisition	135
V - 3.3 L'activité de réalisation d'un prototype.....	135
V - 3.4 L'activité de capitalisation des informations	137
V - 3.5 Les rôles des rapports de synthèse	137
VI LES DONNEES DU SI.....	140

VI - 1 LES ACTEURS	140
VI - 2 LA CARTE D'IDENTITE ICOTS	141
VI - 3 INFORMATIONS ASSOCIEES A LA CARTE D'IDENTITE.....	142
VII CONCLUSION	143

CHAPITRE 5 : LE DEMONSTRATEUR..... 145

I PRESENTATION 145

I - 1 LES SERVICES RENDUS..... 145

I - 2 LES OBJECTIFS

II UTILISATION DU DEMONSTRATEUR..... 147

II - 1 EN FONCTION DES ACTEURS

II - 2 LA DEMARCHE POUR LES UTILISATEURS DE COTS..... 148

II - 2.1 La récupération des cartes ICOTS

II - 2.2 La description de l'application à base de COTS..... 150

II - 2.3 La description des scénarios

II - 2.4 L'association des services des COTS avec les services de l'application à base de COTS
151

II - 2.5 L'utilisation des rapports de synthèse..... 152

III LES BENEFICES OBTENUS..... 155

IV LES LIMITES ET PERSPECTIVES DU DEMONSTRATEUR..... 156

CHAPITRE 6 : CONCLUSION..... 159

REFERENCES..... 165

ANNEXE..... 169

I EXEMPLE D'UTILISATION DU DEMONSTRATEUR..... 169

I - 1 PRESENTATION

I - 2 LE FOURNISSEUR DE COTS

I - 2.1 Son rôle

I - 2.2	Créer une fiche ICOTS.....	173
I - 3	L'UTILISATEUR DE COTS.....	181
I - 3.1	Présentation des actions principales.....	181
I - 3.2	Décrire l'application à base de COTS ou le prototype.....	182
I - 3.3	Le téléchargement des fiches ICOTS.....	184
I - 3.4	Associer les services de l'application à ceux des COTS à intégrer.....	185
I - 3.5	Accéder aux informations proposées par le démonstrateur.....	186
I - 4	CONCLUSION	191

Liste des figures

<i>Figure 1 : EPIC</i>	34
<i>Figure 2 : Nouveau processus d'après Morisio</i>	35
<i>Figure 3 : les cinq grandes phases du processus de développement à base de COTS</i>	40
<i>Figure 4 : WinWin Spiral Model [Boehm02]</i>	44
<i>Figure 5 : spécification des besoins</i>	47
<i>Figure 6 : Premières étapes du processus de développement basé COTS</i>	48
<i>Figure 7 : Les étapes jusqu'à la qualification et l'évaluation des COTS</i>	51
<i>Figure 8 La phase d'adaptation de COTS dans le processus de développement à base de COTS</i>	53
<i>Figure 9 : La phase d'assemblage dans le processus de développement à base de COTS</i>	55
<i>Figure 10 : processus de développement à base de COTS</i>	57
<i>Figure 11 : Phases du processus de développement nécessitant la réalisation du prototype</i>	65
<i>Figure 12 : Les acteurs de COTS</i>	69
<i>Figure 13 : cas d'utilisation pour obtenir des informations</i>	71
<i>Figure 14 : Extrait des « Données Produit » pour la veille technologique</i>	73
<i>Figure 15 : Extrait des « données produit » dans la phase de définition des besoins</i>	74
<i>Figure 16 : Extrait des « données de spécification »</i>	75
<i>Figure 17 : Familles d'informations de la phase 1</i>	75
<i>Figure 18 : Les informations de la phase d'acquisition et de sélection des COTS</i>	77
<i>Figure 19 : Extrait des « données produit » dans la phase d'acquisition et de sélection</i>	77
<i>Figure 20 : Extrait de « données de spécification » avec l'identification des services disponibles dans le COTS pour l'application</i>	78
<i>Figure 21 : Les informations de la phase de qualification et d'évaluation des COTS</i>	79
<i>Figure 22 : Fiche Données produit dans la phase d'adaptation</i>	80

<i>Figure 23 : Les informations de la phase d'adaptation des COTS.</i>	81
<i>Figure 24 : Itérations au niveau décisionnel</i>	82
<i>Figure 25 : Les informations pour l'assemblage des COTS.</i>	83
<i>Figure 26 : Les familles d'informations en fonction des différentes phases du processus de développement à base de COTS.</i>	85
<i>Figure 27 : Utilisation des familles d'informations dans les phases du processus de développement</i>	87
<i>Figure 28 : Réalisation d'un prototype</i>	88
<i>Figure 29 : Décomposition d'un Système d'Information</i>	104
<i>Figure 30 : Les services de COTS.</i>	107
<i>Figure 31 : les acteurs du SI</i>	109
<i>Figure 32: Première séquence pour l'initialisation du fonctionnement du SI</i>	110
<i>Figure 33 : Logiciels par catégories chez SUN</i>	111
<i>Figure 34 : Liste de produits appartenant à une catégorie chez SUN</i>	112
<i>Figure 35 : Description d'un produit chez SUN.</i>	113
<i>Figure 36 : Configuration requise pour l'exécution d'un produit chez SUN.</i>	113
<i>Figure 37 : Attribut pour identifier les COTS et les services intégrables</i>	114
<i>Figure 38 : Recherche de comment réaliser une intégration avec Excel 2003</i>	115
<i>Figure 39 : Informations sur la manière d'intégrer Office Excel 2003</i>	116
<i>Figure 40 : Problème lié à l'intégration d'Office Excel 2003</i>	116
<i>Figure 41 : Tables décrivant le scénario d'intégration.</i>	117
<i>Figure 42 : Liste de produits classés par système d'exploitation et familles sur le site de « 01 informatique »</i>	118
<i>Figure 43 : Opinions sur un produit proposé sur « Télécharger.com »</i>	119
<i>Figure 44 : Echange d'information entre acteurs.</i>	123
<i>Figure 45 : Informations nécessaires pour la réalisation d'un prototype</i>	125
<i>Figure 46 : Tables décrivant le scénario d'exécution</i>	125
<i>Figure 47 : Informations nécessaires pour l'intégration d'un COTS.</i>	126
<i>Figure 48 : Tables décrivant l'environnement d'intégration.</i>	127
<i>Figure 49 : L'ensemble des informations pour intégrer un COTS dans un prototype</i>	128
<i>Figure 50 : Séquence de certification de l'information par "FourCo"</i>	130
<i>Figure 51 : Echange d'information entre acteurs.</i>	131
<i>Figure 52 : Exemple de carte d'identité de COTS (ICOTS).</i>	132

<i>Figure 53 : Utilisation du SI dans la phase de veille technologique par les utilisateurs.....</i>	<i>134</i>
<i>Figure 54 : Utilisation du SI dans la phase de spécification des besoins par les utilisateurs</i>	<i>134</i>
<i>Figure 55 : Utilisation du SI dans la phase d'acquisition et de sélection par les utilisateurs.....</i>	<i>135</i>
<i>Figure 56 : Utilisation du SI dans le cadre de la réalisation du prototype.....</i>	<i>136</i>
<i>Figure 57 : Exemple de rapport montrant une incompatibilité.....</i>	<i>138</i>
<i>Figure 58 : Exemple d'information sur l'environnement d'intégration.....</i>	<i>139</i>
<i>Figure 59 : Les tables pour gérer les acteurs</i>	<i>140</i>
<i>Figure 60 : Les tables permettant d'identifier un COTS.....</i>	<i>141</i>
<i>Figure 61 : Les tables pour l'intégration d'un prototype</i>	<i>142</i>
<i>Figure 62 : Les principales étapes d'utilisation du démonstrateur</i>	<i>149</i>
<i>Figure 63 : Rapport sur les environnements d'exécution</i>	<i>152</i>
<i>Figure 64 : Rapport sur les moyens d'intégration d'un COTS.....</i>	<i>153</i>
<i>Figure 65 : Rapport sur les difficultés et dysfonctionnements</i>	<i>154</i>
<i>Figure 66 : Ecran principal du démonstrateur.....</i>	<i>170</i>
<i>Figure 67 : Arborescence du SI.....</i>	<i>171</i>
<i>Figure 68 : Dictionnaire des Icônes</i>	<i>172</i>
<i>Figure 69 : Les menus du démonstrateur.....</i>	<i>173</i>
<i>Figure 70 : Création de la fiche produit Outlook</i>	<i>174</i>
<i>Figure 71 : Créer la version d'un produit</i>	<i>174</i>
<i>Figure 72 : Ajout de l'environnement d'exécution</i>	<i>175</i>
<i>Figure 73 : Ajout d'un OS dans l'environnement d'exécution.....</i>	<i>176</i>
<i>Figure 74 : Ajout de services dans la fiche ICOTS.....</i>	<i>176</i>
<i>Figure 75 : Ajout d'un scénario d'exécution associé à un service de l'application</i>	<i>177</i>
<i>Figure 76 : Ajout d'un scénario d'intégration associé à un service.....</i>	<i>178</i>
<i>Figure 77 : Ajout du scénario d'Intégration.....</i>	<i>179</i>
<i>Figure 78 : Ajout d'un produit nécessaire pour l'environnement de développement.....</i>	<i>179</i>
<i>Figure 79 : Ajout de l'outil de développement.....</i>	<i>180</i>
<i>Figure 80 : Ajout du langage de programmation</i>	<i>180</i>
<i>Figure 81 : Menu Général du démonstrateur</i>	<i>181</i>
<i>Figure 82 : Fiche IPPOP avec la liste des services.....</i>	<i>182</i>
<i>Figure 83 : Ajout du scénario d'exécution d'IPPOP.....</i>	<i>183</i>

<i>Figure 84 : Outil de développement pour l'intégration des COTS</i>	183
<i>Figure 85 : Téléchargement d'une fiche ICOTS</i>	184
<i>Figure 86 : Association de services</i>	185
<i>Figure 87 : Choix des scénarios d'intégration pour les rapports</i>	186
<i>Figure 88 : Résultat de la synthèse pour l'environnement d'exécution</i>	187
<i>Figure 89 : Résultat du démonstrateur sur les moyens d'intégrations</i>	188
<i>Figure 90 : Exemple d'un rapport de dysfonctionnement</i>	189
<i>Figure 91 : Deuxième exemple de rapports de dysfonctionnement</i>	190

Chapitre 1 : Introduction

I Problématique

Pour des raisons économiques manifestes, les développeurs informatiques sont obligés d'utiliser des applications et/ou des briques d'applications déjà existantes. Ce phénomène n'est pas nouveau mais il prend de plus en plus d'ampleur. Les délais de réalisation réduits et la complexité des applications sont souvent des arguments à l'origine de cet engouement pour l'intégration des entités logicielles préexistantes, qu'elles soient assez générales ou proches du sur-mesure. Les entreprises veulent réduire leurs temps de développement en utilisant un produit déjà testé, validé et déjà en service, réduire leurs coûts de création de nouvelles applications par l'acquisition de produits moins chers que leurs coûts de développement unitaire. Elles veulent aussi recentrer leurs compétences autour du métier principal de l'entreprise via la personnalisation d'éléments logiciels préfabriqués dont l'aptitude à être adaptés et intégrés est importante.

L'utilisation croissante de ces composants logiciels du commerce appelés COTS¹ à la place de développements en interne s'impose dans la réalisation de la quasi-totalité des applications logicielles. En contrepartie des bénéfices attendus, l'utilisation de ces composants s'accompagne d'une perte non négligeable de la maîtrise des systèmes dans lesquels ils sont utilisés et d'une dépendance forte à l'égard des éditeurs de COTS.

Afin de réduire cette perte de maîtrise, il est indispensable d'entretenir une connaissance soutenue du marché des composants et de ses évolutions (acteurs et produits). La collecte

¹ Components Off-The-Shelf

d'informations qui est ainsi nécessaire et les efforts à consentir pour maintenir un niveau correct des informations mises à jour, correspond à un investissement lourd auquel doivent absolument faire face tous les développeurs et intégrateurs de COTS.

Pourtant les informations sont en partie accessibles auprès des éditeurs de COTS et dans la communauté des utilisateurs de ces produits. Elles sont souvent de qualités inégales et insuffisantes. Pourtant, la communauté des COTS détient la plupart du temps une foule d'informations issues des évaluations des produits et des retours d'expériences dans des intégrations. Il serait intéressant de les mettre à disposition de l'ensemble de la communauté des utilisateurs de COTS qui cherchent à collecter des informations pour des développements présents ou futurs.

L'intégration des composants logiciels sur étagère dans une application nécessite l'utilisation d'un processus de développement assis sur une démarche permettant de les sélectionner, de les évaluer et de les comparer. Ainsi, nous devons les départager, les écarter si nécessaire ou les retenir en fonction des moyens et des compétences nécessaires à leurs intégrations. Quelque soit le processus choisi, la normalisation des informations est primordiale pour la réussite de cette démarche allant d'une sélection de nature stratégique des COTS aux spécifications de développement pour intégrer ces produits particuliers.

Ce sont pourtant ces informations qui sont souvent à l'origine des difficultés rencontrées pour la mise en œuvre d'un développement à base de COTS. Ce sont d'abord la maîtrise des coûts d'intégration qui sont difficiles à établir et qui dépendent énormément des connaissances que peuvent avoir les équipes de développement sur la manière d'intégrer les produits. A cela s'ajoute les compétences des équipes de développement dans la maîtrise des techniques d'intégration des COTS sélectionnés. Par la suite, il y a aussi les risques encourus sur des dysfonctionnements lors de l'intégration ou des difficultés lors des phases d'assemblage. Dans tous les cas, plus les informations permettent aux équipes de s'assurer d'une bonne connaissance des fonctionnalités des COTS (mais aussi de leurs limites et incompatibilités) et des mécanismes d'intégration (les environnements cible notamment) dans les phases de développement, plus il est possible d'évaluer et de maîtriser correctement les risques et les coûts.

Pour ce faire, au sein du processus de développement, et en particulier dans le cadre des évaluations, les développeurs réalisent des prototypes leur permettant de recueillir les informations manquantes et indispensables à un bon déroulement d'une conception d'application basée COTS. Cependant, ces mêmes prototypes souffrent de difficultés sur le manque d'informations. En général leurs réalisations peuvent devenir longues et fastidieuses.

Il est donc indispensable de recueillir plusieurs connaissances autour des COTS pour faciliter leurs utilisations dans les développements d'applications logicielles. C'est par exemple les environnements d'utilisation, leurs fonctionnalités, les environnements

d'intégration, les techniques et moyens d'intégration, les limites de fonctionnement ou les incompatibilités d'utilisation avec d'autres produits. La mise en commun de toutes ces informations issues des retours d'expérience de la part d'une communauté de COTS permettrait d'alimenter cette base de connaissance. Elle serait un plus pour la récupération et l'accès aux informations. Cet enrichissement ne peut qu'être favorable à l'utilisation des COTS et à la maîtrise des coûts et des risques.

II Démarche adoptée

Dans cette optique, ce travail porte sur l'aide que nous pouvons apporter aux développeurs dans la mise à disposition de systèmes d'informations supports au recueil de données pour l'évaluation des COTS.

Le chapitre de l'état de l'art nous a ainsi permis d'identifier les produits logiciels assimilés à des COTS et leurs utilisations. Dans ce cadre là, nous considérons tous les produits logiciels qui sont intégrables. D'après la littérature, quelque soit l'appartenance d'un COTS à une classification, la manière d'être intégré ou la façon dont il est conçu, le développement d'une application à base de COTS doit suivre impérativement un processus de développement spécifique. Cela nous semble nécessaire pour réduire les difficultés identifiées à ce jour pour leur utilisation. Parmi les difficultés, l'une d'entre elle retiendra notre attention : celle de pouvoir recueillir les informations et leurs validations. Pour cela, de nombreux auteurs nous proposent l'utilisation de prototypes permettant l'évaluation des COTS et le recueil d'informations manquantes. La solution nous semble intéressante car l'expérimentation donne toujours de bons résultats en termes de recueil d'informations. Par contre, la difficulté de réaliser des prototypes d'applications, les moyens envisagés, la description d'une méthode rigoureuse pour y arriver, sont peu traités dans la littérature.

Pour pallier à de tels problèmes, nous avons donc recherché les informations manipulées au travers des différentes phases du processus de développement et spécifiquement celles pour la réalisation d'un prototype. Nous avons identifié des informations communes comme l'environnement d'exécution du COTS (système d'exploitation, services et composants requis comme par exemple une librairie multimédia, des ressources réseau, base de données). Cette liste n'est pas exhaustive mais permet de poser la question suivante : combien d'environnements, combien d'informations sont nécessaires pour développer le prototype intégrant des COTS et pour le faire fonctionner ?

L'étude des différents cas industriels que nous avons vécus, nous a permis d'établir une liste d'environnements nécessaires pour le fonctionnement et le développement du prototype à base de COTS. Dans le cadre de nos développements industriels il est peu

probable de trouver plusieurs scénarios pour intégrer un même COTS. Nous avons donc recherché sur les sites Internet dédiés aux développeurs la possibilité d'identifier plusieurs scénarios d'intégration pour le même COTS. C'est par exemple le cas pour l'intégration d'un COTS comme Microsoft Project. En fait, nous proposons que chaque environnement permettant de décrire une utilisation de COTS (pour le fonctionnement ou pour l'intégration) doit pouvoir être défini de manière multiple pour chaque COTS. Le système que nous proposons doit pouvoir établir tous les cas de figures d'utilisation et d'intégration envisageables.

Il reste maintenant, pour réduire la difficulté de recueil d'informations, la mise en place d'une solution idoine. Pour cela nous estimons que les informations sur les COTS doivent être diffusées, puis manipulées par les développeurs. La diffusion permettra l'enrichissement de l'information associée à chaque COTS. La manipulation de ces informations favorisera l'étude sur les différents choix possibles d'assemblage des COTS. Elle permettra aussi de réaliser des synthèses notamment sur les difficultés prévisibles comme le conflit entre deux environnements de COTS ou pour déterminer les moyens nécessaires au développement du prototype. De ce fait, l'ensemble de ces informations doit pouvoir être accessible par une communauté de développeurs d'applications à base de COTS. De plus, ces informations doivent pouvoir être échangées et complétées en fonction de l'historique des évaluations faites sur les COTS. Pour ce faire, nous préconisons et caractérisons dans ce mémoire un Système d'Information et ses propriétés car nous pensons qu'il satisfait les besoins exprimés ci-avant.

Nous proposons à ce titre et dans un souci de fournir un système opérationnel et tangible, un démonstrateur qui permet la saisie des informations sur les COTS dans le but de réaliser un prototype. C'est un guide sur l'ensemble des informations à saisir permettant de décrire les éléments qui nous ont semblé importants pour aider le développeur dans sa réalisation d'un prototype. Parmi eux nous pensons que le fait d'associer à chaque COTS un ou plusieurs scénarios de dysfonctionnement aidera le développeur dans le choix des scénarios d'intégration. Nous proposons dans notre travail de regrouper ces informations sous forme de trois rapports. Le premier (l'environnement d'exécution) permet d'avoir une vision de l'ensemble des produits logiciels nécessaires à l'utilisation des COTS. Le second (l'environnement de développement) renseigne sur les outils utilisables pour l'intégration et apporte une aide sur le développement de l'intégration des COTS par l'intermédiaire d'une série de documents (exemples de codes, principes de fonctionnement, ...). Il est possible de les associer pour chaque COTS, mais aussi à chacun des services qu'assure le COTS. Enfin le troisième rapport (les dysfonctionnements) traite des conflits éventuels entre différents produits logiciels nécessaires aux fonctionnements du prototype et de ses COTS ou à l'intégration du COTS dans le prototype. Nous pensons pouvoir alerter le développeur sur de possibles difficultés. Ce dernier résultat est une information indispensable pour l'évaluation. De ce fait, le démonstrateur contribuera aussi dans le recueil d'informations indispensables à cette phase du processus.

Dans un premier temps les informations sur les COTS sont établies par les fournisseurs de COTS. Celles-ci sont souvent insuffisantes pour la réalisation de tous les prototypes envisagés par les utilisateurs des COTS. C'est donc l'expérimentation et la réalisation de ces prototypes qui permettra d'établir des scénarios de fonctionnement ou d'intégration non renseignés dans la fiche ICOTS. Il nous semble donc important de pouvoir récupérer ces compléments d'informations. C'est donc pour cela que dans l'utilisation du démonstrateur nous avons prévu d'établir des relations entre les différents acteurs du Système d'Information (les fournisseurs de COTS, la communauté de COTS et les utilisateurs de COTS). Chaque réalisation d'un prototype permet de recueillir des informations qui peuvent être utiles pour une future intégration de COTS. Il nous semble important de capitaliser les informations recueillies en complétant la fiche de chaque COTS. L'enrichissement de toutes ces informations par le biais de la communauté des COTS et des expérimentations déjà réalisées est une nécessité pour accélérer le recueil d'informations. Elle permettra de réduire certaines difficultés présentées lors du chapitre de l'état de l'art : la pauvreté des informations et leur inaccessibilité.

III Organisation de la thèse

Cette thèse est organisée en 4 parties. La première est un état de l'art sur les COTS, le processus de développement d'une application à base de COTS et sur l'utilisation des prototypes dans le cadre de ce processus. Nous poursuivons l'étude dans un deuxième chapitre par le recueil d'informations utiles pour le processus de développement et la réalisation des prototypes. A partir de cette étude orientée vers les besoins du prototype, nous présentons dans une troisième partie notre approche basée sur l'utilisation d'un Système d'Information des COTS. Enfin, nous terminons ce travail par la présentation d'un démonstrateur qui validera notre approche.

Dans l'état de l'art nous identifions les logiciels intégrables et leur utilisation dans le cadre d'un développement d'une application à base de COTS. La recherche des différentes utilisations des COTS, des processus de développement et de leurs problématiques doivent nous conduire à sélectionner notre axe de recherche : **l'aide aux développeurs dans la réalisation de prototypes à base de COTS.**

Celui-ci sera approfondi dans le deuxième chapitre consacré aux informations utilisables par les COTS. Dans ce chapitre nous confirmerons notre approche par l'étude des

différentes informations manipulées pendant les différentes phases du processus et en particulier pour la réalisation d'un prototype. Cependant, l'identification de ces informations n'est pas suffisante pour aider le développeur. Il faut en plus, établir le moyen d'y accéder, de les manipuler et de les enrichir. Une solution est proposée : le système d'Information de COTS. Cette solution est décrite dans le chapitre suivant et correspond à notre contribution.

La troisième partie permet de présenter notre solution (le Système d'Information de COTS) en intégrant les informations manipulées et étudiées dans le second chapitre de la thèse. Notre contribution est présentée en commençant par la justification de l'utilisation d'un Système d'Information comme étant une solution adaptée à notre problématique et en vérifiant son adéquation avec les besoins identifiés dans le chapitre précédent. A partir de là, nous nous efforçons de présenter les acteurs, les données et l'utilisation du Système d'Information correspondant à notre proposition. Les acteurs et les données sont déduits des résultats sur les informations utiles pour la réalisation des prototypes identifiés dans le chapitre précédent. Cela se traduit par la présentation d'une structure de donnée de type entité/association qui préfigure la réalisation du prototype nous permettant de supporter notre approche. L'utilisation du Système d'Information tient compte de la démarche de saisie nécessaire à la création des fiches ICOTS, du rôle de chaque acteur dans le renseignement ou dans l'exploitation des fiches et, enfin, de la manière de décrire l'intégration des COTS dans l'application à base de COTS.

Le quatrième chapitre de la thèse est consacré à la description de l'outil qui sert de support à notre proposition. Nous décrivons son utilisation en présentant les écrans et des exemples d'informations manipulées par le Système d'Information. Ces derniers correspondent aux informations identifiées dans la deuxième partie de la thèse. Nous décrivons les trois rapports de synthèses présentés dans l'outil et correspondant à nos travaux : l'environnement d'exécution, l'environnement de développement et les dysfonctionnements. Nous établissons à partir de cette présentation les bénéfices attendus de son utilisation : l'aide à la saisie des informations utiles pour l'aide à la réalisation d'un prototype, la possibilité de capitaliser des informations associées à chaque COTS ou services, l'identification des moyens pour l'intégration et l'utilisation des COTS, la détection des dysfonctionnements déjà identifiés par d'autres utilisateurs de COTS.

Dans la conclusion de ces travaux, nous indiquons que notre proposition va faciliter la réalisation des prototypes à bases de COTS. Elle permet notamment l'identification des moyens d'exécution et d'intégration des COTS en considérant l'ensemble des produits logiciels nécessaires. Cela conduira aussi à la détection de dysfonctionnements de manière plus efficace. Tout cela contribuera dans la qualité des évaluations. Le gain de temps et la facilité d'obtention des informations dans le cadre de la réalisation du prototype permettront la réalisation d'un nombre plus important d'expérimentations et donc un recueil plus efficace d'informations qui contribueront aussi dans la qualité des évaluations, notamment par l'enrichissement des fiches ICOTS. L'intérêt de cette proposition ne doit pas s'arrêter à l'aide pour la réalisation de prototype à base de COTS, mais elle doit pouvoir s'étendre à

une utilisation plus globale dans le cadre des autres phases du processus de développement. Nous pensons à une aide à la saisie d'informations relatives aux phases de sélections et d'évaluations mais aussi aux traitements des informations pour les estimations des risques et des coûts avec une capitalisation de toutes ces informations.

Chapitre 2 : Etat de l'art

I Introduction

L'utilisation des COTS dans les développements logiciels est apparue comme une solution pour résoudre un certain nombre de problèmes (Qualité des logiciels, délais de réalisation trop long, la nécessité de maîtriser des domaines métiers, ...) liés à la complexité croissante des applications développées. Des critères de qualité de plus en plus exigeants ont aussi présidé à l'émergence des logiciels basés COTS et bien entendu la nécessité de baisser les coûts de façon drastique. En fait, en utilisant les COTS, les développeurs prévoient d'obtenir notamment des gains sur les temps de développement, une meilleure qualité des produits réalisés et une meilleure accessibilité aux nouvelles technologies.

Cependant, comme nous le verrons dans cet état de l'art, l'utilisation des COTS a généré d'autres problèmes dans sa mise en œuvre. Dans la littérature on parle volontiers du besoin d'avoir un processus spécifique pour les développements d'applications à base de COTS. En fait, le choix, l'évaluation des COTS et l'estimation des risques et des coûts semblent être une réelle difficulté.

Nous allons donc aborder cet état de l'art en se demandant comment aider les développeurs à base de COTS. Cela doit nous permettre d'identifier un axe d'investigation et de recherche sur le développement logiciel sur étagère en général, et sur le besoin d'environnements idoines (processus, outillage) en particulier. Pour ce faire, nous avons voulu prendre en compte la totalité des logiciels qu'ils soient conçus ou pas pour être réutilisables.

La nécessité d'utiliser un processus de développement spécifique est devenu rapidement une évidence. Les difficultés sur la sélection et l'évaluation des COTS correspondant à des phases du processus de développement, nous permettent d'identifier un problème récurrent sur l'accessibilité et le manque d'informations des COTS. Ce problème nuit gravement à leur utilisation à grande échelle et de façon sûre et systématique. Nous étudierons en particulier la solution qui consiste à développer des prototypes intégrant des COTS pour le recueil d'informations.

II Les COTS

II - 1 Définition des COTS

Un COTS est un composant issu du marché ou plus communément appelé un composant sur étagère. Plusieurs auteurs ont proposé ou compilé des définitions [Basili01, Brownsword00, FAR96, Heineman01, Obendorf00, Obendorf97, Vidger97, Vigder96]. De fait, tous ces travaux ont permis de construire une définition des COTS.

Un composant issu du marché est :

- un produit logiciel existant en de multiples copies,
- dont le code source est disponible ou non,
- vendu, loué ou fourni gratuitement par un vendeur,
- ayant des mises à jour périodiques s'accompagnant d'un accroissement des fonctionnalités fournies alors que certaines autres deviennent obsolètes.

Cette définition est suffisamment ouverte pour que beaucoup de produits logiciels puissent être considérés comme des COTS. Pourtant, si on est d'accord sur cette définition reprise majoritairement par différents acteurs de la communauté des COTS, il faudrait la compléter avec un critère qui semble essentiel : **la possibilité d'intégration**. Un COTS doit pouvoir impérativement être utilisé dans le cadre du développement d'un ensemble logiciel répondant à un certain nombre de services ou fonctionnalités. Il doit avoir la possibilité d'apporter à cet ensemble un ou plusieurs de ses services. Ces derniers y sont intégrés, et apparaissent comme un des multiples services proposés par l'application à base de COTS.

De ce fait, nous proposons de modifier la définition d'un COTS comme présenté ci-dessous en ajoutant en gras notre vision.

Un composant issu du marché est :

- un produit logiciel existant en de multiples copies,
- dont le code source, parfois accessible sous forme « Open Source », est le plus souvent indisponible,
- vendu, loué ou fourni gratuitement par un vendeur,
- ayant des mises à jour périodiques s'accompagnant d'un accroissement des fonctionnalités fournies alors que certaines autres deviennent obsolètes.
- **un produit ayant la capacité de proposer l'un de ses services (ou plus) comme partie indispensable au fonctionnement d'une autre application.**

II - 2 Classification des COTS

Avec la définition des COTS, il existe dans la communauté des terminologies communément utilisées permettant d'identifier ou de classer les COTS.

Nous réunissons ici, d'après nos observations, les quatre dénominations principalement utilisées :

- « COTS System » ou « COTS Based Product »,
- « COTS Software »,
- « COTS Product »,
- « COTS Components ».

Lorsqu'on parle de « COTS system » les auteurs de ces articles [Carney97, Morisio00] considèrent le COTS comme un ensemble logiciel utilisant des COTS. Dans ce cas, ce n'est pas le produit de manière unitaire que l'on étudie mais le système dans son intégralité.

Le « COTS Software » permet d'identifier l'ensemble des produits. Bien des articles traitent, sous cette dénomination, de la sélection et de la comparaison des produits. Le « COTS Software » est de notre point de vue, une vision plus restrictive que le « COTS

system ». Si ce dernier correspond à une vision d'ensemble comprenant l'application à base de COTS, le « COTS software » ne s'intéresse qu'aux différents COTS indépendamment de l'application à développer.

Une autre restriction dans la vision des COTS correspond à la troisième dénomination « COTS product » qui s'attarde sur la définition d'un produit et les problématiques liées à son intégration dans l'application. Dans ce cadre là, la relation, la comparaison et les contraintes entre produits ne sont pas abordées comme précédemment. Seule les qualités des produits, les fonctionnalités ou les questions techniques pour l'intégration et le développement sont abordées.

Pour finir, les « COTS components » rassemblent une catégorie de COTS construits autour des techniques de composants logiciels. C'est une vue particulière des « COTS product » autour de composants prévus pour être intégrés dès leur conception.

A travers ces différentes dénominations, il apparaît un premier niveau de classification permettant d'identifier le périmètre d'étude des COTS du plus grand avec les « COTS software » au périmètre le plus spécifique des « COTS components ».

A ces dénominations recueillies dans la littérature, on complète la classification par les trois systèmes à base de COTS que Carney et Morisio ont identifié dans [Carney03] et [Morisio00] :

- « Les systèmes clés en main » construits à partir d'un produit que le développeur personnalise sans en modifier son fonctionnement (ex de la suite office de Microsoft),
- « les systèmes intermédiaires » utilisent un produit central (ex Système de Gestion de Bases de Données) autour duquel le développeur ajoute ses propres fonctionnalités ou d'autres composants.
- « Les systèmes intégrés » composés d'un assemblage de plusieurs COTS.

Cette classification est plus orientée technologie ou intégration. Contrairement à la précédente qui définit un périmètre d'étude, celle-ci s'applique à différencier les méthodes d'utilisation et d'intégration des COTS.

Dans le cadre des systèmes clés en main, les fournisseurs de COTS proposent des outils et des moyens permettant de réaliser des développements complémentaires, des modifications d'Interfaces Homme Machine (IHM) et des ajouts de fonctionnalités tout en utilisant celles proposées par le système lui-même. C'est souvent autour d'un langage de programmation spécifique au produit que les adaptations sont possibles. Dans ce cadre là, seules certaines parties d'IHM et certains services sont utilisables pour l'adaptation. Leur nombre est très variable d'un produit à l'autre et la manière de réaliser ces modifications

sont souvent spécifiques au produit. Au final, l'application utilisera les IHM et les services mais aussi toutes les adaptations apportées à ce produit clé en main.

Dans le cadre des systèmes intermédiaires, le produit central propose peu ou pas d'IHM pour l'utilisateur de l'application. Cela sous-entend que les IHM doivent être réalisées en utilisant les techniques et les méthodes permettant d'utiliser les services (de ce système intermédiaire) dans le développement de l'application. Dans le cadre d'une base de données, les techniques d'accès et de traitement des informations sont spécifiques à chaque produit. Néanmoins, il existe des standards comme l'ODBC (Open Data Base Connectivity) qui garantissent pour les développeurs une unité dans les procédures d'accès et de modifications des données, et cela quelque soit le fournisseur de système de bases de données.

Enfin, dans le cadre des systèmes intégrés, il est possible de construire une application à partir de différentes briques logicielles communiquant entre elles. L'application, ainsi créée, est un ensemble cohérent avec des informations bien souvent partagées par les différentes applications la constituant. Ces dernières doivent respecter le formalisme propriétaire du système intégré autorisant l'assemblage cohérent des différentes briques entre elles.

En diminuant la granularité des systèmes à base de COTS, on a la classification communément acceptée par tous et qui décrit l'accessibilité du code source des COTS. Il en existe trois sortes :

- « White Box », dont le code source est accessible et peut être modifié en vue de son intégration dans l'application à base de COTS,
- « Grey Box », dont le code source n'est pas modifiable, mais le produit propose ses propres interfaces de programmation et un langage pour s'interfacer avec une autre application (API ou Application Program Interface),
- « Black Box », où seulement la forme exécutable binaire du produit existe mais sans API.

Cette classification permet d'identifier les produits en définissant les approches d'intégration à mettre en œuvre en fonction de chaque catégorie. Dans le premier cas, la modification du code permet de réaliser toutes les adaptations voulues mais, en contre partie, elle nécessite une maintenance importante de l'ensemble du code. C'est l'inverse dans les autres cas où le code n'est pas modifié et les possibilités d'adaptation dépendent des produits. Par contre, il est nécessaire pour les équipes de développement de se former aux techniques spécifiques pour l'adaptation. En fait chaque catégorie apporte son lot d'avantages et d'inconvénients qui seront pris en compte dans certaines phases du

processus de développement de l'application à base de COTS. C'est le cas par exemple dans la phase d'estimation des risques et des coûts.

De la même manière, la classification de Carney et Morisio [Carney03, Morisio00] aura une incidence sur la phase d'adaptation. En fonction de l'appartenance du COTS à une catégorie, les techniques d'assemblages ne seront pas les mêmes. On retrouve ce lien entre les phases du processus de développement et la classification sous forme de dénomination des COTS. En effet, les articles sur les « COTS software » traitent plus particulièrement de sujets relatifs aux traitements de la phase de sélection et de leurs problématiques (éventuellement sur les autres phases du processus).

Quelque soit la classification des COTS, on peut considérer que les processus de développement spécifiques aux applications à base de COTS, doivent permettre de traiter les différentes catégories de COTS et leurs difficultés d'intégrations. C'est souvent par le biais du processus de développement qu'on essaye d'aider les développeurs dans le traitement des différentes informations et difficultés pour la réalisation d'une application à base de COTS. Dans ces processus, il existe des phases de validation et de contrôle, avec des possibilités de revenir sur une phase précédente. De plus chaque phase a un rôle bien précis qui délimite le travail à faire et guide de ce fait le développeur.

II - 3 Les processus de développement d'une application à base de COTS

Plusieurs articles proposent des processus de développement spécifiques à une application à base de COTS. C'est à partir des difficultés à utiliser le processus traditionnel de développement que ces processus sont apparus. Par exemple, dans EPIC (Evolutionary Process for Integrating COTS), Albert propose une évolution de ce processus [Albert02] présenté dans la Figure 1.

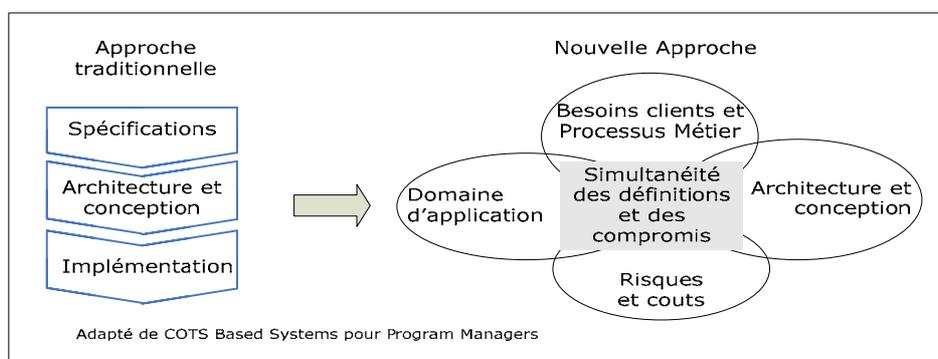


Figure 1 : EPIC

D'après l'auteur, l'approche traditionnelle en mode séquentiel (l'une après l'autre) des différentes étapes du processus ne peut pas être utilisée dans le cadre d'un développement d'application à base de COTS. Il faut obligatoirement utiliser un mode asynchrone (en parallèle).

En fait, cela est imposée par rapport aux difficultés rencontrées sur les COTS : entre autres sur l'absence de maîtrise de l'évolution des COTS (notamment sur les services fournis) ou sur des problèmes d'interopérabilité (avec l'environnement d'utilisation ou avec d'autres produits). Les phases du processus sur l'estimation des risques et des coûts sont là pour maîtriser ces difficultés. Elles ont besoin des informations d'autres phases du processus comme présentées dans l'approche EPIC et en même temps elles peuvent remettre en cause les travaux menés dans les autres phases du processus. C'est la simultanéité des définitions et des différences qui pilote l'ensemble. Cette simultanéité des spécifications et des développements montre une complexité accrue du traitement des informations permettant de faire avancer ce processus.

Pour répondre à ces préoccupations, une autre représentation est utilisée et acceptée par tous dans la littérature. Elle décrit ce processus et décompose en plusieurs étapes le cycle de vie d'une application à base de COTS [Morisio02]. Elle matérialise en plus la relation entre le vendeur de produits et le consommateur ou développeur qui assemblera les COTS dans une application comme le montre la Figure 2.

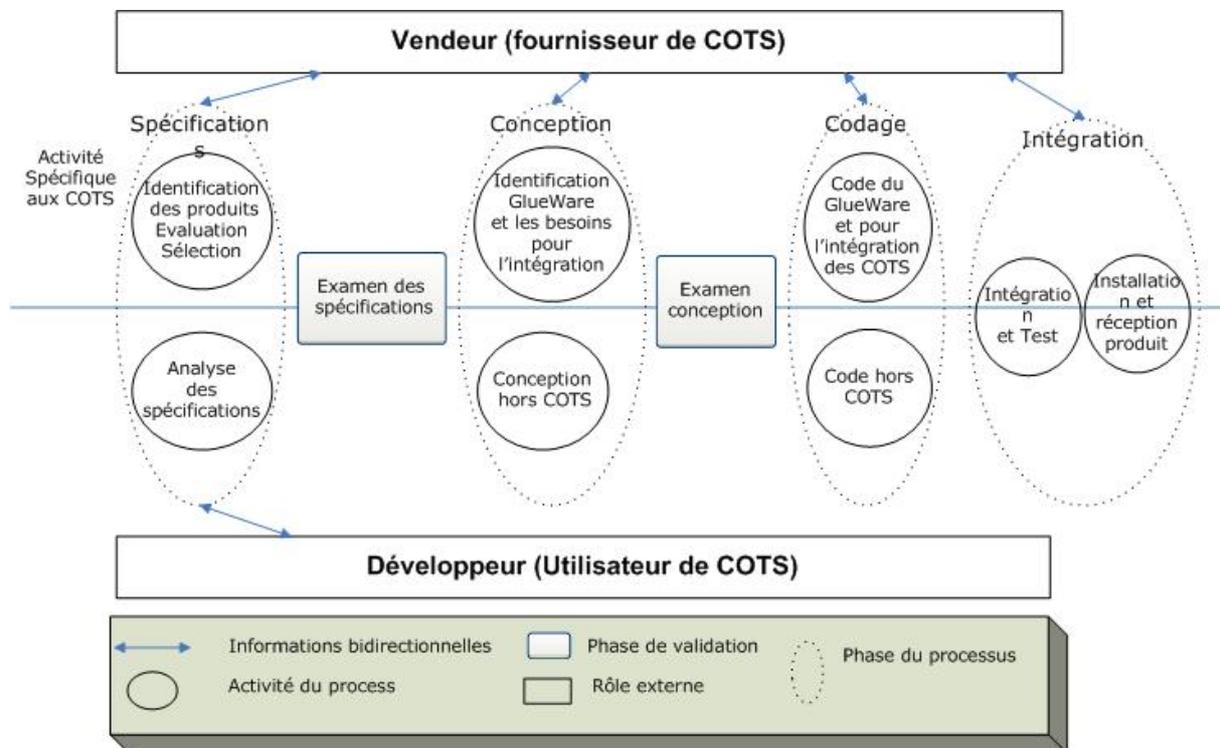


Figure 2 : Nouveau processus d'après Morisio

A partir de ce schéma d'autres auteurs ont montré le caractère itératif de la démarche. En effet dans chaque phase présentée dans [Morisio02], les candidats à l'intégration peuvent être remis en cause. Cela entraîne la reprise partielle ou complète du processus de développement. Parmi les causes de reprise du processus, on note la présence de produits ne correspondant pas aux besoins ou ne pouvant pas être assemblés comme prévus. La détection de ces difficultés peut se faire dans toutes les étapes de ce processus, plus elle se fera tardivement et plus cela pénalisera l'intégration. Ces difficultés dues aux mauvais choix de COTS représentent un risque d'échec pour le bon développement de l'application. D'ailleurs, dans EPIC [Albert02], l'origine de la notion de risque, considérée comme importante par d'autres auteurs, est présentée comme la conséquence de ces mauvais choix.

II - 4 Des difficultés pour l'intégration des COTS

Certaines difficultés pour l'intégration des COTS ne sont pas identifiables facilement du fait que le code source est indisponible, que les vendeurs ont souvent la maîtrise des produits sans en donner le contrôle, que les informations fournies par les vendeurs sont souvent insuffisantes.

Le fait que le code soit indisponible ne permettra pas d'établir une évaluation complète des qualités du produit. Et pourtant c'est un critère qui est nécessaire pour la sélection des COTS comme on le verra plus tard dans ce document par la description d'un processus de développement.

Les mises à jour périodiques traduisent une évolution des fonctionnalités que le développeur d'applications à base de COTS ne peut maîtriser puisqu'il est l'œuvre du vendeur. Une perte de fonctionnalités dans une version future du COTS pourrait remettre en cause l'utilisation du produit dans l'application qui l'utilise. Le développeur de cette application à base de COTS ne peut pas intervenir sur le vendeur pour contrôler les évolutions. Cela nécessitera forcément des évolutions de l'application pour intégrer la fonctionnalité manquante ou pour le remplacement du COTS devenu défaillant et ne permettant plus d'assurer la totalité des services de l'application à base de COTS. De ce fait, une simple mise à jour d'un COTS peut remettre en cause le fonctionnement de l'application. La détection de ces dysfonctionnements ou la perte de fonctionnalités se fait par la récupération des informations auprès du vendeur (qui aura tendance à ne pas fournir d'informations sur des pertes de fonctionnalités et proposera plutôt les nouveautés de son produit) et par l'analyse et l'évaluation des nouvelles versions. En fait, les nouvelles versions des COTS ne garantissent pas le même niveau de qualité de l'application. De ce

fait, un nouveau processus de validation, voire d'adaptation, doit être réalisé. Malheureusement le client d'une application peut très bien mettre à jour un COTS sans l'avis du fournisseur l'intégrant dans son application. C'est, par exemple, le cas d'un COTS partagé par plusieurs applications. L'une d'elles, lors d'une mise à jour modifie la version du COTS qu'elle utilise et de ce fait l'autre application devient instable pour cause d'incompatibilité sur la version de COTS employée. Les aspects analyse et validation du COTS n'étant pas faites, la garantie de fonctionnement et de qualité de l'application ne peuvent être assurée.

Le fait que le produit soit intégrable doit permettre une cohabitation avec l'ensemble des produits constituant l'application à base de COTS. Les conditions d'utilisations des COTS sont par exemple un des pré-requis à l'intégration. Dans certains cas d'incompatibilité entre produits, comme sur les conditions d'utilisation, cette interopérabilité peut ne pas être assurée (même dans des cas de mise à jour de COTS).

Ainsi, nous pouvons dire des COTS :

- que les contrôles sur les fonctionnalités ou la performance du produit logiciel ne peuvent être assurés en totalité,
- qu'aucun contrôle ne peut être assuré sur l'évolution du composant,
- que certains COTS, ne sont pas forcément conçus pour inter-opérer avec d'autres, contrairement à leur vocation,
- que les comportements des distributeurs/éditeurs de composants de type COTS sont imprévisibles.

A partir de cette liste, nous pouvons prédire des difficultés sur la sélection des COTS, le développement et l'intégration dans l'application à base de COTS. Les processus de développement essaient de proposer des méthodes et des moyens permettant d'aider les développeurs à réduire les difficultés d'utilisation des COTS dans une application, comme nous le verrons dans le chapitre suivant.

Cependant, le problème est souvent porté sur l'évaluation des coûts et des risques. Plusieurs auteurs utilisent cette évaluation pour en déterminer le coût et voir si celui-ci n'est pas supérieur à celui de son développement en interne. De plus, il faut pouvoir estimer le budget de réalisation et éviter les dérapages dus à une mauvaise estimation des efforts ou des difficultés d'intégration.

L'étude du STSC (Software Technology Support Center) [STSC03] décrit les différentes causes de risque de l'utilisation des COTS. Les défauts rencontrés sont :

- Nombre insuffisant d'entrées,
- Architecture trop pauvre,
- Conflits,
- Mauvaise évaluation des coûts,
- Problèmes de technologies,
- Dépassement des budgets.

La liste ci-dessus n'est pas exhaustive, mais la plupart des problèmes ou des facteurs de risques sont liés à un manque d'informations permettant de prévenir ces risques.

Si l'on reprend cette liste, on remarque que la détection précoce du nombre insuffisant d'entrées (premier défaut de cette liste) devrait permettre de ne pas sélectionner un COTS qui lors d'une phase ultérieure (phase d'assemblage par exemple) s'avère être un mauvais candidat. C'est le même cas de figure pour le deuxième défaut de la liste : l'architecture trop pauvre. Cela veut dire qu'il nous faut un ensemble d'informations permettant aux développeurs de détecter ces différents facteurs de risque.

C'est un travail important du fait du volume d'informations à traiter : Madachy identifie 600 conditions de risque ou des combinaisons de paramètres d'entrée [Madachy97]. Il propose une interdépendance du coût d'intégration et de l'estimation du risque. Cela est dû au fait que des informations pour le calcul des coûts peuvent influencer l'estimation des risques et vice-versa.

Les modèles de coût sont souvent employés pour la planification et l'évaluation du projet en termes d'effort humain (homme/mois) et de délai de réalisation. La notion de risque peut influencer ces résultats en étudiant les facteurs qui ont permis le calcul du coût. L'interdépendance proposée par Madachy pour les deux types de calcul (estimations des risques et des coûts) devrait permettre d'identifier l'influence d'un facteur (les risques) sur l'autre (le coût de développement). Elle est donc indispensable pour l'évaluation et le choix des COTS.

Tout cela semble efficace, dans la mesure où les informations permettant ces évaluations sont accessibles et disponibles.

II - 5 Les COTS dans une application

Les COTS sont des produits logiciels appartenant à des tiers et ayant la capacité de proposer l'un de ses services (ou plus) comme partie indispensable au fonctionnement d'une autre application. C'est dans le processus de développement de cette dernière qu'il faut prévoir une approche permettant l'intégration et l'utilisation des COTS. C'est le cas dans les processus d'EPIC (Figure 1) et de celui de Morisio (Figure 2). Cela n'empêche pas qu'il faut résoudre certaines difficultés d'intégration comme celles décrites par l'étude du STSC [STSC03] pour pouvoir utiliser ces COTS.

Dans le paragraphe suivant (III), nous décrirons l'utilisation des COTS dans le développement d'une application depuis la sélection des produits jusqu'à son assemblage. Nous parlerons aussi des gains attendus et des difficultés rencontrées afin de les obtenir.

III Utilisation des COTS

III - 1 Principe d'utilisation

Les COTS sont utilisés totalement ou partiellement dans le cadre d'une application plus importante. Pour faire en sorte qu'une application intègre les services d'un COTS, il va falloir de manière simpliste, le sélectionner (voire l'acquérir) et l'assembler dans l'application à base de COTS. Ce sont les deux opérations minimales qu'il va falloir réaliser pour l'intégrer dans le développement.

Le fait que ces services soient déjà réalisés par le COTS, le développeur pense pouvoir les utiliser à moindre effort. Dans l'esprit des développeurs, l'utilisation des COTS doit leur permettre d'obtenir des gains dans plusieurs domaines du développement d'une application. Néanmoins, si ces gains peuvent être obtenus, il existe un certain nombre de difficultés qu'il faudra prendre en compte dans l'utilisation des COTS comme le volume d'informations à traiter, la difficulté de les obtenir, la détection des conflits et dysfonctionnements, l'effort de développement pour l'adaptation des COTS qui dans certains cas est loin d'être négligeable, la notion de risque et de coût qui n'est pas évaluable facilement. Cette liste, non exhaustive, est un échantillon des difficultés que nous vous proposons d'aborder ci-après à travers les différentes phases du processus de développement d'une application à base de COTS.

III - 2 La sélection des COTS

La sélection des produits est un processus long qui démarre dès la définition des besoins de l'application à base de COTS. Il est donc présent dans toutes les phases de la vie de l'application. Son rôle est de sélectionner les produits en adéquation avec les besoins et de prévenir au mieux des risques de dérapage du processus d'intégration en termes de délais et de coûts notamment.

Au tout début, des candidats potentiels sont sélectionnés. Puis au fur et à mesure des évaluations, des tests d'intégration, le nombre de candidats diminuera et il ne restera que les produits que l'équipe de développement aura choisis. Ceux là seront assemblés dans l'application. Ils pourront toujours être remis en cause lors de la phase d'assemblage pour des problèmes de faisabilité ou lors de la mise à jour de l'application à base de COTS pour des raisons stratégiques d'évolution ou de gains de performance.

A chaque fois qu'un produit est susceptible d'être assemblé dans l'application, un processus de développement spécifique à ce type d'application est mené par les équipes de développement. Ce processus est constitué de cinq grandes phases :

- La sélection des COTS,
- La qualification,
- L'adaptation,
- L'assemblage,
- La mise à jour.

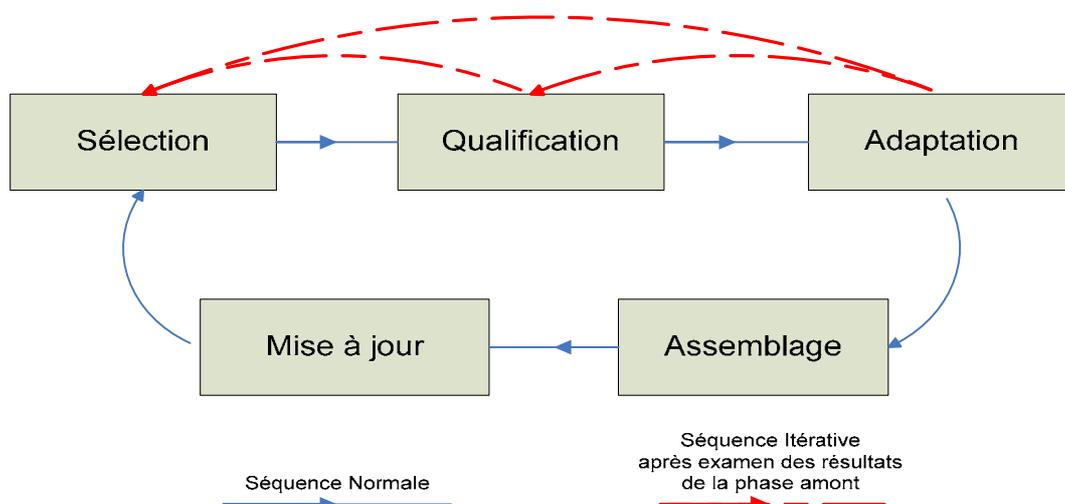


Figure 3 : les cinq grandes phases du processus de développement à base de COTS

La sélection des produits et leur évaluation sont présentes dans toutes les phases sauf celle de l'assemblage. Cette dernière utilise les COTS sélectionnés pour la réalisation de l'application finale. Son objectif n'est pas d'évaluer les COTS mais l'application dans son ensemble. Il est indispensable de détecter les incompatibilités sur les COTS avant de les intégrer. Il est donc nécessaire pour réaliser une sélection correcte d'obtenir un nombre important d'informations utiles dans chaque phase citée ci-dessus. Pour cela Torchiano propose une série d'attributs permettant de caractériser les produits [Torchiano02]. Ils sont issus principalement de la norme qualité ISO9126. Il propose d'autres attributs, notamment pour le calcul du coût de l'assemblage d'un COTS. Ils ont tous pour objectif de permettre :

- d'identifier les services et la qualité logicielle,
- la comparaison des produits entre eux,
- de s'assurer d'un choix en accord avec les critères des sélections préétablis.

De ce fait, avoir toutes ces informations permet de réaliser au mieux les étapes du processus du développement, ce qui est primordial. De même, il faut détecter au plus vite les difficultés pour les phases d'assemblages et d'intégration pour pouvoir mesurer les risques encourus.

III - 3 L'assemblage des COTS

La sélection définitive étant faite (Figure 3), il reste seulement les produits qui seront utilisés dans l'application à base de COTS. Il s'agit dans cette phase de les assembler de manière à réaliser une application cohérente utilisant les services des COTS et ceux développés par l'équipe de développement. Cet assemblage doit être cohérent avec la spécification des besoins réalisée en amont du projet.

Cette phase utilise les informations des phases précédentes (Sélection, Qualification, Adaptation), et notamment l'identification des services où la manière d'intégrer un COTS. Toutes ces informations serviront à la phase d'assemblage. Les principales difficultés ont été normalement évaluées et résolues lorsque l'on se situe dans cette phase du processus de développement. Une remise en cause d'un produit sélectionné dans cette étape avancée du processus serait dramatique car elle remettrait en cause la totalité du processus de développement. Cela entraînerait une perte de temps et d'argent considérable (on doit refaire toutes les étapes du processus de développement). Certains développeurs, peuvent ainsi justifier que l'utilisation des COTS n'est pas désirable : il vaut mieux tout développer en interne avec son équipe de développement afin de mieux maîtriser les impondérables.

Pourtant les processus de développement spécifiques à l'utilisation des COTS ont été mis en place pour faciliter leur utilisation et limiter les risques. Plusieurs phases ont été établies et chacune d'entre elles essaye d'apporter des solutions aux difficultés de mise en œuvre (Chapitre 1 : III - 5). Nous les détaillerons dans la section suivante (Chapitre 1 : IV) en fonction des phases du processus.

III - 4 Gains attendus

Le premier des gains attendus, car le plus évident, c'est le gain financier. Le développement d'un produit est coûteux en moyens et celui-ci est en principe réparti sur l'ensemble des acheteurs de COTS. Il revient de ce fait, au moment de l'acquisition, moins cher que son développement unitaire.

Puis, le gain de rapidité, en ce qui concerne l'accès aux services, est tout aussi évident car le produit existe. L'accès au produit et à ses services est obtenu plus rapidement que s'il fallait les développer du fait qu'ils existent déjà (pas de développement à faire). De ce fait, le délai de présence sur le marché du produit à base de COTS devrait être plus rapide.

Un autre gain attendu, c'est la qualité des produits qui devrait être meilleure car ils sont réalisés et testés par des professionnels du domaine d'activité du COTS. Ils ont en principe une meilleure connaissance du métier auquel fait référence le produit.

De plus ils sont parfois développés avec les dernières technologies qu'il est difficile d'acquérir et de maintenir dans les équipes de développement. Il est de fait plus facile en principe de se positionner sur la technologie la plus adaptée avec l'application qui sera développée.

Pourtant, tous ces gains cachent de réelles difficultés de mise en œuvre que l'on découvre dans toutes les phases du cycle de vie de l'application à base de COTS : de la sélection des produits jusqu'à leur mise à jour.

III - 5 Les difficultés de mise en œuvre

Les gains proposés dans le paragraphe précédent ont tous un défaut. Le gain sur le coût d'acquisition a sa contrepartie. Il faut régulièrement investir pour posséder les dernières versions, corrections ou compatibilités avec les systèmes opérants.

Le gain de temps lors de l'acquisition cache un effort non négligeable pour l'intégration du produit. En utilisant une technique d'intégration comme le « Glue Code », (fréquemment employée) dans laquelle, il faudra réaliser des développements spécifiques pour obtenir l'intégration. Mais, quelles que soient les techniques d'intégration, propres à chaque produit, les équipes de développement vont devoir apprendre à les maîtriser. Bref, l'effort d'intégration dépendra du volume de travail nécessaire pour l'assemblage du produit dans l'application à base de COTS et des besoins en formation pour les développeurs. Cet assemblage n'est pas forcément immédiat et, dans le processus de développement, il est prévu de réaliser des évaluations du coût homme/mois nécessaire afin d'estimer ce gain de temps et d'argent que l'on vient de citer. D'autant plus que le code n'est pas exempt de bugs ou de non qualité. D'où la nécessité de réaliser des évaluations pour les détecter et en définir les risques [Carney03].

Tout cela nécessite un travail en collaboration avec les équipes de support des vendeurs de COTS afin d'évaluer le produit, de l'adapter aux besoins et de l'assembler dans l'application à base de COTS. Les concepteurs de ces produits doivent pouvoir aider les équipes de développement en fournissant les informations nécessaires à ces activités. Cependant, l'une des principales difficultés de l'utilisation des COTS provient de l'absence d'informations permettant la sélection, l'évaluation et l'intégration des produits dans une application. Bertoa a réalisé une étude [Bertoa03] sur le nombre de COTS permettant de réaliser des mesures sur les attributs issus de la norme ISO9126 (norme sur la qualité logicielle [ISO99]). Les résultats démontrent parfaitement bien cette absence d'information qui nuit à l'utilisation des COTS. La très grande majorité des produits ne permettent pas de réaliser leur propre évaluation. De ce fait, la sélection des COTS est d'autant plus difficile que les produits ne peuvent pas être comparés lors de la première phase de présélection. Ce n'est qu'après les évaluations que bien souvent les produits sont éliminés car ne correspondant pas avec les critères définis lors des spécifications des besoins. Cela est dommageable car si les informations sur les produits étaient présentes dès la première phase de sélection, les développeurs auraient :

- évité des évaluations inutiles,
- gagné du temps qu'ils consacraient à une meilleure évaluation des autres produits restants,
- permis la sélection d'autres produits, peut être plus adaptés à leurs besoins, non sélectionnés faute de temps.

Dans tous les cas, une réponse plus approfondie sur les évaluations, un éventail plus large de produits en adéquation avec le besoin se veut être plus bénéfique. De fait, les développeurs se consacreront plus aux choix du meilleur produit en termes de qualité, d'adéquation aux besoins ou de tout autre critère mis en place pour la sélection des produits ou pour leur élimination.

De cela découle un autre problème de l'utilisation des COTS : le risque encouru lors de la sélection d'un mauvais produit dont on se rendrait compte de l'erreur dans une phase avancée du processus de développement de l'application à base de COTS. La remise en cause d'un produit renvoie vers la première phase du processus de développement : la sélection d'un nouveau candidat. Chaque itération est un coût supplémentaire pour l'utilisation du produit comme le présente Boehm dans le « Spiral Model » [Boehm86] complété par la Figure 4 dans un modèle appelé « WinWin Spiral Model »

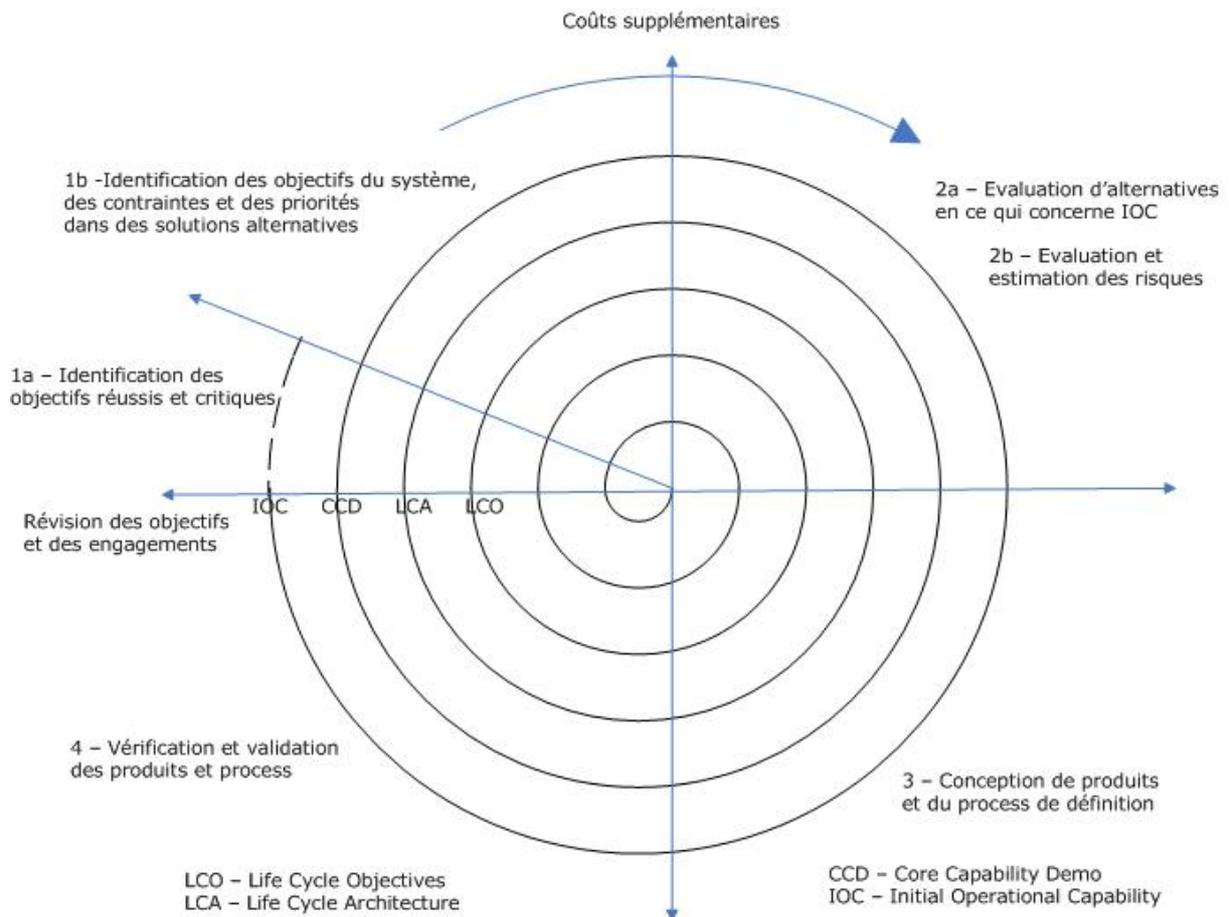


Figure 4 : WinWin Spiral Model [Boehm02]

Plus le nombre d'itérations est élevé et plus les gains attendus s'amenuisent. Car au coût d'acquisition s'est ajouté les coûts d'évaluation et d'intégration. Mais c'est aussi les délais de réalisation qui augmentent à chaque itération. Ce qui réduit les gains attendus en termes de rapidité d'acquisition et de mise en œuvre de l'application dans un marché concurrentiel. Pourtant l'utilisation des COTS semblent être une bonne solution. Pour réduire ces difficultés, l'utilisation de processus de développements spécifiques semble être le moyen le mieux adapté.

III - 6 La nécessité d'un processus de développement

Les développeurs d'applications veulent trouver dans l'utilisation des COTS un moyen d'obtenir des gains financiers, des améliorations sur les délais de réalisation et une amélioration de la qualité logicielle. Pourtant, nous avons remarqué que des choix erronés, par manque d'informations sur les produits, pouvaient conduire à un effet inverse rendant l'utilisation des COTS moins intéressante que le développement de l'ensemble des services d'une application. Pourtant l'idée même de ne pas tout redévelopper est plus que séduisante. **Dans d'autres domaines comme l'électronique, la conception de voitures ou d'autres produits, la réutilisation d'éléments existants est une réalité que personne ne remet en cause.**

Comme nous venons de le voir, les processus de sélection et des tests d'assemblage permettent d'appréhender les risques. C'est le cas notamment des normes qualité comme l'ISO9126, ou les processus itératifs comme le WinWin Spiral Model présenté précédemment. Les phases d'assemblages et de prototypage permettent de recueillir les informations qui permettent d'évaluer les risques et les coûts.

C'est donc à l'intérieur d'un processus de développement spécifique aux COTS (décrit dans la section qui suit) qu'il va falloir intégrer les moyens de récupérer les informations permettant de limiter les coûts et les risques d'utilisation des COTS.

IV Le processus de développement d'une application à base de COTS

IV - 1 Contexte

Dans le cadre de la réalisation d'une application à base de COTS, son processus de développement peut être décomposé en différentes activités ou phases. On y trouve communément [Morisio02] :

- Une phase de spécification des besoins,
- Une phase d'acquisition et de sélection,

- Une phase de qualification des produits sélectionnés et devant être intégrés à l'application,
- Une phase d'adaptation permettant d'étudier l'intégration des COTS,
- Une phase d'assemblage et donc de réalisation de l'application avec les COTS choisis,
- Une phase de mise à jour permettant de faire évoluer l'application à base de COTS.

Il est possible à partir des descriptions de ces phases présentées par Morisio et communément acceptées par tous, de décrire la séquence du déroulement d'un processus de développement d'application à base de COTS. Nous proposons de les décomposer en trois niveaux (Stratégique, Opérationnel et Décisionnel) et d'identifier parmi ces niveaux celui dont l'utilisation d'un prototype est nécessaire.

- Le niveau stratégique regroupe les deux premières phases du processus (veille technologique et définition des spécifications) et la dernière (la mise à jour de l'application). Ce niveau correspond à une activité de recueil d'informations permettant le lancement du processus de développement de l'application à base de COTS.
- Le niveau opérationnel correspond à la phase d'acquisition et de sélection, puis à la phase d'assemblage. Ces deux phases correspondent à une activité de réalisation qui s'appuie essentiellement sur les informations recueillies dans les activités relatives aux autres niveaux.
- Le niveau décisionnel recueille les autres phases du processus (Evaluation, Qualification, Adaptation, Tri et sélection) et doit permettre de décider du choix des COTS à utiliser dans l'application. C'est notamment dans les activités de ce niveau que se situe principalement l'utilisation des prototypes.

IV - 2 La phase de spécification des besoins

C'est une phase indispensable à tout processus de développement logiciel. Elle est présente dès la première étape du processus d'un développement classique (sans COTS) que nous situons dans le niveau stratégique. Par contre, dans le développement à base de COTS c'est une phase qui peut être menée en parallèle avec de la veille technologique sur les COTS. Selon Botella [Botella02] cette phase de définition des besoins peut être négociée pendant la sélection des produits faisant partie de l'application. C'est donc la

veille technologique qui guiderait une partie de la définition des besoins. La Figure 5 présente un processus avec deux branches parallèles :

- la connaissance du domaine ou métier et l'évaluation de la qualité logicielle,
- la sélection des produits pour en extraire la spécification.

En fonction des spécifications de qualité logicielle et la description des produits métiers la spécification des besoins est réalisée.

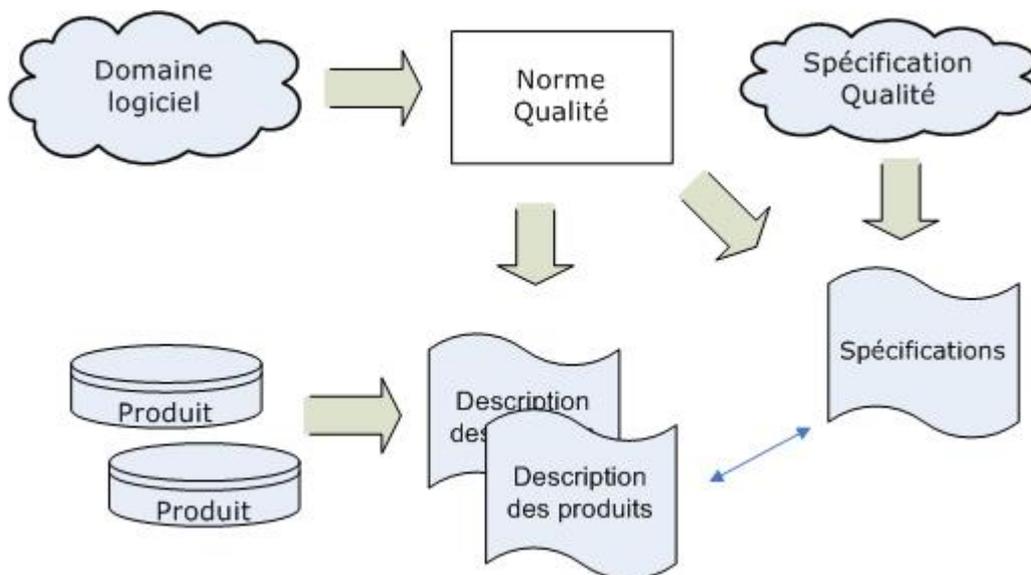


Figure 5 : spécification des besoins

Cette approche (Figure 5) montre l'importance de la veille technologique qui doit permettre :

- d'évaluer la qualité des produits,
- d'extraire les descriptions fonctionnelles des COTS.

L'objectif de cette approche est de permettre d'une part d'avoir suffisamment d'informations pour la réalisation des spécifications de l'application à base de COTS et d'autre part de permettre une sélection plus proche des besoins puisque les spécifications ont été réalisées à partir de la description des produits.

Ce sont les spécifications sur la qualité recherchée et les besoins fonctionnels qui définissent les candidats qui seront évalués dans la phase suivante d'acquisition et de sélection.

IV - 3 La phase d'acquisition et de sélection

La phase d'acquisition et de sélection est la deuxième étape du processus de développement comme nous le proposons dans la Figure 6.

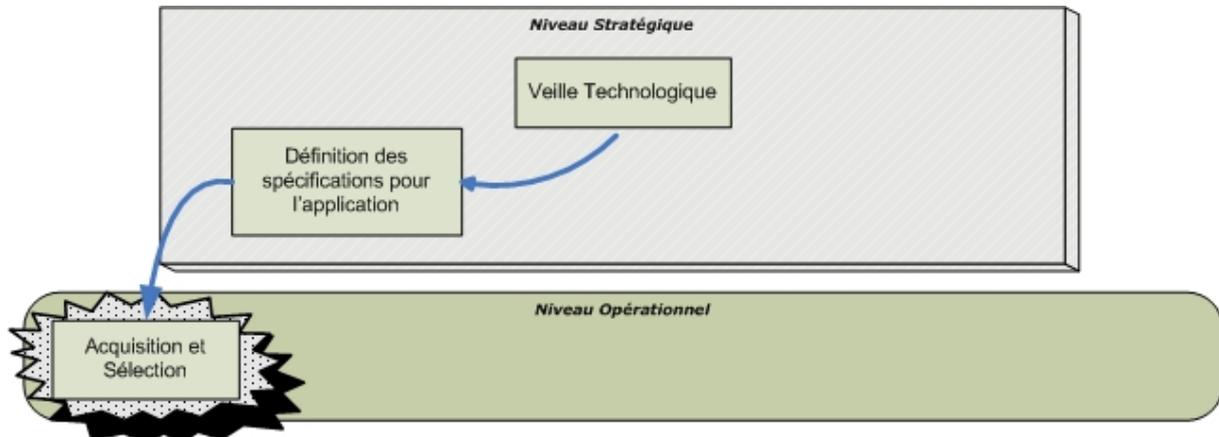


Figure 6 : Premières étapes du processus de développement basé COTS.

Cette phase doit permettre la sélection des COTS susceptibles de convenir à l'application à base de COTS qui doit être développée. Elle doit réunir les candidats qui seront audités dans la phase suivante de qualification. De ce fait, il faut identifier et évaluer différents produits. C'est donc une phase opérationnelle rattachée au niveau du même nom.

Pour cela, il existe une grande variété d'approches d'évaluation de COTS dont voici une brève liste des méthodes les plus rencontrées et fournies par Port [Port04]:

- PORE (Procurement Oriented Requirements Engineering method) conduit en parallèle la description des besoins de l'application à développer et de celle des fonctionnalités des COTS. Le choix se faisant sur la meilleure adéquation entre les besoins et les services proposés par les COTS [NCube99],
- OTSO (Off The Shelf Option) propose une sélection conduisant au choix du COTS le plus approprié dans un contexte donné de conditions. Pour cela il utilise plusieurs phases avec un ensemble complet de mesures, d'évaluations détaillées, d'évaluation de coûts et de valeurs, et puis la décision d'achat pour un logiciel spécifique de COTS [Kontio95],
- CDSEM (Checklist Driven Software Evaluation Methodology) propose une évaluation à l'aide d'une liste de valeurs métriques permettant un classement aisé des candidats [JeanRenaud94],

- CISD (COTS-based Integrated System Development) est une méthode qui se déroule en deux phases : l'identification et la sélection des candidats puis une phase d'évaluation [Tran97],
- CAP (COTS Acquisition Process method) propose un processus en trois parties : initialisation, exécution et réutilisation. La première partie traite de la planification du procédé d'acquisition et de son évaluation des coûts. La deuxième partie conduit à l'évaluation des COTS et à la prise de décision pour l'achat des produits. La troisième partie est responsable de la capitalisation des informations afin de diminuer le coût de futurs procédés d'acquisition de COTS [Ochs01],
- CRE (COTS-Based Requirements Engineering method) est une méthode qui s'attarde plus particulièrement à la définition et à l'analyse des conditions non fonctionnelles pendant les phases de l'évaluation et du choix des COTS [Alves01],
- CEP (COTS-Based Requirements Engineering method) propose une sélection des COTS par la comparaison de l'estimation des coûts d'intégration des produits [Phillips02]
- CBA (COTS-Based Application Process Decision Framework) [Boehm02],
- STACE (Socio-Technical Approach to COTS Evaluation) se décompose en 4 activités qui sont la définition des besoins, la définition des critères sociaux et techniques, l'étude des alternatives et les évaluations [Kunda99],
- PECA est une méthode qui se décompose en 4 phases : planification de l'évaluation, l'établissement des critères, le rassemblement des données et l'analyse des données [Comella02].

Dans toutes ces méthodes décrites dans la littérature on y retrouve des activités communes comme :

- L'identification des produits,
- L'évaluation des produits,
- La présélection des candidats.

Dans le cadre de l'identification des produits, on parle souvent de l'importance de la veille technologique et de la connaissance des logiciels appartenant à un domaine ou à un métier. Elles permettront entre autres d'identifier les services à mettre en œuvre dans l'application à base de COTS et d'établir des évaluations qualitatives des produits. Cette

anticipation sur leurs évaluations doit permettre de présélectionner plus rapidement les bons éléments (ceux qui seront utilisables dans l'application à base de COTS). Cependant, les produits existants et utiles pour l'application à base de COTS à développer sont souvent méconnus et mal évalués. Pourtant, c'est leur connaissance qui rendra les phases d'identification et d'évaluation plus efficaces et réduira les temps de traitement.

Ici, leur évaluation sert à faire un premier tri permettant d'éliminer tous les COTS qui ne correspondent pas à nos besoins ou aux critères de sélection établis. C'est pour cela que dans cette phase il est fréquent de réaliser des tests d'adéquation et de qualité. Ils sont tous deux des éléments de la présélection des candidats.

Plusieurs critères sont utilisés [Alves01, Comella02, Dean00, Kunda99] :

- Les garanties offertes par le vendeur de COTS
- Les contraintes techniques (architecture, technologies, programmation)
- Les services rendus et la qualité de service (exemple des temps de réponses, de la fiabilité et autres)
- La nature du COTS : White, Grey or Black Box

Kunda [Kunda99] ajoute même des critères sociaux économiques (la réputation des vendeurs, le produit est-il leader du marché ?, le type et la durée du contrat de licences, etc...). Dans la méthode PECA [Comella02], c'est l'équipe de développement qui définit les critères à mettre en œuvre pour la sélection en établissant un poids à chaque attribut de l'évaluation. Ainsi, il sera possible de classer les différents produits en compétition.

Dans tous les cas, les développeurs cherchent tous à obtenir le COTS qui correspondront au mieux à leurs besoins, aux contraintes techniques, qualitatives, économiques ou subjectives. Cependant, il reste maintenant à vérifier les qualités des candidats à l'intégration dans l'application à base de COTS, notamment en termes de difficultés de développement, de coûts de réalisation et de qualité des produits.

IV - 4 La phase de qualification

La phase de qualification est la suite logique aux étapes précédentes, ce qui nous permet de compléter notre schéma de la Figure 7.

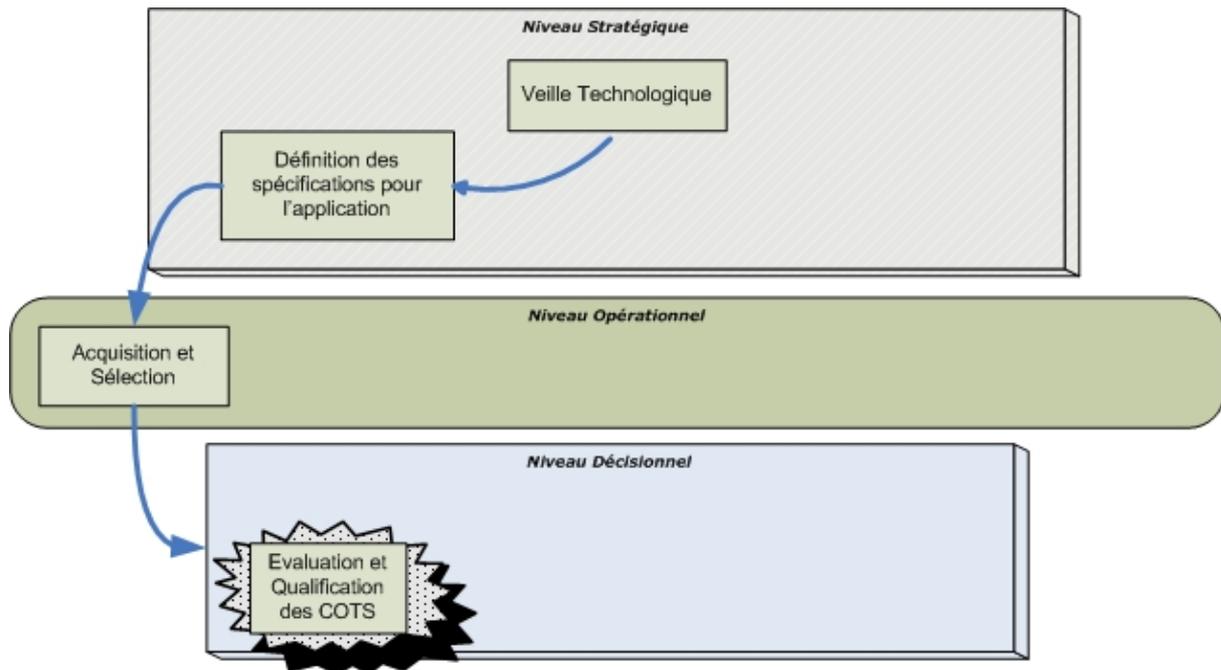


Figure 7 : Les étapes jusqu'à la qualification et l'évaluation des COTS

Cette phase doit aller plus loin dans l'évaluation des COTS. Elle concerne principalement les activités [Kunda99, Dean00] :

- d'identification des services appartenant aux composants face à l'expression des besoins,
- de test de la qualité des produits en fonction des objectifs et des contraintes de l'application à base de COTS,
- d'estimation des coûts d'intégration et des risques.

L'identification des services doit permettre d'évaluer l'adéquation du COTS avec les besoins. Elle est parfois mesurée par un pourcentage de couverture des fonctionnalités désirées et pour chacune d'entre elles un pourcentage de correspondance aux spécifications. Ces mesures sont faites de manière qualitative et non quantitative.

C'est dans l'activité liée aux tests qualité que des mesures plus quantitatives vont étayer les résultats précédents. Pour cela, c'est la référence à la norme ISO9126 composée de 6 caractéristiques (fonctionnalité, fiabilité, utilisation, efficacité, maintenable et portabilité) qui sera le plus souvent utilisé. De cette activité en découle une bonne connaissance de l'utilisation et de l'intégration des produits.

Ceci est nécessaire pour évaluer notamment l'effort de développement qu'il faudra réaliser dans la phase d'assemblage.

Pour cela deux modèles sont particulièrement présents dans la littérature : COCOTS (*COTS Software Life Cycle Cost Model*) et COCOMO II (*Constructive COST Models*) [ESI01]. Ils compilent un certains nombres d'informations permettant de réaliser l'estimation des coûts d'intégration et l'évaluation des risques.

COCOMO qui est le premier modèle [Boehm81] conçu parmi les deux cités précédemment réalise une estimation en fonction :

- des temps et des coûts humains de développement,
- de la capacité des développeurs à réaliser l'intégration. En tout, quinze paramètres supplémentaires sont proposés pour tenir compte de l'environnement de développement.

La version II de COCOMO intègre des notions supplémentaires

- dans les premières phases du développement. Elle intègre la taille du produit, la nature de la plate-forme cible et du personnel impliqué dans le processus de développement,
- dans les phases d'évaluations des projets, elle intègre les coûts des différentes itérations nécessaires aux choix des COTS.

La limite du modèle COCOMO est la nécessité d'avoir le code source des produits pour l'évaluation. C'est là que COCOTS (*CONstructive COTS integration model*) est complémentaire à COCOMO puisqu'il traite les cas où le code source n'est pas disponible. Pour cela, il utilise les quatre principaux domaines de coût :

- l'évaluation des composants,
- mise en œuvre des composants,
- l'insertion du « Glue Code »,
- la mise à niveau des COTS.

Ces modèles manipulent un nombre important d'attributs qui vont permettre d'avoir une bonne connaissance des produits en termes : d'utilisation, de la manière de les assembler, de coûts d'intégration et des risques encourus.

Ils permettent aussi de préparer la phase suivante d'adaptation pour laquelle il est nécessaire d'avoir notamment des informations techniques que l'on retrouve en partie dans les modèles COCOTS ou la norme ISO9126, et qui permettront la réalisation des prototypes indispensables à la phase suivante (l'adaptation) comme nous le verrons par la suite.

IV - 5 La phase d'adaptation des COTS dans l'application basée COTS

La phase d'adaptation permet de poursuivre le travail effectué dans la phase précédente de qualification. Cependant, du fait des résultats des évaluations et des estimations de risque, il est possible de trier les COTS qui seront étudiés dans cette phase. C'est un filtre qui, en fonction des résultats de la phase précédente, va permettre d'éliminer ou de donner une priorité sur les COTS qui seront étudiés dans la phase « d'adaptation des COTS ». C'est donc une partie de la phase de sélection qui se trouve intercalée entre la phase d'évaluation et d'adaptation comme présenté dans la Figure 8 et que nous avons nommé « Tri et Sélection des COTS (1) ».

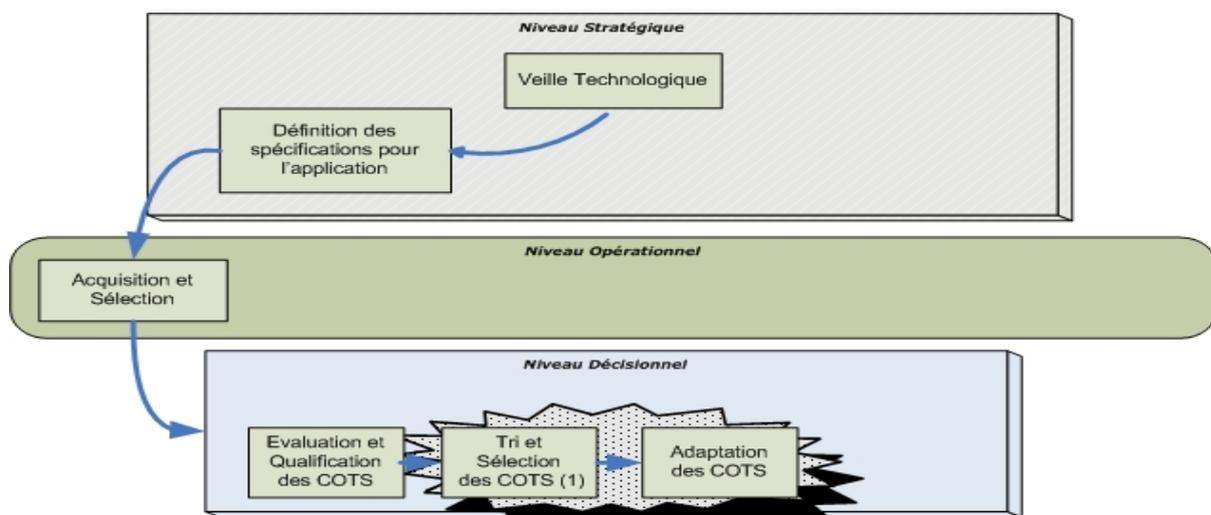


Figure 8 La phase d'adaptation de COTS dans le processus de développement à base de COTS.

Pour l'adaptation des COTS dans l'application finale nous pouvons utiliser plusieurs méthodes d'intégration. Vigder présente dans [Vidger97] l'intégration de composants par l'utilisation de « Wrappers », « Glue Code » et « Component tailoring ». C'est selon l'auteur les trois méthodes d'intégration de composants.

Le « wrappers » correspond à du code enveloppe permettant l'intégration en isolant les fonctionnalités non désirées des COTS. Ils fournissent de ce fait, les seules méthodes permises pour accéder aux COTS.

Plusieurs articles parlent du « Glue Code » comme l'élément le plus couramment utilisé dans l'intégration de COTS. Il permet de relier les différents composants entre eux et se compose :

- de fonctionnalités utilisant les interfaces des composants,
- de bouts de code réalisant les adaptations permettant la compatibilité des données entre composants,
- du contrôle des exceptions.

Les « Components tailoring » correspondent à tous les composants qui sans modification du code du vendeur permettent l'addition de fonctionnalités utilisables par une application. C'est le cas des « Plug-in » qui ajoutent des fonctionnalités comme par exemple l'encodage au format MP3 des fichiers audio pour une application comme Nero.

Ces différentes techniques sont spécifiques à chaque produit à assembler. L'objectif de cette phase est :

- de définir la technique à mettre en œuvre (Glue Code, tailoring, wrappers),
- de décrire les moyens d'intégration des services dont aura besoin l'application,
- de permettre la réalisation de tests qualité pour s'assurer de la cohérence avec les spécifications qualité définies pendant la phase de définition des besoins,
- de vérifier les problèmes éventuels qui pourraient apparaître lors de l'assemblage final,
- de valider la sélection de COTS pour la phase de développement et d'assemblage.

Pour cela il est nécessaire d'obtenir toutes les informations techniques permettant la réalisation d'un prototype. D'après [Leung02], ce dernier est présent dans toutes les phases où une activité d'évaluation, de contrôle ou de faisabilité est nécessaire. En particulier dans la phase d'adaptation qui doit permettre d'établir la faisabilité de l'intégration.

IV - 6 La phase d'assemblage

La phase d'assemblage ne sera réalisée que si la faisabilité de l'intégration est établie. C'est à dire dans la phase amont à l'assemblage que nous avons nommé « tri et sélection des COTS (2) ». Cette phase est là pour établir un « choix définitif » sur les COTS à assembler, mais aussi pour introduire une boucle de retour dans le niveau décisionnel pour établir de nouvelles phases de qualification et d'évaluation dans le cas où d'autres évaluations et qualifications de COTS s'avèrent nécessaires. Elles permettront d'affiner les résultats déjà obtenus. Cette itération correspond à celle présentée dans le « WinWin Model » de Boehm (Figure 4).

De ce fait, nous complétons le schéma du processus de développement avec la Figure 9.

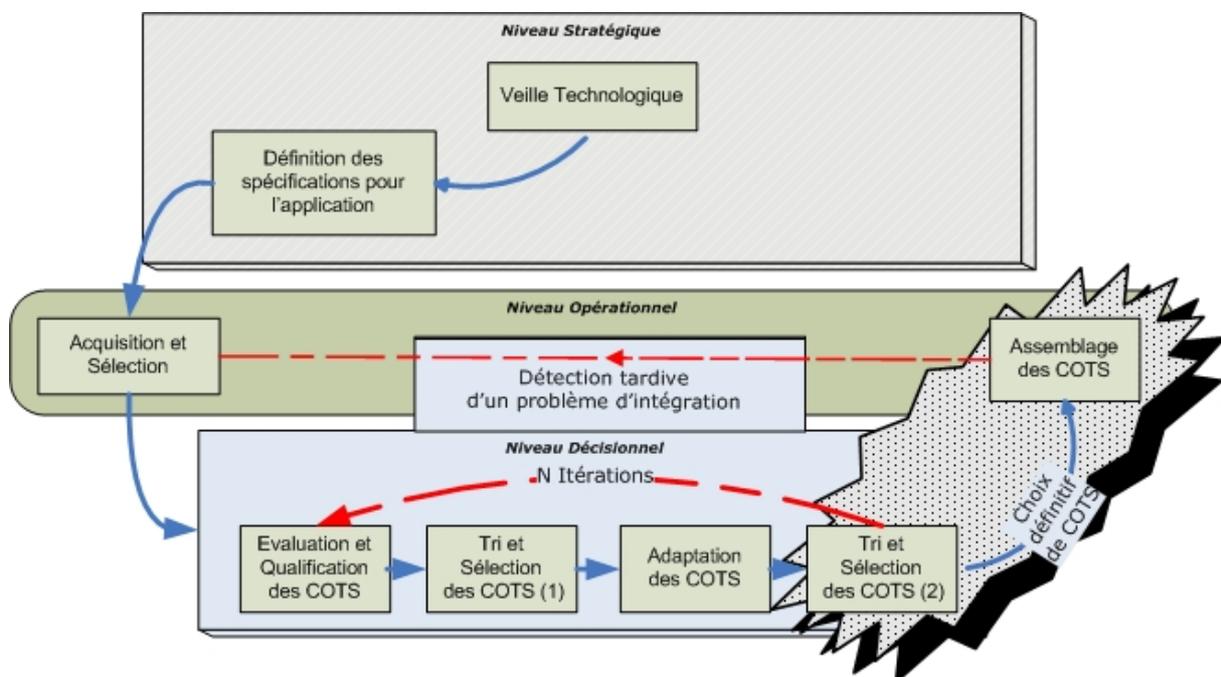


Figure 9 : La phase d'assemblage dans le processus de développement à base de COTS

A partir des techniques d'assemblage définies dans la phase précédente, le développeur réalise dans cette phase l'intégration de tous les COTS dans l'application finale. Le but

principal de la phase d'assemblage c'est aussi de réaliser, comme dans tout processus classique de développement logiciel, les tests qualité permettant de s'assurer du bon fonctionnement de l'ensemble.

Le fait d'avoir des produits dont on ne connaît pas complètement le comportement, ni le code source ajoute une difficulté supplémentaire à ces évaluations. Pour Morisio, dans cette phase, les tests d'intégration doivent prendre en compte le système entier plutôt que le bon fonctionnement des COTS indépendamment de l'application dans laquelle ils sont assemblés [Morisio00]. C'est l'exécution du système entier avec l'addition des COTS qui doit être testé afin de vérifier qu'ils ne perturbent pas le fonctionnement de l'ensemble.

Un dysfonctionnement à cause d'un COTS, à ce niveau du développement, aurait de graves conséquences sur les temps et le coût de développement car ils peuvent remettre en cause tout le travail d'évaluation et de sélection des COTS. Cela peut se traduire par une remise en cause des COTS sélectionnés. De ce fait, le processus de développement devra reprendre à sa deuxième phase : l'acquisition et la sélection de nouveaux COTS.

Dans la plupart des cas, les COTS sélectionnés ne sont pas remis en cause, et l'étape suivante se poursuit par le déploiement de l'application sur le système cible. Son cycle de vie ne se termine pas pour autant puisqu'il est toujours possible de modifier une application pour l'adapter à de nouveaux besoins, pour des améliorations ou des corrections par exemple : ceci fait partie de la phase de mise à jour.

IV - 7 La phase de mise à jour

Une fois l'application terminée avec un numéro de version établie et en cours d'utilisation, il est prévu dans un processus classique d'amélioration logicielle de faire évoluer l'application. Dans ce cadre là, les développeurs peuvent vouloir remplacer ou ajouter de nouveaux COTS. Cette phase de mise à jour prévoit de dérouler à nouveau le processus de développement à base de COTS que l'on vient de décrire.

Cela permet de présenter la totalité du processus de développement à base de COTS dans la Figure 10.

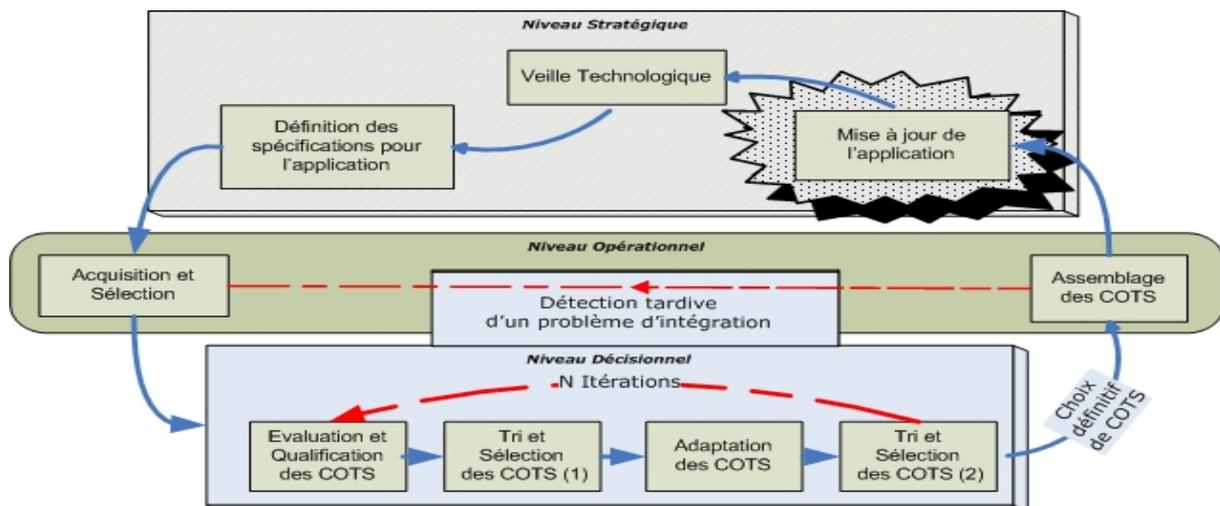


Figure 10 : processus de développement à base de COTS.

Dans ce processus, on ajoute obligatoirement dans la phase de sélection, la comparaison de l'ancienne et de la nouvelle version de l'application à base de COTS. Les évaluations de coût ou de qualité logicielle doivent démontrer l'utilité ou l'intérêt de remplacer un composant dans l'application. A ces raisons objectives ou tangibles peuvent s'ajouter, comme dans les phases précédentes, des raisons subjectives à la validation des modifications.

C'est donc un nouveau cycle complet du processus de développement qui peut reprendre. Ce cycle itératif est infini. Il ne peut s'arrêter que par l'arrêt de l'évolution de l'application à base de COTS.

IV - 8 Bilan

Le schéma de la Figure 10 est notre vision sur l'enchaînement des phases permettant de dérouler séquentiellement un processus de développement à base de COTS que nous avons décomposé en trois niveaux (Stratégique, Opérationnel et Décisionnel). Nous avons identifié dans cette représentation le niveau dans lequel l'utilisation d'un prototype est nécessaire (niveau décisionnel). C'est donc dans ce niveau que nous voulons apporter notre réflexion pour faciliter la réalisation des prototypes indispensables à l'obtention des informations qui font défaut.

C'est avec la réalisation des prototypes et donc d'une partie de l'intégration qu'il sera possible de déterminer les valeurs nécessaires au déroulement de ces phases et notamment à l'évaluation des risques et des coûts. C'est en fait, pour l'ensemble des équipes (de chaque phase), un volume important d'informations qu'il va falloir traiter et évaluer pour

détecter les problèmes et trouver des solutions, même si ces informations sont parfois difficiles à obtenir de la part des fournisseurs de produits.

C'est donc en agissant sur le recueil des informations et sur leurs partages entre les différentes activités au sein du processus que l'on contribuera à la détection des problèmes et leurs résolutions. Pour cela nous allons établir la cartographie des informations nécessaires et disponibles pendant les phases du processus de développement à base de COTS. Elles permettront de les comparer avec les informations qui sont nécessaires aux développeurs pour établir les évaluations de risques et de coûts qui sont définis comme des critères importants de la bonne sélection et utilisation des COTS.

V L'utilisation d'un prototype pour le recueil d'informations

V - 1 Le rôle du prototype

L'obtention des informations est primordial pour le bon déroulement du processus de développement de l'application basée COTS. Pour réduire les difficultés sur le sujet Leung [Leung02] propose de réaliser des prototypes qu'il juge indispensable à plusieurs phases du processus de développement pour le recueil et la validation des informations.

Le prototype doit permettre d'étudier l'intégration d'un COTS ou de plusieurs COTS cohabitant ensemble. De cette étude les développeurs vont obtenir et valider des informations indispensables pour chaque phase du processus de développement de l'application à base de COTS. L'utilisation de ces prototypes dans des expérimentations est une démarche qui garantit selon Leung l'accès aux informations manquantes et difficilement accessibles. Indépendamment de savoir si cela représente le seul moyen d'obtenir ces informations, le fait est, que l'expérimentation est souvent le meilleur moyen pour s'assurer de la validité des informations recueillies. Il nous semble donc intéressant d'explorer cet axe comme un moyen d'aide aux développeurs d'application à base de COTS.

V - 2 A travers les phases du processus de développement d'application à base de COTS

V - 2.1 Contexte

L'utilisation des prototypes peut se justifier dans toutes les phases où il est nécessaire d'avoir des informations techniques. Il devient aussi nécessaire à partir du moment où les informations sur le fonctionnement, la qualité du produit nécessite la réalisation d'une expérimentation.

A partir de ce constat, on s'aperçoit que la plupart des phases du processus de développement d'application à base de COTS peuvent avoir besoin de réaliser un prototype. C'est ainsi que la phase de sélection fait appel à sa réalisation à partir du moment où la comparaison des COTS nécessite des informations sur la manière d'intégrer les différents produits. Cela peut correspondre aux environnements d'exécution ou d'intégration des COTS mais aussi aux moyens et aux coûts d'utilisation et de développement pour intégrer ces COTS. En fait la phase de sélection, comme nous l'avons décrite dans le paragraphe IV de ce chapitre, fait appel aux phases d'évaluations, de qualification et à celle d'estimations des coûts et des risques pour obtenir des informations permettant d'affiner le choix des COTS.

Cette phase spécifique de sélection en fonction des résultats des évaluations et des estimations de risques et de coûts a été décrite dans la Figure 10 : processus de développement à base de COTS. Elle fait partie du niveau décisionnel représentant les méthodes d'acquisition et de sélection établie dans le paragraphe du Chapitre 2 : IV - 3. Ces méthodes ont toutes une manière itérative de fonctionner que nous avons reproduites dans notre schéma de la Figure 10. C'est un niveau où l'ensemble des phases peuvent avoir besoin d'utiliser un prototype pour l'expérimentation conduisant au recueil et à la validation des informations nécessaires aux processus de développement de l'application à base de COTS.

V - 2.2 A partir des besoins

L'utilisation d'une méthode pour la définition des besoins est une pratique courante. Dans le cadre de notre étude nous avons besoin de manipuler des informations et de les échanger entre différents acteurs du processus de développement. Il est donc nécessaire de standardiser le format des informations ainsi que le mode d'échange. Il en va de même lors du choix des moyens de modélisation. L'utilisation d'un langage fortement utilisée dans l'industrie comme UML (*Unified Modeling Language*) est donc un plus car il est devenu un standard de fait. Pour notre exemple, son utilisation semble naturelle car elle est

supportée par l'OMG (Object Management Group) qui est un organisme de normalisation international qui standardise aujourd'hui UML et qui regroupe les plus grands éditeurs de logiciels.

De la méthode associée à l'utilisation de ce langage de modélisation, nous utilisons sa première étape (les cas d'utilisation) permettant de décrire les services de l'application. C'est cette base d'information qui permettra aux acteurs en charge de la sélection des COTS d'identifier les produits qui pourront suivre les différentes étapes du processus de développement de l'application à base de COTS.

Les produits sélectionnés doivent correspondre aux fonctionnalités désirées pour l'application mais aussi aux contraintes comme la qualité logicielle et l'environnement de développement ou d'exécution. L'ensemble de ces informations doit permettre notamment la sélection des COTS. Elles définissent aussi le cadre des informations qu'il va falloir acquérir auprès des différents fournisseurs de COTS.

D'autre part, au moment de la définition des besoins il est nécessaire d'établir des critères de sélection qui permettront de départager les candidats à l'intégration. Ces critères sont souvent établis sur la base des différents besoins et contraintes définies au début du projet. D'où l'importance des informations définies à partir de cette phase des besoins.

D'autres informations seront utiles, mais elles seront établies en fonctions des différentes étapes que nous décrirons par la suite. Pour l'instant, nous établissons pour la définition des besoins, un ensemble d'informations qui nous semble nécessaire autour :

- des données sur le produit avec les services, les contraintes techniques d'exécution ou de développement et les conditions d'acquisition des produits (licence, documentation, hot line, ...)
- des données concernant la qualité avec les normes qualité comme l'ISO9126,
- des données concernant les spécifications avec les critères de sélection et les critères socio-économiques.

Cela reprend les besoins exprimés par différents auteurs qui nous ont permis de décrire les différentes phases d'un développement à base de COTS.

Parmi ces informations la première (les données sur le produit) sont particulièrement utile à la description du développement d'un prototype. C'est à partir des services à développer, des environnements d'exécution cible et des techniques de développement que le prototype sera réalisé. Ces informations sont donc particulièrement utiles pour le développement d'un prototype.

Les données sur la qualité logicielle ne servent pas directement à la réalisation d'un prototype. Elles sont nécessaires pour le processus de développement de l'application à base de COTS et à ce titre il est indispensable que le prototype permette le recueil des informations s'y rapportant.

Enfin, les dernières données ne sont pas exploitables par un prototype mais servent exclusivement le processus de développement et notamment pour la phase suivante de sélection des COTS.

V - 2.3 Pour sélectionner des COTS

La sélection des COTS est la deuxième étape après la définition des besoins. C'est dans un premier temps l'acquisition des COTS à évaluer, puis dans un second temps leur évaluation, les comparaisons et l'estimation des coûts et des risques qui termineront cette phase de sélection par un choix définitif des COTS et qui permettront le développement de l'application.

C'est donc une phase qui ne se termine réellement que lorsque l'assemblage des COTS se réalise. En référence à la Figure 11, à chaque itération le nombre de candidats se réduit jusqu'à la description d'un scénario comportant les COTS à assembler. C'est pour cela que l'on retrouve cette sélection dans deux phases différentes de la Figure 11 que l'on a nommé :

- « Acquisition et sélection de COTS »,
- « Tri et sélection de COTS ».

La première phase sert au démarrage de l'itération ou lorsque les étapes du processus ont rejeté les premières sélections de COTS et qu'il sera nécessaire de retrouver de nouveaux candidats à l'intégration. C'est le cas notamment de la phase « assemblage de COTS » qui détecte l'impossibilité d'intégrer un produit ayant pourtant réalisé avec succès les autres phases du processus.

Pour cela, au tout début de ce processus, c'est le cas où l'itération ne fait que commencer, ce sont les informations issues de la phase de définition des besoins qui sont utilisées et plus particulièrement dans :

- la définition des services et les contraintes techniques et
- les critères de sélection.

A ces informations, nous allons ajouter, à chaque itération, des informations issues des autres phases (« l'évaluation et qualification » et « l'adaptation »). Elles doivent fournir

des informations qui vont permettre le tri et la sélection des candidats aptes à la poursuite du processus de développement dans la phase que nous avons nommée : « Tri et sélection de COTS ». C'est pour cela que cette phase est dédoublée : chacune utilisant les informations en provenance respectivement des phases « d'évaluation et de qualification » et « d'adaptation ».

L'utilisation d'un prototype dans cette phase dépend des besoins des autres phases qui la constituent : les évaluations, le tri et les estimations de risques et de coûts. C'est donc dans ces phases que l'utilisation du prototype sera la plus importante et nécessaire.

V - 2.4 *Pour trier des COTS*

V - 2.4.1 Comparer des COTS

Pour le choix du produit à intégrer, il est nécessaire de :

- Récupérer des informations citées ci-dessus,
- Comparer les informations de chaque produit avec les besoins et les critères de sélection exprimés dans la première phase du développement.

De fait, on note que les informations définies au tout début du processus sont utilisées en aval de celui-ci. Il est donc important que dans chaque activité de ce processus on puisse disposer des informations nécessaires ou recueillies dans les activités en amont.

Cette comparaison se fait en deux temps :

- Entre un candidat et les besoins exprimés pour l'application,
- Entre candidats potentiels.

Cela doit permettre d'une part de trier les candidats entre eux et d'autre part de vérifier l'adéquation aux besoins.

La comparaison des produits entre eux nécessite les informations déjà citées précédemment : les services, les contraintes techniques, les conditions d'acquisition des produits et la norme qualité. Ce sont ces informations qui sont prises en compte le plus souvent et qui sont citées dans les articles traitant du sujet.

Le deuxième volet, c'est la comparaison avec les besoins pour vérifier l'aptitude des COTS à l'intégration. Ce sont donc les informations appartenant à l'application à réaliser qui sont prises en compte :

- Les services,

- La norme qualité,
- Les contraintes techniques,
- Les critères de sélection dans lesquels on retrouve des critères socio-économiques.

L'utilisation d'un prototype va permettre de manipuler ces informations notamment dans ce deuxième volet. Il validera par l'exemple les informations déjà recueillies et permettra de réaliser les tests qualité.

Cette comparaison ne suffit pas elle seule à départager les produits ni à déterminer s'ils peuvent être intégrés. Pour cela on a besoin de l'estimation des coûts et des risques d'utilisation des COTS. Lors de la comparaison, seuls les produits ne répondant pas aux besoins ou trop éloignés des critères de sélection ne poursuivront pas le processus de développement.

V - 2.4.2 Estimer les risques et les coûts de l'utilisation des COTS

C'est donc, avec le calcul de l'estimation des risques et des coûts que la décision de conserver un produit sera quasi définitive. Seule, son impossibilité à être adaptée et intégrée, pourra le remettre en cause dans l'activité que nous décrivons ci-après.

Pour ces estimations, on utilise des outils comme COCOMO et COCOTS. Ils ont besoin des informations déjà citées précédemment (les services, les conditions d'acquisition et les contraintes technique). Mais il faut recueillir aussi des informations liées aux normes de qualité logicielle et une liste d'attributs propres à chaque outil. On trouve par exemple le nombre de lignes de codes d'un produit ou sa taille. Des éléments estimatifs sur le nombre de services à coder et leurs difficultés de codage pour les équipes de développement. Ce sont des informations qui sont recueillies, pour une grande majorité, avec la réalisation de prototypes mettant en œuvre les services à intégrer. Cet ensemble d'informations ainsi obtenues par la mise en situation, propre aux outils d'estimations des risques et des coûts, permettra de définir définitivement les produits qui feront partie intégrante de l'application finale.

Il ne restera plus qu'à étudier la faisabilité technique. Même si pour l'estimation des coûts, la question a commencé à être abordée, il va falloir maintenant définir la procédure complète de l'adaptation du COTS.

V - 2.5 Pour évaluer et Qualifier des COTS

Dans plusieurs phases du processus de développement, les développeurs auront à réaliser des évaluations. En fonction de l'avancement dans ce processus, le besoin en information ou les critères d'évaluation peuvent être modifiés. Néanmoins, l'objectif reste le même, les développeurs doivent pouvoir déterminer la capacité des produits à répondre aux exigences définies dans la phase de définition des besoins. Ces évaluations vont départager les COTS dans le cadre de la phase de « tri et de sélection ».

Pour cela, plusieurs types d'informations sont manipulés. On trouve principalement les informations sur :

- le produit, avec les services et les contraintes techniques,
- la qualité logicielle et notamment la norme qualité du fournisseur ou client,
- les spécifications avec les critères de sélection.

Celles-ci sont principalement utilisées pour la réalisation des prototypes permettant d'obtenir les évaluations. C'est notamment le cas des tests logiciels qui vont utiliser la méthode de l'injection de fautes pour déterminer les valeurs à assigner aux attributs de la norme qualité.

Les informations sur les conditions d'acquisition et critères socio économiques, définies dans le paragraphe précédent, servent à rechercher les informations auprès des fournisseurs pour départager les candidats à l'intégration. Par contre, elles ne sont pas utilisées directement dans la construction des prototypes. Les informations ainsi récoltées servent à la sélection des COTS par une phase plus spécifique de tri qui démarre le plus souvent par la comparaison des informations recueillies sur les COTS.

V - 2.6 Pour étudier l'adaptation des COTS

C'est une étude plus approfondie sur la manière d'intégrer le produit à l'application. Les informations utilisées dans les phases précédentes pourront être affinées ou complétées dans la phase d'adaptation. Comme pour la comparaison des COTS, il est souhaitable que les informations soient disponibles dans toutes les phases du processus de développement de l'application à base de COTS permettant d'affiner ainsi l'ensemble des résultats obtenus pendant ce processus.

La réalisation d'un prototype dans cette phase permettra de manipuler principalement les informations techniques et les services des COTS. C'est avec ces informations que l'intégration sera réalisée par la suite lors d'une phase d'assemblage. L'ensemble des difficultés et des scénarios de bon fonctionnement sera établi pour s'assurer du choix des

COTS qui seront intégrés et de la faisabilité de l'assemblage. Ces informations serviront dans la phase suivante de « tri et de sélection de COTS » afin de valider les choix déjà réalisés dans une autre itération et d'établir la nécessité de réaliser une itération supplémentaire.

V - 3 *Bilan*

La réalisation du prototype est l'une des tâches principale des phases « d'évaluation et de qualification » et « d'adaptation » des COTS et fait partie du cycle itératif comme nous venons de le voir. La Figure 11 correspondant à notre vision du processus de développement à base de COTS dans laquelle nous voulons apporter notre contribution :

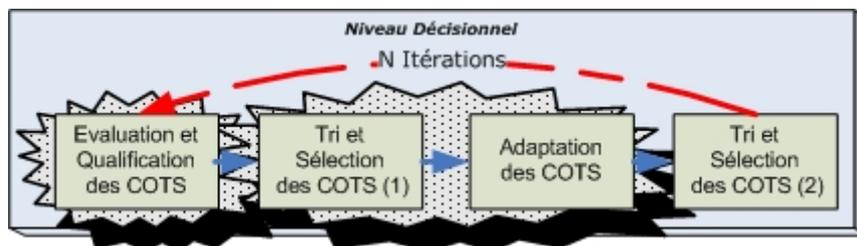


Figure 11 : Phases du processus de développement nécessitant la réalisation du prototype

Les informations recueillies sur les phases amont sont nécessaires pour la réalisation du prototype mais aussi dans les différentes itérations qui auront lieu pendant le processus de développement.

Ce prototype est devenu une nécessité pour le recueil des informations nécessaires aux évaluations des risques et des coûts. Il est une aide pour l'utilisation des COTS, et c'est aussi un outil présent dans la quasi-totalité des phases du processus de développement d'application à base de COTS comme nous avons pu le voir. Il est de fait, un maillon essentiel pour le développement de l'application.

Il est donc naturel de présenter cet outil comme un élément permettant de réduire certaines difficultés rencontrées lors de l'utilisation des COTS. En particulier, il permettra de s'assurer de la cohérence entre les besoins utilisateurs et les services proposés par les COTS, mais aussi de vérifier la faisabilité d'une telle intégration.

C'est donc, à l'aide du prototype, de son utilisation dans un processus de développement et des outils d'évaluation qu'il est possible de maîtriser son intégration et donc de réduire les difficultés de son utilisation.

VI Conclusion

Si l'utilisation des COTS dans un contexte de développement rapide d'applications de plus en plus complexes est une nécessité, les difficultés rencontrées peuvent devenir un frein à son utilisation. L'utilisation d'un processus adapté avec des phases spécifiques à ce type de développement ont permis de traiter les difficultés spécifiques. Pour cela des outils d'évaluation des risques et des coûts ont été créés. Ils permettent aussi de limiter les difficultés.

Cependant, la plupart de ces aides ne semblent pas suffisantes. On ne remet pas en cause tous ces moyens, pourtant, il apparaît une difficulté majeure : c'est le recueil des données [Bertoa03]. Elles sont indispensables dans tous les moyens mis en œuvre pour aider les développeurs à faire l'intégration des COTS. Mais le constat de Bertoa sur le volume important d'informations à traiter et la pauvreté des informations disponibles, nous fait penser, que la problématique sur comment aider les développeurs dans l'utilisation des COTS, est principalement axée aujourd'hui sur les moyens leurs permettant d'obtenir et de traiter les données nécessaires au processus de développement et aux outils d'évaluations.

La réalisation de prototypes est une solution pour obtenir par l'expérimentation les informations nécessaires à ce processus de développement. De plus, elle permettra de certifier les données recueillies et donc de s'assurer de la qualité de ces informations. C'est donc une bonne solution, cependant, sa conception nécessite elle aussi un volume important d'informations qu'il est difficile d'acquérir. Bien souvent elles semblent être les mêmes que celles pour le processus de développement d'une application à base de COTS. C'est pour cette raison que nous vous proposons de les étudier dans le cadre du processus de développement et en particulier dans la réalisation des prototypes permettant le recueil des informations sur les COTS et leur intégration. C'est à partir de l'aide que nous pourrons apporter au développement de ces prototypes que nous envisageons de réduire la difficulté majeure présenté par Bertoa ci-dessus.

Chapitre 3 : Les informations pour l'utilisation des COTS

I Introduction

La sélection et le choix sont les deux principales difficultés à l'utilisation des COTS. Cela se justifie par le fait qu'un mauvais choix entraîne des surcoûts, des problèmes de qualité voire même des problèmes dus à la non réalisation de toutes les fonctionnalités devant être implémentées. Pour ces phases de choix et de sélection, les développeurs vont essayer d'obtenir le plus d'informations possibles sur les produits. Cependant, le volume d'informations à traiter est particulièrement important comme nous le verrons par la suite. C'est pour cela que nous proposerons une classification afin de les manipuler plus facilement dans chaque phase du processus et d'établir un rapport de synthèse permettant de rassembler et de trier les données pour aider le développeur dans leurs manipulations.

Toutes ces informations seront manipulées ou recueillies à l'aide de prototypes. Ce dernier, comme nous l'avons vu dans [Leung02], est indispensable pour obtenir les informations permettant de réaliser le choix des COTS. C'est un moyen d'aider le développeur à acquérir ces informations pour la phase de choix et de sélection. De fait, il est lui aussi une étape importante dans le processus de développement d'une application à base de COTS. Pourtant, nous verrons que pour leur réalisation, il est nécessaire d'obtenir

et de manipuler, là aussi, un volume conséquent d'informations et que la difficulté est maintenant reportée en partie sur la réalisation de ces prototypes.

Il est donc indispensable de connaître les informations disponibles permettant la réalisation des COTS et d'avoir un moyen d'obtenir les informations manquantes.

II Les informations nécessaires pour le processus de développement

II - 1 Introduction

Les fournisseurs de COTS possèdent normalement les informations indispensables pour la réalisation de l'intégration par les développeurs. D'autres acteurs que nous appellerons ici la « communauté des COTS » ont, du fait d'expériences passées dans leurs intégrations, des informations indispensables pour les développeurs de l'application à base de COTS (historiques sur les difficultés rencontrées ou les configurations de bon fonctionnement). Si ces derniers (fournisseurs et communauté de COTS) sont plutôt fournisseurs d'informations, les développeurs d'applications à base de COTS sont plutôt les clients et demandeurs d'informations. Il apparaît donc une relation client / fournisseur entre différents acteurs de l'utilisation des COTS. Le client est le développeur d'applications à base de COTS. Les concepteurs et les vendeurs de COTS que nous regroupons sous le terme de « FourCo » (les vendeurs et les concepteurs de COTS) ainsi que la « communauté des COTS » sont les fournisseurs de cette relation. Dans tous les cas, nous décrivons ici le rôle porté par un acteur à un instant donné. En effet, le client fera partie de la « communauté des COTS » s'il fournit de l'information aux développeurs d'applications à base de COTS que nous avons identifiés ici comme les « clients ». Il y a de ce fait, un changement de rôle de la part de l'acteur en fonction de son comportement comme fournisseur ou client. C'est dans ce cadre là, que nous proposons notre représentation à l'aide d'UML de ces trois acteurs de COTS (Figure 12).

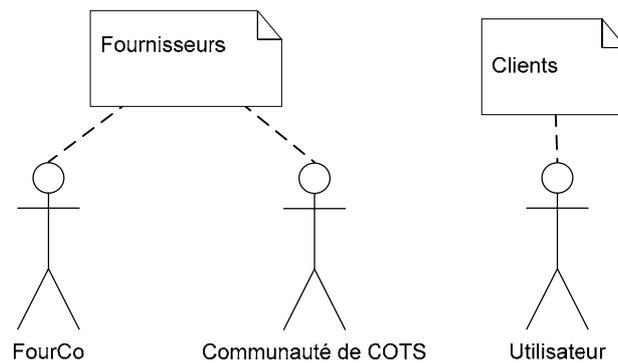


Figure 12 : Les acteurs de COTS

Ce besoin en informations est présent dans toutes les phases du processus de développement. Comme nous l'avons décrit, le nombre d'attributs à renseigner est important, notamment dans la phase de sélection et d'évaluation. De plus, certains sont utilisés dans plusieurs phases du processus. Nous prendrons un exemple simple parmi toutes ces informations : la description des services. C'est sûrement celle qui apparaît en premier avec la qualité des produits. Lorsqu'on recherche un COTS c'est dans un premier temps pour ses fonctionnalités puis pour ses qualités. La description des services est donc un point d'entrée important dans le processus.

Par contre, en fonction de la phase de développement, la granularité des informations concernant les services n'est pas la même. Elle est adaptée au contexte d'utilisation. Il en est de même pour toutes les autres données qui sont présentes dans les différentes étapes du processus comme par exemple les attributs de la qualité ou les critères de sélection.

De ce fait, il est nécessaire de filtrer les informations en fonction des phases du processus de développement et de leur utilité. Pour cela nous allons tout d'abord identifier les données disponibles puis nous classerons les informations en fonction des besoins et du contexte.

II - 2 Les données disponibles

La disponibilité des données est une des difficultés que [Bertoa03] a mise en évidence dans son étude : « le manque d'information sur les COTS est une réalité sur la quasi-totalité des produits ». Les développeurs d'application à base de COTS ont pourtant besoin d'informations pour réaliser toutes les étapes du processus de développement comme vu précédemment. Ils doivent impérativement rechercher par différents moyens les informations dont ils ont besoin. L'absence de données sur les produits ne peut

qu'augmenter les risques et les coûts d'intégration des COTS comme déjà vu dans le Chapitre 2 : IV.

Les utilisateurs peuvent obtenir des informations en :

- sollicitant les vendeurs de COTS
- communiquant avec la « communauté des COTS »
- réalisant des prototypes qui sont nécessaires aux phases d'évaluations, de test qualité et d'estimations des coûts et des risques.

Nous proposons de regrouper dans un cas d'utilisation les activités conduisant à la recherche d'informations. Pour pouvoir les obtenir il faut tout d'abord que ceux qui en possèdent, c'est-à-dire les acteurs (Concepteur et communauté des COTS), puissent les mettre à disposition : c'est pourquoi nous introduisons une activité appelée « diffuser de l'information ». Elle est indispensable pour qu'une requête de recherche puisse aboutir à un résultat non nul pour l'acteur que nous avons appelé « utilisateur ». Si elle n'est pas disponible nous considérons que l'utilisateur ne peut pas y accéder. Celui-ci aura dans notre vision une activité directement de recherche qui correspond à la première phase du processus de développement à savoir la veille technologique, mais aussi dans d'autres phases du processus de développement. Dans le cas d'utilisation présenté dans la Figure 13 nous proposons que les activités de « description d'application », de « réalisation d'un prototype », « d'évaluation d'un COTS » et « d'estimation des risques et des coûts » utilisent directement l'activité de « recherche d'informations ». Ce sont des activités que l'utilisateur a besoin de réaliser dans le cadre du processus de développement d'une application à base de COTS. Dans le cas de la description d'une application, l'utilisateur a besoin des informations issues de sa veille technologique et des produits qui permettront sa description comme présenté dans [Botella02] - Figure 5. L'activité « d'estimation des risques et des coûts » correspond à une phase importante permettant l'évaluation et la sélection des COTS dans son processus d'intégration. Pour cela il est nécessaire d'obtenir des informations notamment pour l'activité « d'évaluation de COTS » que nous retrouvons aussi dans les phases du processus de développement d'application à base de COTS. Cette dernière aura besoin des informations dans le cadre des attributs de la norme ISO9126 qu'elle utilise mais aussi à travers d'une autre activité qui lui est indispensable pour obtenir les informations manquantes : la « réalisation d'un prototype ». Cette dernière nous intéresse plus particulièrement puisque nous voulons nous focaliser sur sa réalisation considérée comme importante dans le cadre de la recherche et l'obtention des informations [Leung02].

Notre représentation ci-dessous est notre réflexion en UML des activités conduisant à la recherche d'information et mettant en relation les clients et les fournisseurs. « L'utilisateur » en tant que client, sera l'acteur qui aura le plus besoin de cette activité de recherche. Cela démarre directement par la première phase du processus de développement déjà décrite d'une application à base de COTS (Figure 10) : la « veille technologique ». Cela se poursuit dans d'autres activités comme la réalisation d'un prototype, l'évaluation des COTS ou l'estimation des risques.

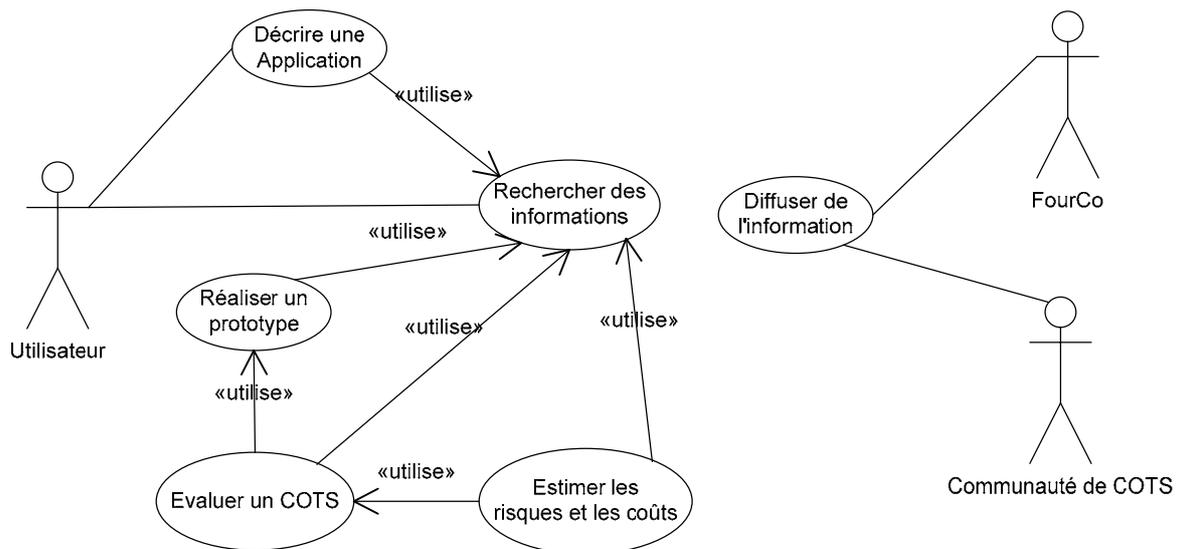


Figure 13 : cas d'utilisation pour obtenir des informations

D'autre part au niveau des fournisseurs et comme nous l'avons introduit, il ne peut y avoir de recherche d'information que si celle-ci est mise à disposition des utilisateurs. De fait, nous constatons que ce sont les vendeurs de COTS qui fournissent pour la plupart des documentations permettant d'obtenir des informations valables sur une ou plusieurs phases du processus de développement. On retrouve par exemple :

- Des documentations générales ou commerciales permettant d'identifier des fonctionnalités, des coûts de licences et quelques pré-requis utiles dans la première phase du processus,
- Des documentations techniques permettant de compléter une liste d'attributs de la norme ISO9126 utilisée notamment dans la phase d'évaluation,
- Des documents de développement (exemple de code, scénario d'intégration, de déploiement, etc.) permettant de réaliser l'assemblage du COTS dans une application, et correspondant à la phase d'adaptation du processus de développement.

La liste des documents est variable selon les vendeurs et les produits. Nous avons cité ici les plus couramment rencontrés.

Pourtant, si toutes les informations ne sont pas trouvées chez les vendeurs de COTS, le développeur va devoir chercher les données manquantes chez les autres utilisateurs. Pour cela il peut utiliser les forums de discussions, les séminaires pour développeurs ou tout autre endroit où il sera possible d'échanger de l'information ou de trouver de l'aide. C'est l'expérience des autres développeurs qui est recherchée avec :

- Les scénarios de bon fonctionnement,
- Les scénarios de dysfonctionnement,
- Les difficultés rencontrées et leurs solutions.

Tous ces échanges et toutes les informations récoltées serviront le processus de développement de l'application à base de COTS, et en particulier la réalisation du prototype à partir duquel il sera possible de compléter les informations nécessaires à chaque phase de ce processus. On se propose dans le paragraphe suivant de classer les informations et de les affecter aux différentes phases du processus.

II - 3 Les données nécessaires

Dans chaque étape du développement, nous pouvons identifier des familles d'informations qui y sont nécessaires. Nous entendons par famille l'ensemble des informations que l'on peut rassembler sous un même terme et objectif.

Nous reprenons dans ce chapitre les données initialement traitées dans le chapitre de l'état de l'art. L'objectif de cette section est de classer les informations en fonction des familles d'informations que nous allons définir. Nous établirons que ces familles d'informations sont réutilisées dans différentes phases du processus de développement à des niveaux de détail ou pour des objectifs différents. C'est aussi pour nous le moyen de présenter ces informations dans notre future proposition qui intégrerait la saisie et le recueil de ces données.

II - 3.1 Pour la veille technologique et la définition des besoins

Dans la première phase présentée dans [Botella02], les données nécessaires sont :

- l'identification des services des COTS et de l'application en cours de définition,
- la description générale des produits (services, plate forme d'utilisation, licences, coûts),
- les attributs de la norme ISO9126 pour la qualité logicielle,
- les critères de sélection en fonction des objectifs de réalisation de l'application à base de COTS.

L'identification des services et la description générale servent à identifier le produit et les éléments techniques nécessaires à son fonctionnement. On retrouve ces informations dans les « Données Produit » dont voici un extrait de la maquette du démonstrateur que nous présenterons au chapitre 4 en fonction des phases appelées « veille technologique » (Figure 14) et « définition des besoins » (Figure 15).

Veille Technologique : Données Produit

Nom du Produit : Microsoft Outlook

Catégorie du Produit : Client de messagerie

Principaux Services

- Envoi et réception de courrier
- Carnet d'adresses
- Agenda
- Autres services ...

Licence : Payante

Plateforme : Windows

Figure 14 : Extrait des « Données Produit » pour la veille technologique

Dans l'exemple ci-dessus, les informations fournies doivent permettre d'identifier de manière générale les principaux services rendus par le produit. A ce niveau, il n'est pas nécessaire d'obtenir des détails sur le fonctionnement de chaque service, ni sur des caractéristiques techniques ou de qualité. Ces dernières informations seront recherchées ultérieurement car le but de cette phase est avant tout d'identifier des candidats et les macro-fonctionnalités de chaque produit. Avec ce niveau de détail, les développeurs vont faire une première sélection, étudier les services rendus par tel ou tel type d'application et

l'expertise plus détaillée sera réalisée dans une phase ultérieure. D'ailleurs, lorsque l'utilisateur va rentrer dans sa phase de définition des besoins, il devra s'intéresser de plus près aux COTS pour identifier les services indispensables à son application que ces derniers fournissent habituellement dans un domaine ou métier. C'est le cas dans la Figure 15 où les sous-services sont maintenant détaillés pour que l'utilisateur puisse identifier l'ensemble des services et sous services du COTS.



Figure 15 : Extrait des « données produit » dans la phase de définition des besoins

Les utilisateurs, au fur et à mesure qu'ils avancent dans le processus de développement, cherchent à obtenir plus d'informations sur les COTS étudiés. Pour cela, ils vont s'appuyer sur les attributs de la norme ISO9126 de la qualité logicielle. De fait, nous pouvons créer une deuxième famille que nous appellerons les « données qualité » et qui va recenser l'ensemble de ces attributs.

Enfin, à partir des besoins nous avons les critères de sélection. Ils utilisent les informations issues des spécifications et des besoins pour la réalisation d'un projet de développement d'une application à base de COTS. Nous nommerons l'ensemble de ces informations, les « données de spécification » avec dans la Figure 16 un extrait de ces données. Dans ce cadre là, la fiche est comparable à la fiche produit. Nous retrouvons la catégorie du produit, les principaux services et sous-services, ainsi que la ou les plateformes cibles. Nous complétons les informations avec le niveau de priorité sur ces exigences : Indispensable, Non prioritaire et Optionnel. Le niveau « Indispensable » correspond à une obligation de résultat. L'absence de ce service par exemple est critique pour le bon fonctionnement de l'application. Il est donc impératif de le réaliser. Le niveau « Non prioritaire » ne remet pas en cause le fonctionnement de l'application, mais il était souhaitable de la part des développeurs ou des futurs utilisateurs que le service soit présent.

La décision de ne pas le réaliser nécessite une modification des spécifications et l'accord du client pour qui l'application est développée. Enfin, le niveau « Optionnel » correspond aux services qu'il est communément admis comme ne faisant pas partie de la fourniture dès la première version pour des raisons de délais ou de difficultés de réalisation par exemple. La décision de ne pas l'inclure ne nécessite pas une prise de décision unilatérale entre développeurs et clients : une simple information du client est suffisante.



Définition des spécificatins pour l'application : Données spécification

Notre Application

Nom du Produit : Notre Application à base de COTS

Catégorie du Produit : Catégorie X

Principaux Services

- Service 1
- Carnet d'adresses
- Service 2
- Autres services ...

Licence : Free

Plateforme Cible : Windows

Figure 16 : Extrait des « données de spécification »

En résumé, nous identifions trois familles d'informations différentes (Figure 17). Nous proposons de les nommer, les données :

- « Produit » permettant de connaître le fonctionnement des COTS,
- « Qualité » relatives à la norme ISO9126 principalement,
- « De spécification » issue de la définition des besoins et de spécification.

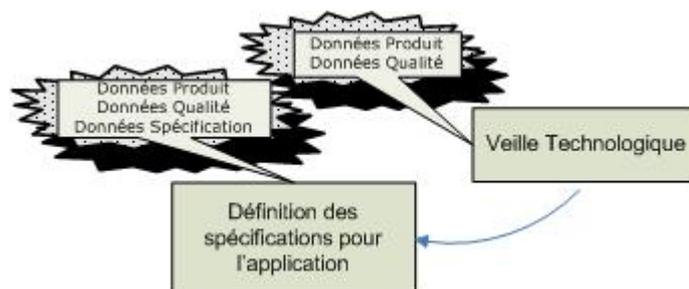


Figure 17 : Familles d'informations de la phase 1

Si on retrouve des familles identiques (« Données Produit » et « Données Qualité ») dans la « veille technologique » et la « définition des spécifications pour l'application », les données ne sont pas de même niveau. Par exemple lors de la veille technologique, les « données produit » correspondent à un niveau de granularité assez élevé ou seulement les grandes fonctionnalités sont recherchées. Alors que dans les « données produit » correspondant à la « définition des spécifications pour l'application », ces mêmes fonctionnalités vont être décomposées en plusieurs services et sous services et donc avec une granularité plus fine comme nous pouvons le voir dans la Figure 19.

Dans l'étape « définition des spécifications pour l'application », les développeurs utilisent les « données de spécifications » pour établir la cartographie de l'application qu'ils veulent réaliser. On y trouve les mêmes informations que dans les « Données Produit » mais elles correspondent aux besoins et aux contraintes. Lorsqu'on décrit un service, c'est pour qu'il soit présent dans l'application finale. Lorsqu'on décrit plusieurs plates-formes cibles, on définit les environnements d'exécution supportés par l'application ce qui deviendra une contrainte pour la phase suivant d'acquisition et de sélection des COTS car ils doivent impérativement fonctionner sous ces environnements cibles.

II - 3.2 Pour l'acquisition et la sélection des COTS

En fait, on s'aperçoit que l'information (l'identification des services) est reconduite dans cette phase mais qu'elle est traitée à un niveau de granularité différent, plus en accord avec les objectifs de l'étape autour de l'utilisation de la norme qualité citée ci-dessus.

Dans le paragraphe Chapitre 2 : IV, nous avons présenté plusieurs processus itératifs d'acquisition des COTS et les activités communes à tous ces processus, dans lesquels nous pouvons identifier des informations reconduites dans plusieurs phases. C'est notamment le cas de l'identification des services cités ci-dessus, des attributs de la norme ISO9126 pour les tests qualité et les critères de sélection. C'est dans ce cadre là que nous proposons de continuer à utiliser les trois familles de données précédentes.

En fait, les données Produit, Qualité et Spécification sont maintenant utilisées, puis complétées, pour la réalisation de cette phase (Figure 18). Alors que précédemment nous étions dans des phases de définition et de veille que nous regroupions sous le terme de niveau stratégique, la phase « d'acquisition et sélection » va devoir être plus opérationnelle (d'où son appartenance à un « niveau opérationnel ») car elle va exploiter les informations précédemment recueillies dans les trois familles de données pour déterminer si les produits étudiés dans les phases précédentes méritent d'être sélectionnés

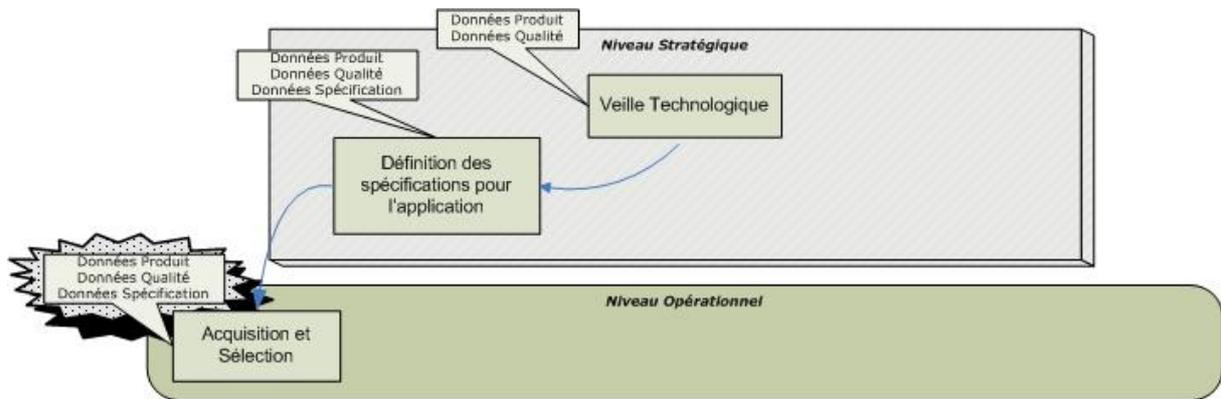


Figure 18 : Les informations de la phase d'acquisition et de sélection des COTS.

Dans cette phase les « Données Produit » doivent permettre d'identifier les premières informations sur le fonctionnement des COTS, les moyens de développement nécessaires et les services de formations, d'assistance et leurs coûts (Figure 19).

Acquisition et Sélection : Données Produit

Nom du Produit : Microsoft Outlook

- [-] Carnet d'adresses
 - [-] Fiches contacts
 - [-] Listes des champs d'une fiche
 - [-] Autres ...
 - [-] Autres ...

Coût de la Licence : xx euros

Services associés au produit : Support technique, Formation, ...

Plateforme et configurations : Windows + Internet Explorer

Services associés au produit : Support technique, Formation, ...

Plateforme de développement : Langages et outils de développement possibles

Technologie utilisée : Bibliothèque .Net

Figure 19 : Extrait des « données produit » dans la phase d'acquisition et de sélection

Ces éléments peuvent se retrouver dans les critères de sélection qui seront définis dans la famille de données que nous avons appelée les « données de spécification ». La comparaison des informations entre les besoins (« données de spécification ») et ce que proposent les produits à intégrer (« données Produit ») est l'activité principale de la phase d'acquisition et de sélection. Le fait qu'un COTS propose des services définis comme indispensables à l'application est un minimum pour qu'il soit sélectionnable comme nous le montre la fiche de données de spécification de cette phase (Figure 20).

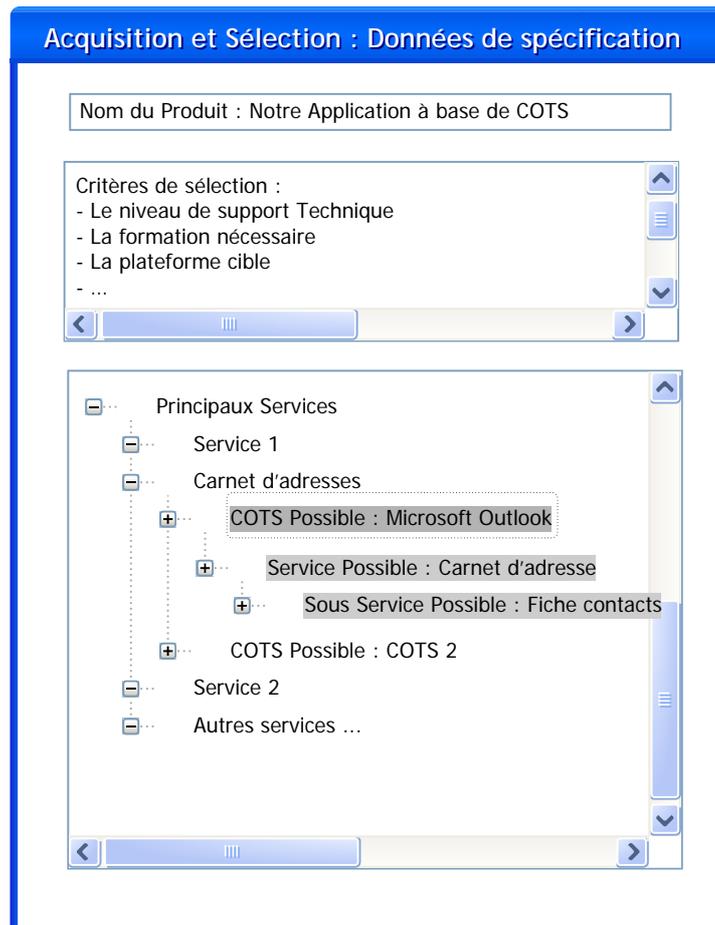


Figure 20 : Extrait de « données de spécification » avec l'identification des services disponibles dans le COTS pour l'application

Dans la Figure 20 nous identifions en particulier les services qui sont réalisables par des COTS (dans notre exemple deux COTS sont candidats pour remplir le service du « Carnet d'adresse »). Nous utilisons pour cela les informations issues des « Données Produit » qui sont enrichies en fonction du niveau de granularité nécessaire pour identifier le COTS à sélectionner et passer à la phase suivante de qualification.

II - 3.3 Pour la qualification des COTS

C'est la première phase permettant d'identifier et de décider sur les COTS qui seront utilisés dans l'application à base de COTS. C'est pour cela que nous avons regroupé l'ensemble de ces phases sous un niveau que nous avons nommé « niveau décisionnel » (Figure 21).

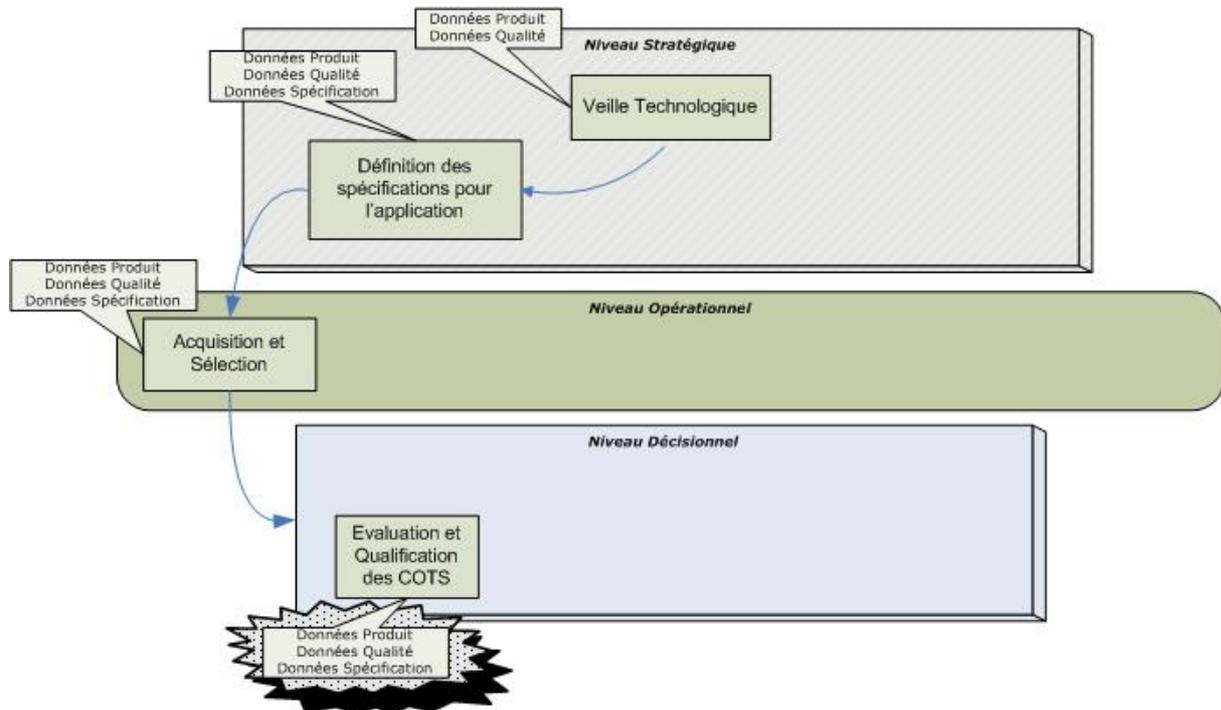


Figure 21 : Les informations de la phase de qualification et d'évaluation des COTS.

Les deux premières familles de données («Données produit et Qualité) utilisées dans la première phase du processus de développement (Veille technologique) sont reconduites dans cette phase. En référence à notre description de l'étape dans le paragraphe Chapitre 2 : IV - 4 (La phase de qualification), les « données produit » sont utilisées pour déterminer les services présents et manquants dans le COTS par rapport aux besoins de l'application en les comparant avec les « données de spécification ». Dans ces dernières nous trouvons aussi les critères de sélection en fonction des normes qualité. Elles sont utilisées avec les « données qualité » pour établir une évaluation et une qualification.

Pourtant, les informations au niveau des « données produit » et des « données qualité » sont souvent incomplètes ou insuffisantes pour faire l'évaluation et la qualification. Les développeurs vont devoir dans cette phase recueillir les informations manquantes. Ils demanderont ces informations auprès du fournisseur du produit et de la communauté des COTS mais aussi en réalisant un prototype qui lui permettront d'obtenir par l'expérimentation les données manquantes.

Nous proposons de compléter les familles de **données produit** et **qualité** pour établir une cartographie plus précise du COTS. Avant de passer à la phase suivante, une étape de tri et de sélection sera réalisée en fonction des résultats de cette étape et permettra de valider le choix des COTS qui a été faite dans la phase précédente. Cela aura pour conséquence de réduire les COTS qui seront étudiés dans l'activité suivante (l'adaptation des COTS) dans ce niveau décisionnel. Ainsi il sera possible de vérifier techniquement comment l'assemblage va pouvoir être réalisé et d'estimer la faisabilité et le coût de l'intégration.

II - 3.4 Pour l'adaptation des COTS

Dans cette étape, contrairement aux autres, les acteurs vont utiliser les informations récoltées dans les autres phases pour la réalisation des prototypes. Ce sont essentiellement les informations provenant de la famille des « données produit » (Figure 22) qui seront utilisées pour réaliser les scénarios d'adaptation qui sont les principaux objectifs de cette phase. On cherche avant tout à connaître des exemples d'environnement de développement et des exemples d'intégration ainsi que toute la documentation nécessaire à la réalisation de cette phase.

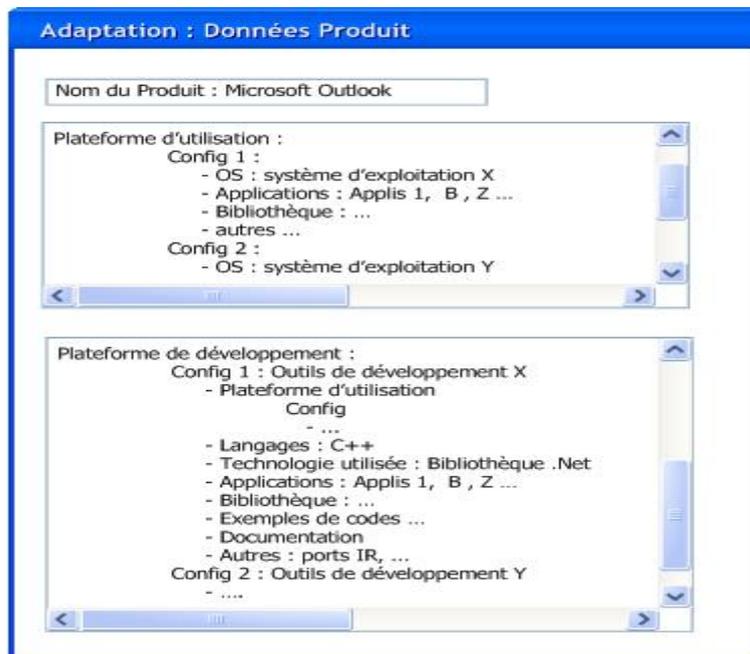


Figure 22 : Fiche Données produit dans la phase d'adaptation

Les scénarios d'intégration décrits dans les « Données Produit » peuvent être incomplets ou ne correspondant pas à l'adaptation voulue par le développeur. Dans ce cas, il doit être possible d'introduire dans les « données produit » d'autres scénarios qui enrichiront la base d'information que constitue le Système d'Information de COTS.

Comme dans l'exemple ci-dessus où deux scénarios appelés « config 1 » et « config 2 » correspondent à deux environnements de développement permettant l'intégration du COTS « Microsoft Outlook ». Cela sera utile notamment dans le cadre de la réalisation d'autres projets si l'un de ces scénarios doit être réutilisé.

Cette phase d'adaptation doit permettre aussi de réaliser des évaluations de coûts et de risques que nous avons présentés dans le Chapitre 2 : IV - 5. Pour cela il est nécessaire d'établir une liste d'informations utiles pour ce travail que nous appellerons les données d'estimations. De ce fait, nous proposons de rajouter cette famille d'informations (« données d'estimation ») dans notre schéma regroupant les informations traitées par chaque phase du processus de développement.

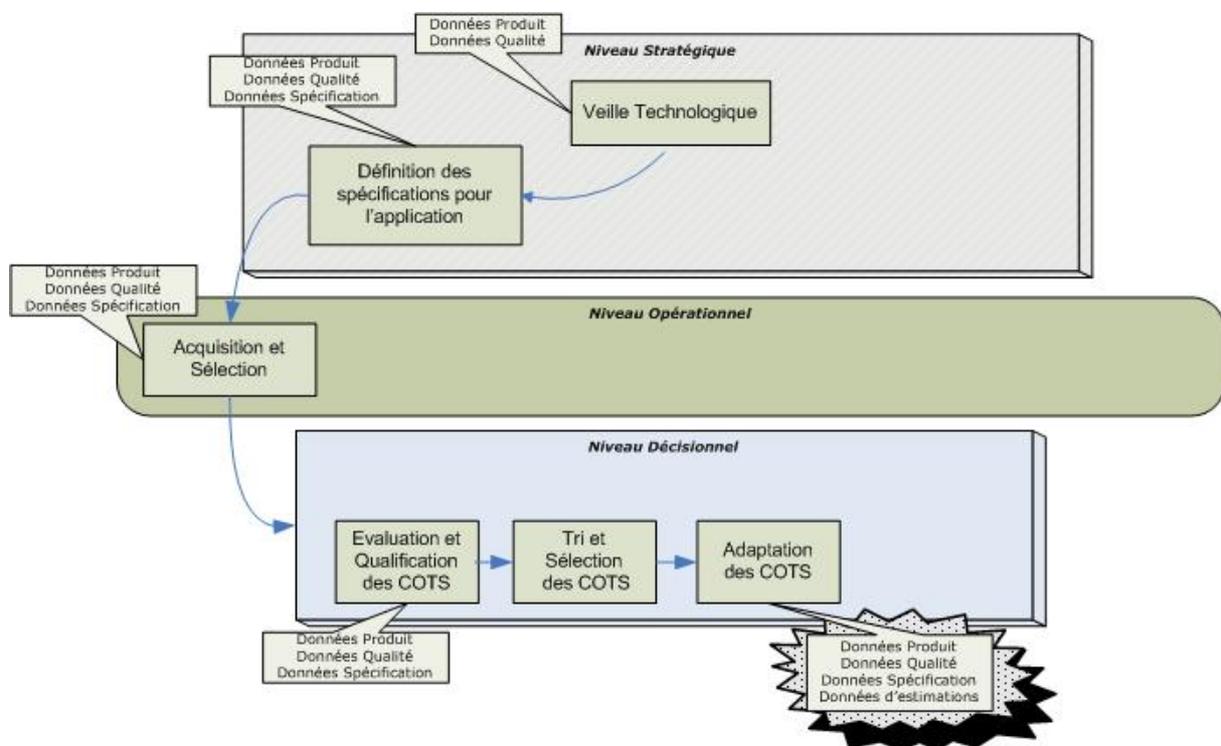


Figure 23 : Les informations de la phase d'adaptation des COTS.

Les « données d'estimations » comme les autres données pourront être enrichies à chaque itération dans le niveau décisionnel (Figure 24). Si certaines données sont recueillies pendant la phase d'évaluation et permettent de compléter les données d'estimations elles permettront d'établir un nouveau calcul de l'estimation des coûts et des risques. Les phases de « Tri et de Sélection des COTS » bénéficieront du nouveau calcul proposé par les formules de COCOTS et COCOMO (que nous présenterons dans le Chapitre 3 : III - 4.3) et donc il sera possible d'intégrer le résultat à la décision de sélection des produits.

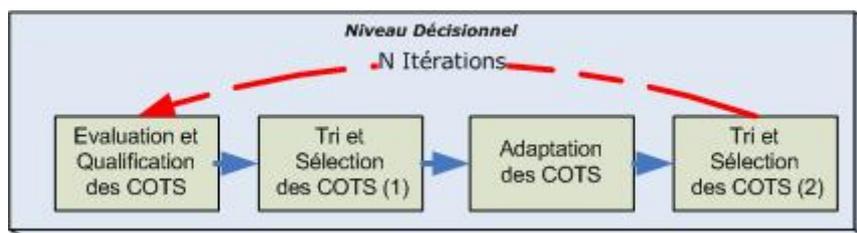


Figure 24 : Itérations au niveau décisionnel

On remarque que dans les différents articles nous ayant permis de décrire les différentes phases du développement à base de COTS, la famille des « données d'estimation » n'est pas utilisée dans toutes les phases. C'est peut être la difficulté de rassembler les informations pendant la phase d'adaptation et donc le temps nécessaire à un nouveau calcul de ces estimations qui réduit l'intérêt d'utiliser cette famille d'informations. Pourtant, il nous semble que les informations relatives à cette famille pourraient être affinées à chaque itération comme nous venons de le présenter en utilisant les informations recueillies dans chaque étape de ce processus.

Nous proposons donc que l'ensemble des quatre familles d'informations soient recensées. Cette combinaison permettra de définir la faisabilité de l'assemblage et de choisir au mieux les COTS qui seront assemblés dans l'application à base de COTS. En fait, nous ajoutons cette famille d'informations (« données d'estimation ») dans toutes les phases du niveau décisionnel.

II - 3.5 Pour l'assemblage des COTS

L'assemblage est une étape opérationnelle (Figure 25) dans laquelle on va exploiter les données issues de toutes les phases amont. C'est aussi l'occasion de remplir la fiche des « données produit » qui permettront d'identifier les services fournis, la qualité du produit, toutes les informations nécessaires pour son utilisation et son intégration dans une application.

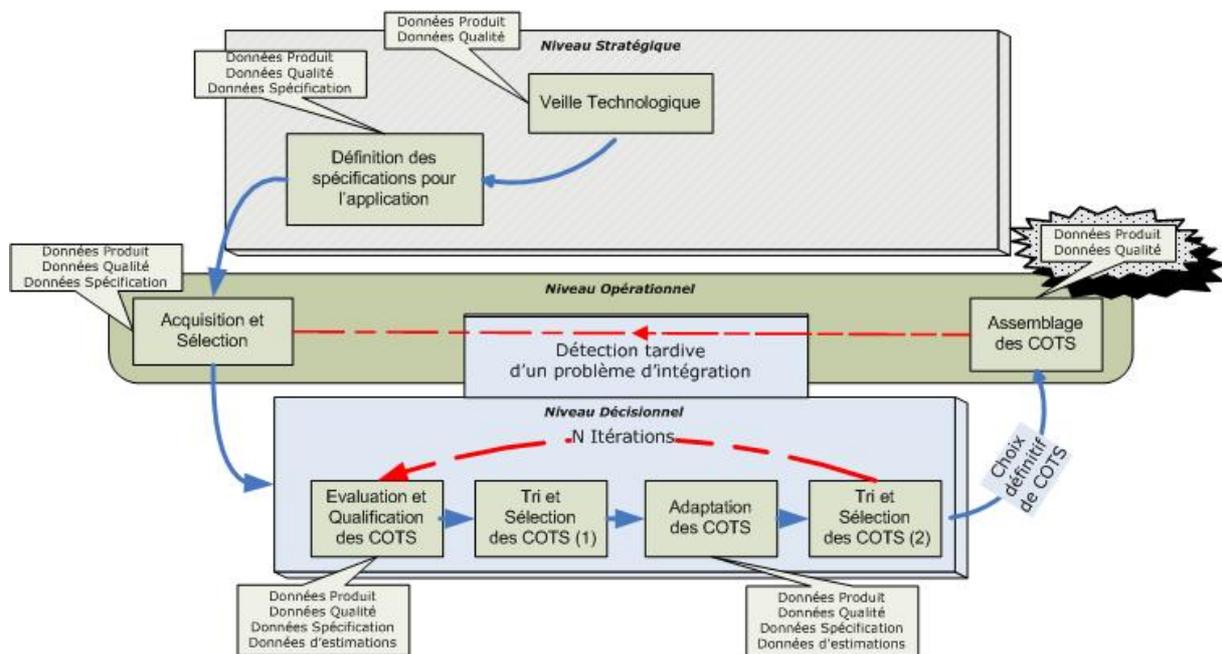


Figure 25 : Les informations pour l'assemblage des COTS.

On utilise, tout d'abord, les « données produit » correspondant à l'application à base de COTS qui doivent décrire l'ensemble des COTS qui sont définitivement sélectionnés pour cette phase. Cela permettra d'identifier d'une part l'ensemble des services qui restent à développer et d'autre part l'ensemble des COTS qui doivent être intégrés. Pour chacun d'entre eux, les « données produit » indiquent les environnements de développement et les moyens pour réaliser l'intégration. C'est avec ces informations recueillies pendant la phase d'adaptation que les développeurs mèneront à bien l'assemblage.

Là aussi, il est nécessaire de continuer à remplir les « données produit » pour identifier d'une part les conditions d'utilisation et d'autre part l'environnement de développement qui permettraient, si c'était le cas, d'intégrer ce produit dans une application plus importante encore. Ce sont aussi les « données qualité » issues des tests organisés pendant le développement et la mise en service qui peuvent être renseignées.

Toutes ces informations peuvent être utilisées dans le cadre d'un nouveau projet ou de la mise à jour de l'application (phase suivante du processus de développement). Cette capitalisation de l'information permettra notamment de réduire les spirales décrites par Bohem (Figure 4 : WinWin Spiral Model [Boehm02]) et donc un gain de temps mais aussi une réduction des risques par le fait de posséder des informations plus précises car déjà testées dans un précédent développement.

II - 3.6 Pour la mise à jour de l'application à base de COTS

Le développement d'une application ne se termine pas forcément avec la diffusion du produit. Il est habituel de la faire évoluer. C'est le rôle de la phase de mise à jour que de proposer une modification de la version en cours de validation.

Pour cela, il est nécessaire de reprendre la totalité du processus de développement à base de COTS. En référence à la description de cette première phase dans la Figure 5, la veille technologique et la bonne connaissance des COTS est une phase importante du processus. A chaque projet, et en déroulant les étapes du processus de développement à base de COTS, il est possible de recueillir les informations permettant de mieux connaître les produits déjà qualifiés et évalués dans d'autres projets. Les informations recueillies pendant les étapes d'un ancien développement sont utiles dans le cadre d'un nouveau projet. Elles permettent de ne pas refaire deux fois les mêmes erreurs et surtout la difficulté première de l'intégration des COTS (le manque d'informations) est amoindrie par le fait que l'on possède les informations en provenance d'autres expériences.

Il est donc important de capitaliser ces informations. Nous proposons le recueil de ces données dans les « données produit » et les « données qualité ». Ces deux familles d'informations sont présentes dans la phase d'assemblage et comme présenté dans le paragraphe précédent ; elles permettront de compléter les informations diffusées par les « fournisseurs » de COTS et d'établir les données pour l'application en cours de développement. Elles serviront pour comparer les informations des COTS utilisés dans la version en cours de validité et celles des futurs candidats.

L'utilisation dans cette phase des « données produit » et des « données qualité » (de la part des fournisseurs et de la capitalisation), mais aussi celles utilisés pour l'application actuelle. Ainsi nous pouvons présenter notre proposition rassemblant toutes les familles de données utilisables dans chaque phase du processus de développement et représentée dans la Figure 26.

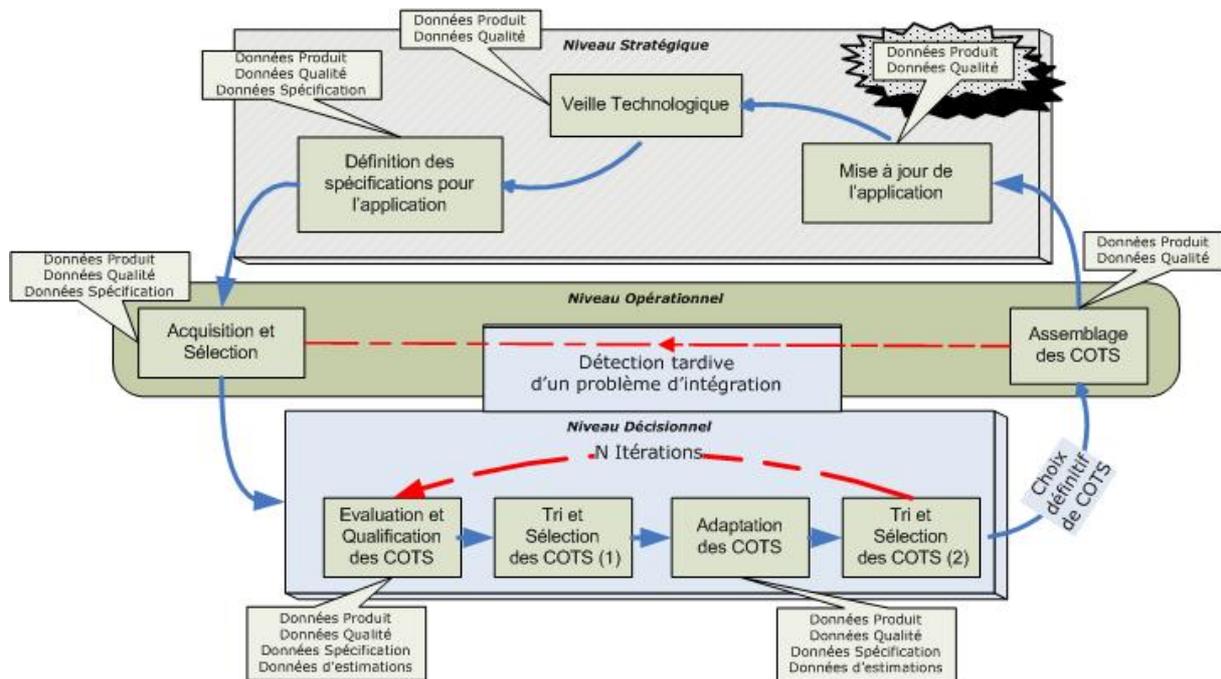


Figure 26 : Les familles d'informations en fonction des différentes phases du processus de développement à base de COTS.

Avec ce schéma nous avons les différentes familles d'informations utilisables dans chaque phase mais, pour aider le développeur à appréhender le volume important d'informations, il est indispensable de fournir des synthèses sur les données recueillies.

II - 4 La synthèse sur les informations recueillies

Cette synthèse doit permettre d'aider le développeur dans les décisions à prendre (par exemple sur le choix entre produits), d'établir la liste des informations nécessaires à la réalisation des différentes phases du processus de développement et notamment de lister pour chaque produit les informations recueillies et restant à traiter.

Les décisions prises dans les différentes phases du processus sont guidées par la comparaison des informations recueillies et par des outils d'aide à la décision comme par exemple le calcul de l'estimation des risques et des coûts. De plus ces informations sont comparées aux spécifications des besoins définis dès la première phase du développement de l'application à base de COTS. Notamment, il est détecté dans ces phases certains problèmes d'incompatibilité entre la solution apportée par le COTS et les besoins exprimés ou les contraintes définies au sein du projet. C'est par exemple une version du système

d'exploitation cible non supporté par le COTS ou un langage de programmation non connu par l'équipe de développement. Ce sont des informations faisant partie des critères de sélection et qui sont renseignées dans les « données de spécification » (Figure 20).

Si certains problèmes détectés peuvent être bloquants (l'exemple du système d'exploitation cible), d'autres sont liés à des stratégies d'entreprise et des investissements pour le projet comme l'exemple d'un langage de programmation à acquérir pour réaliser l'intégration. Dans tous les cas, il n'est pas possible pour le rapport de synthèse, de prendre la décision en lieu et place du développeur. Il est donc nécessaire de proposer les informations et d'alerter sur des incompatibilités comme le ferait un système de mesure qui connaîtrait des seuils à ne pas dépasser.

De ce fait, le rapport de synthèse doit pouvoir, en fonction de l'étape du processus et du type d'information nécessaire, proposer les données pertinentes. Nous proposons que ce rapport fournisse :

- la liste des données correspondant à chaque étape du processus et à une activité dans ce processus,
- la liste des données manquantes ou non identifiées dans une activité appartenant à une étape du processus,
- les incompatibilités qui seraient détectables (en comparant avec la famille de « données de spécifications »),
- un comparatif entre produits, si les données le permettent.

II - 5 Bilan

La difficulté d'utilisation des COTS est souvent proportionnelle à la difficulté d'obtenir des informations sur ces derniers. De ce fait, les évaluations et les estimations sont de moins bonne qualité par manque de données suffisantes. Cela entraîne la sélection de produits avec plus de risques pour le déroulement du développement de l'application à base de COTS.

Nous vous avons proposé une classification des informations en quatre familles qui sont :

- «Données Produit » permettant de connaître le fonctionnement des COTS,
- « Données Qualité » relatives à la norme ISO9126 principalement,

- « Données de Spécification » issues de la définition des besoins et de spécification et
- « Données d'Estimations » regroupant les données nécessaires aux estimations de coûts et de risques.

Puis nous les avons affectées aux différentes phases du processus de développement comme proposé dans la Figure 27.

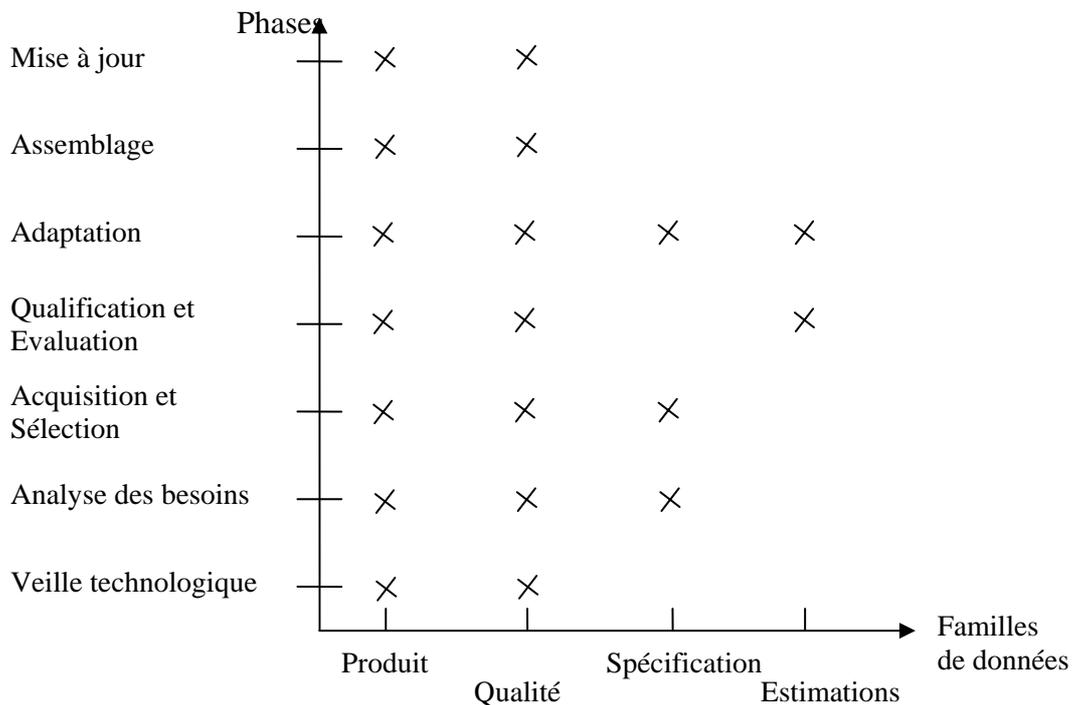


Figure 27 : Utilisation des familles d'informations dans les phases du processus de développement

Tout cela doit permettre d'identifier les informations en fonction de leur utilité dans le processus de développement lors de leur présentation sous la forme d'un rapport de synthèse. Ce rapport, comme nous l'avons proposé, doit permettre d'identifier les informations manipulées avec les résultats obtenus ou restant à obtenir. Dans ce cadre là, l'utilisation des familles d'informations et leur classement en fonction des phases du processus de développement doit permettre de visualiser plus rapidement les informations relatives à chaque étape du processus : celles obtenues ou restant à obtenir.

Pour acquérir les informations manquantes, l'utilisation des prototypes est nécessaire. Pourtant pour cette réalisation, il est nécessaire de posséder des informations spécifiques à leurs développements. Bien souvent les informations utiles pour la réalisation du prototype sont indispensables aux processus de développement et permettront de compléter l'ensemble des données Produit, Qualité et d'Estimation.

III Les informations nécessaires pour l'intégration d'un COTS dans un prototype

III - 1 Introduction

La réalisation d'un prototype doit permettre d'identifier un certain nombre d'informations comme nous l'avons vu précédemment. Elles doivent permettre de réaliser les évaluations, l'estimation des coûts et l'adaptation nécessaires à la comparaison et à la sélection des COTS comme présentés dans la Figure 28. Elle complète ainsi notre modélisation des activités menées par les différents acteurs du processus de développement à base de COTS.

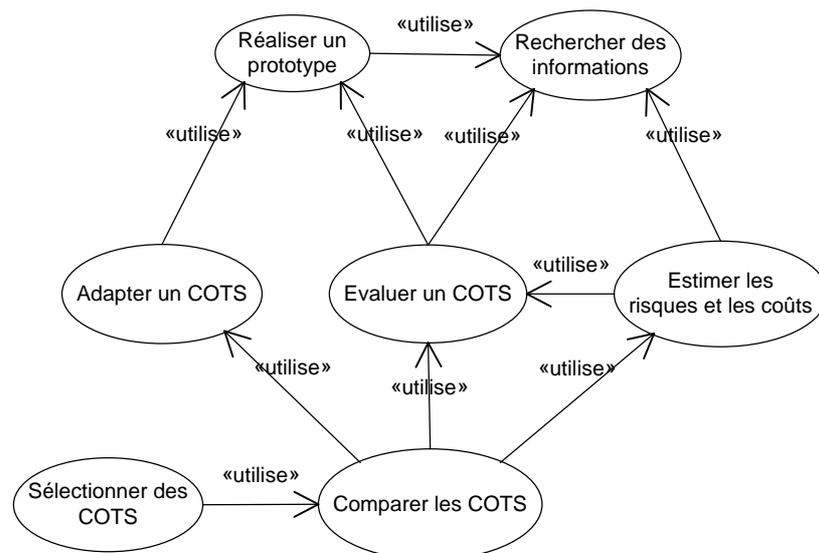


Figure 28 : Réalisation d'un prototype

Pour cela, les développeurs vont devoir réaliser techniquement le prototype. La programmation de ce dernier nécessite des connaissances techniques sur le produit à intégrer et les technologies employées par ce dernier.

On distingue deux types d'informations utiles lors de la réalisation du prototype et pour les activités dont il dépend :

- Le scénario d'intégration
- Le scénario d'incompatibilité

Le premier doit permettre la réalisation des procédures d'assemblages du COTS. C'est directement l'un des objectifs de la phase d'adaptation des COTS, l'autre objectif étant de fournir des informations utiles pour l'évaluation des COTS et l'estimation des coûts et des risques.

Dans le second cas, il sera intéressant de remonter des informations pour les notions de risques et notamment dans le cadre d'un assemblage de plusieurs COTS différents. Des incompatibilités ou des contraintes techniques d'utilisation peuvent empêcher l'assemblage simultané d'une combinaison de COTS. Ce qui augmente le risque d'échec de l'assemblage dans la dernière phase de ce processus et donc la réalisation de l'application.

La détection précoce de ces difficultés sur les incompatibilités et les contraintes techniques doit permettre d'invalider le choix du COTS pour l'intégration. Il est donc important lors de la réalisation des prototypes d'identifier les deux types de scénarios (compatibilité et incompatibilité) afin de mieux servir le processus de développement. L'identification de ces difficultés nécessite aujourd'hui une ou plusieurs itérations complètes du « winwin spiral model ». C'est donc leurs connaissances qui permettent d'éviter ces itérations et donc de gagner du temps qui peut être employé à une meilleure appréciation des risques et des coûts par exemple. Tout cela ne peut qu'améliorer le processus de développement.

Au-delà de l'utilité de ces deux types de scénarios, chacun d'entre eux doit répondre aux mêmes exigences. Une application, comme un prototype, nécessite la mise en œuvre d'environnements lui permettant d'exister dans les phases de création et d'exécution. Ces dernières sont souvent guidées par les besoins ou les contraintes définies en amont du projet. L'environnement d'exécution est indispensable à toute application. Il est donc nécessaire pour les outils de développement pour l'intégration des COTS. Dans le cadre du développement d'une application, il est indispensable de connaître les langages, les technologies et les méthodes d'intégration permettant sa réalisation. Pour cela les outils de la connaissance (formations, documentations, exemples de codes ou forums) doivent permettre d'aider le développeur dans sa réalisation du prototype. D'autant plus que dans

le cadre de l'utilisation des outils d'estimation des risques et des coûts d'intégration des COTS, cette notion est nécessaire pour faire le calcul.

A partir de ce constat, nous proposons, pour les scénarios, quatre types d'informations :

- L'environnement d'exécution du COTS,
- L'environnement pour l'intégration du COTS,
- L'environnement d'exécution de ces outils,
- Les connaissances techniques nécessaires pour l'intégration (langages, technologies, exemples d'intégration) que nous présentons dans les paragraphes suivants.

III - 2 Le scénario d'intégration

III - 2.1 Un environnement d'exécution du COTS

Tout logiciel a besoin d'un environnement pour fonctionner. Il n'est pas exclu de combiner plusieurs environnements pour permettre l'exécution d'un produit.

Parmi les logiciels servant d'environnement, nous avons un type de produit qui est indispensable pour tous les logiciels (sauf pour lui-même) c'est le système d'exploitation et sa version (SE). S'il n'est pas rattaché en tant qu'environnement directement au COTS, il est forcément rattaché au produit servant d'environnement à ce même COTS. Dans l'exemple d'un programme « Java » il n'est pas nécessaire d'avoir un SE mais il demande l'utilisation d'une machine virtuelle qui elle-même a besoin de ce système d'exploitation. Dans tout les cas, il est indispensable de l'avoir dans l'environnement final.

L'environnement matériel peut être lui aussi nécessaire pour l'exécution d'un COTS comme pour

- un type de carte vidéo (ayant la possibilité de traiter de la 3D),
- une connexion avec un périphérique spécifique comme le port série, USB ou autres,
- un système de fichier (l'adressage des informations ne pouvant se faire que si le système de fichier le permet comme dans le cas des différences entre de la « Fat16 », « Fat32 » ou « NTFS »),

- une carte matérielle spécifique permettant le traitement indispensable au COTS comme la décompression « MPEG2 » par exemple.

Dans les mêmes conditions, nous avons à définir les environnements de tous les outils nécessaires à la mise en œuvre de l'intégration du COTS dans l'application comme nous allons le voir dans le paragraphe suivant.

III - 2.2 Un environnement pour l'intégration du COTS

Pour réaliser l'intégration d'un COTS, on a besoin d'un certain nombre d'éléments. Ils peuvent être, là aussi, logiciels ou matériels

Ils dépendent de la manière dont on devra réaliser l'intégration. [Vidger97] présente trois méthodes possibles : « Wrapping », « Glue Code » et « Tailoring ».

Du fait de la forte différence des méthodes d'intégration, les environnements d'intégration peuvent être très différents. Cependant ils ont tous un point commun, ils sont indispensables pour la réalisation de l'intégration et aucun d'entre eux ne peut se passer d'au moins un environnement qui dépendra de la méthode utilisée.

Les « wrappers » sont décrits comme des composants standards permettant l'intégration. En exemple, Vidger cite CORBA et ODBC. Ils ont besoin pour l'environnement d'intégration de logiciels ou de fichiers spécifiques. Par exemple dans le cas d'ODBC, un « driver » compatible avec le COTS est nécessaire. De plus le « wrapper » peut avoir besoin d'être présent dans une version précise et dans un environnement d'exécution spécifique. Il est même possible de définir une configuration qui ne sera pas acceptable pour l'intégration par l'intermédiaire d'un « Wrappers » particulier. C'est un scénario de non fonctionnement. Par exemple, le driver ODBC de MS Access ne fonctionne qu'avec les anciennes versions du produit, par contre il est nécessaire d'utiliser la version « Jet 4.0 » pour les versions d'Access supérieure à Access 2000.

Une autre méthode, le « Glue Code », correspond aux développements qui sont nécessaires pour utiliser les fonctionnalités des COTS. Dans ce cadre là, c'est surtout un environnement permettant le codage du « Glue Code » qui sera recherché, par exemple des langages de programmations et des outils de programmation. Ces derniers ont besoin d'un environnement d'exécution comme nous l'avons présenté dans le paragraphe précédent. Et comme dans le cas des « Wrappers », il est possible de définir des scénarios de non fonctionnement de l'environnement d'intégration.

Puis la dernière méthode, le « Tailoring », est présentée comme l'ajout de fonctionnalités à un composant existant et ne nécessitant pas la modification du code de ce composant. Comme précédemment, des outils et des langages de programmation sont

nécessaires et correspondent à cet environnement d'intégration. De même, les scénarii d'un environnement incompatible peuvent être utiles pour identifier des incompatibilités comme nous le décrivons dans le paragraphe suivant : « III - 3 Les scénarios d'incompatibilité ».

De ce fait, pour l'environnement d'intégration, nous retenons le besoin d'identifier les éléments permettant cette intégration comme :

- des logiciels, drivers et autres produits binaires,
- des technologies (ODBC, Corba, ou autres dans le cadre des « wrappers »),
- des langages de programmation (C++, Java dans le cadre du *Glue Code* ou du *Tailoring*),
- des outils de développement utilisant ces langages de programmation,
- des configurations de fonctionnement.
- de la documentation

Tout cela doit aider la mise en place du prototype en fournissant les informations nécessaires à la mise en œuvre de l'environnement d'intégration, c'est-à-dire des connaissances et des moyens nécessaires au développement du code permettant l'intégration.

Pourtant cela ne suffit pas, car pour que le développement puisse se faire ou que les outils nécessaires aux développements puissent fonctionner, il va falloir indiquer l'environnement ou les moyens permettant l'exécution de ces éléments.

III - 2.3 Un environnement d'exécution des outils pour l'intégration

A chaque fois que des outils sont utilisés, on a besoin d'un environnement d'exécution. Au même titre que pour « l'environnement d'exécution des COTS », on a besoin d'indiquer avec quel système d'exploitation les outils sont utilisables ou quels sont les autres produits indispensables pour un bon fonctionnement.

Prenons deux exemples : dans le premier cas, l'application MS Outlook nécessite un environnement d'exécution qui est celui d'un système d'exploitation Microsoft dans des versions dépendantes de la version du produit ; de plus certains services comme la prise de réunion en groupe ont besoin d'applications supplémentaires (MS Exchange dans cet exemple). C'est donc l'ensemble de ces informations qu'il faut détailler dans les données

produits pour identifier tous les éléments permettant l'exécution du COTS et de ces services.

Dans un deuxième cas, celui d'un développement avec un périphérique mobile, il faut avoir un environnement matériel pour la connexion avec le périphérique (le plus souvent par USB ou infrarouge) en plus du périphérique lui-même. Si on utilise un autre moyen, la connexion est impossible et le développement ne pourra pas se réaliser. Dans les « données produit » présentées dans la Figure 22, nous avons justement un exemple de configuration nécessitant un port spécifique de communication.

La mise en œuvre de ces outils peut nécessiter des informations ou un apprentissage. C'est pour cela que l'on estime nécessaire, dans un scénario d'intégration de COTS, l'identification de toutes les documentations ou informations pour sa mise en œuvre.

III - 2.4 Les connaissances techniques pour réaliser l'intégration

Les différentes briques logicielles ou matérielles qui sont nécessaires à la réalisation de l'intégration d'un COTS dans un prototype ne sont pas suffisantes si on ne maîtrise pas la manière de les intégrer.

Pour cela, toute la documentation et les expériences passées devraient permettre d'aider le développeur dans la réalisation de ce prototype. De plus, ce type d'information fait partie des données nécessaires aux phases de qualification et d'évaluation, notamment dans le cadre des estimations de coûts et de risques. Elles sont parties intégrantes du calcul de COCOTS et COCOMO [ESI01].

Parmi les connaissances qui sont recherchées et qui permettent de réduire les risques et de calibrer le coût de l'intégration, nous citons ici les plus communes :

- les technologies,
- les langages de développement,
- les exemples d'intégration.

Les technologies et les langages permettront notamment d'estimer l'effort nécessaire aux développeurs pour l'acquisition des connaissances indispensables pour le développement, car dans la majorité des cas, l'expérience d'autres développeurs dans ces domaines, permet de connaître les efforts consentis à l'intégration de COTS utilisant ces mêmes technologies et langages. A partir de ces expériences, il est possible d'évaluer un temps moyen passé par différents développeurs ce qui permettra d'utiliser ces informations pour émettre une estimation des efforts dans le cadre du développement en cours.

Les exemples d'intégration permettront de réduire les risques sur la faisabilité d'intégration dans la mesure où ces exemples sont compatibles avec les besoins d'intégration. En effet, plus il y aura d'expériences en accord avec notre besoin, plus la valeur moyenne des efforts consentis pour leur réalisation permettront de mieux cerner le coût estimé. Dans le cas contraire, et pour réduire l'incertitude sur les efforts nécessaires, il faudra identifier les parties sur lesquelles le développeur est en manque de références pour essayer de clarifier la faisabilité technique dans la phase d'adaptation.

Dans tous les cas, que ce soit pour la réalisation du prototype ou pour recueillir des informations utiles à d'autres phases, la manière, les moyens et les connaissances pour réaliser l'intégration sont indispensables. De plus, on peut les compléter avec des informations de dysfonctionnement de l'intégration ou d'incompatibilité avec d'autres produits ou COTS (exemple de deux produits utilisant une application commune mais dont les versions sont différentes au niveau des services rendus).

Les expériences malheureuses sont aussi une source intéressante pour la phase de « Tri et de sélection » pour déterminer le bon choix de COTS à intégrer comme nous allons le voir par la suite.

III - 3 Les scénarios d'incompatibilité

Le rôle des scénarii d'incompatibilité, est d'identifier les cas de figure où l'exécution, ou alors l'environnement d'intégration du COTS, pose un problème ou engendre un dysfonctionnement. L'identification se fera par la description de l'environnement conduisant à un défaut. Par exemple, dans le cas d'un produit qui existe en deux versions, l'une pour les OS Windows de la famille NT (NT4, 2000 et XP) et un autre pour la famille des OS Windows n'ayant pas la technologie NT (95, 98 et Millenium). Le premier produit aura comme environnement d'exécution l'un des OS NT et par contre affichera une incompatibilité avec les OS 95, 98 et Millenium.

Cela doit permettre de connaître à l'avance les incompatibilités ou les procédures d'intégration qui ne fonctionneront pas. Ainsi les développeurs pourront éviter, lors de la réalisation des prototypes, la réalisation de scénarios voués à l'échec. Mieux encore, lors des phases de tri et de sélection des COTS, on pourra écarter des scénarios ou des COTS pour incompatibilité avec les besoins exprimés dans la première phase du processus de développement. Dans l'exemple précédent, on ne peut pas assembler le produit dans une application dont le système d'exploitation cible est la famille d'application Windows depuis la version 95 à la dernière version en date XP.

De ce fait, les risques sur l'intégration seront en partie réduits. De plus, on évitera de réaliser des travaux dans les phases de qualification et d'adaptation si l'incompatibilité est détectée assez tôt. Au mieux, on devrait éviter d'identifier les futurs problèmes lors de l'intégration de COTS dans le prototype ou pendant la phase d'assemblage, la remise en cause tardive d'un COTS étant l'une des difficultés identifiées sur l'utilisation des COTS.

Pourtant ces cas de figures sont parfois difficiles à détecter, notamment dans le cadre de l'utilisation conjointe de plusieurs COTS dans l'intégration. Plus la dimension du problème augmente et plus les difficultés de traitement et de vérification sont complexes. Ces scénarii représentent un prétraitement de l'information utile dans le cadre de l'estimation des risques.

Nous pensons que nous pouvons ainsi améliorer le tri et la sélection des COTS. C'est une information qui doit être regroupée dans un rapport que nous présentons ci-après. Son rôle est de synthétiser les données permettant d'aider les développeurs à réaliser un prototype, à identifier des informations utiles pour les phases d'évaluations, de tri et d'adaptation.

III - 4 Un rapport de synthèse

III - 4.1 Introduction

Le rapport de synthèse doit permettre d'aider les phases d'évaluation, d'adaptation et de tri des COTS. En outre, il doit aider dans la décision de mettre en œuvre ou non une boucle supplémentaire du « winwin » spiral modèle.

Dans la réalisation du prototype et pour la phase d'adaptation, il doit fournir des informations utiles à son développement. Les scénarii d'intégration ou de dysfonctionnement décrits plus haut peuvent aider le développeur du prototype. Les informations sur les COTS et notamment les « données produit » avec les services, les manques par rapport aux objectifs de l'application ou les contraintes d'utilisation de systèmes cibles sont tout autant d'informations utiles.

Dans la phase d'évaluation, certaines informations techniques sont nécessaires aux calculs de coûts et d'estimations des risques comme par exemple les services manquants de COTS ou le travail nécessaire pour l'intégration. On y trouve des informations sur les environnements, les technologies, les langages avec des exemples d'assemblage et les outils nécessaires à une intégration. Là aussi, nous retrouvons ces informations dans les « données Produit » que nous venons de présenter.

Pour pouvoir trier les COTS et décider s'il faut refaire une nouvelle phase d'évaluation et d'adaptation, il est nécessaire de comparer les informations récupérées pendant ces phases pour chaque COTS.

III - 4.2 Un rapport sur les COTS

III - 4.2.1L'apport de chaque COTS

La réalisation d'un prototype nécessite de mettre en œuvre les services d'un COTS correspondant aux besoins de l'application à base de COTS. Il s'agit donc de les identifier et de connaître techniquement le ou les moyens de les intégrer. Ils peuvent nécessiter des environnements de développement et d'utilisation, des outils ou des connaissances spécifiques.

Chaque COTS apporte donc ses services et ses contraintes. Ils peuvent être mesurés ou évalués comme dans le cas des contrôles qualité de la phase d'évaluation. En fait, et comme nous l'avons présenté dans l'introduction, ce sont surtout les « données produit » qui sont sollicitées. De ce fait, ce sont elles qu'il faut synthétiser.

Les développeurs attendent essentiellement des informations sur :

- Ce que le COTS sait faire,
- Comment il s'intègre dans un prototype.

Ainsi nous assisterons le développeur dans sa tâche d'élaboration du prototype qui lui permettra d'obtenir des informations pour les phases du processus correspondant au niveau décisionnel de notre schéma (Figure 25). On peut les compléter avec des informations sur ce qu'il ne sait pas faire par rapport aux spécifications des besoins exprimés dans la première phase du processus.

III - 4.2.2Les manques

Chaque COTS apporte des services utiles pour l'application mais il est fréquent qu'il ne réalise pas toutes les fonctionnalités attendues. Il est donc indispensable de les identifier. Cela sera utile par exemple dans la phase d'évaluation pour les calculs de coûts notamment car ils nécessiteront d'être développés ou complétés par d'autres COTS. De ce fait, il est nécessaire pour chaque service d'estimer les coûts en fonction du travail à réaliser (développement complet ou adaptation d'un COTS).

La synthèse doit mentionner les manques par rapport aux besoins. Cela permet d'identifier le travail restant à faire pour les phases de tri et de sélection. C'est en créant un scénario d'intégration des COTS que l'on choisit les produits à évaluer et à intégrer. Si

plusieurs solutions sont à l'étude, l'information sur le travail restant et le coût de celui-ci peuvent être déterminants pour privilégier un scénario lors de la réalisation d'une itération au sens du « winwin spiral model ». Parmi des scénarios différents, nous pouvons avoir par exemple le choix entre un COTS qui réaliserait la majorité des services demandés et plusieurs COTS qui par la mise en commun de leurs services, réaliseraient aussi ces fonctionnalités.

C'est pour cela que nous introduisons dans notre rapport ces notions de manques, mais aussi des informations sur les contraintes des COTS.

III - 4.2.3 Les contraintes

Dans la même famille d'informations, nous avons l'ensemble des contraintes techniques qui définissent la manière d'utiliser et d'intégrer le COTS. Elles seront utiles pour les développeurs en charge du prototype puis dans une phase d'assemblage, aux développeurs d'application. Pour les premiers cela facilitera le travail s'ils connaissent le mode d'intégration ou les scénarios à ne pas utiliser car ne fonctionnant pas. Pour les seconds, la connaissance des dysfonctionnements peut s'avérer utile, notamment pour la correction de problèmes survenus pendant l'assemblage.

Les contraintes ce sont aussi des informations sur les critères qualité. Par exemple, la précision des résultats issus d'une fonctionnalité du COTS en est une. Cela peut devenir une contrainte, lors de l'intégration, s'il faut prendre en compte cette caractéristique. Pour cela, la famille d'information que nous avons appelé les « données qualité » doit faire partie des informations que l'on doit retrouver dans le rapport de chaque COTS.

Dans certains cas, des incompatibilités peuvent devenir la cause d'une non sélection de COTS. Par exemple si dans un prototype nous assemblons un « COTS A » et un « COTS B ». Le premier a besoin pour son exécution d'un produit « Y » qui utilise un produit « Z » mais incompatible avec l'utilisation du produit « X ». Le second a besoin d'un produit « V » utilisant le produit « X ». Dans cet exemple, nous pouvons conclure que l'utilisation simultanée du COTS A et B n'est pas possible : c'est donc une incompatibilité que le développeur devra trancher, soit par l'utilisation de l'un ou l'autre des COTS, soit par le développement des services non remplis par le COTS manquant ou alors par le remplacement de l'un d'entre eux.

Le fait de détecter au plus vite ce type d'information évitera la réalisation d'un prototype voué à l'échec par incompatibilité lors de l'intégration ou ne respectant pas des contraintes d'outils ou d'applicatifs qui sont impérativement utilisés, alors conjointement avec l'application développée. Le rapport de synthèse peut informer sur ce type de problème majeur qui remettra en cause le COTS ou les conditions à l'origine des incompatibilités.

III - 4.3 L'estimation des coûts et des risques

D'après l'étude sur l'utilisation de COCOMO II par Madachy [Madachy97] sur l'estimation des coûts et des risques, Madachy considère ce modèle (développé par Boehm) comme accepté par tous. C'est effectivement pour cet auteur un modèle multi-niveaux de référence pour estimer l'effort d'intégration des COTS qui est utilisé dans plusieurs outils d'évaluation dans l'industrie et la recherche. La formule de COCOMO II proposée par Boehm [Boehm89] est la suivante :

$$PM = a * \left\{ [(size) * (1 + Brak / 100)] \left[1.01^{+0.01 \sum_{j=1}^{n \text{ niveaux de complexité}} SF_j} \right] \right\} * \prod_{i=1}^{\text{nombre de cot s}} EM_i + (auto \ adaptation \ effort)$$

Dans laquelle :

- « PM » correspond aux efforts mesurés en homme/mois,
- « a » est une constante linéaire provenant des données du projet (par défaut elle est égale à 2.5),
- « size » est une expression qui tient compte du nombre de lignes du code source, fonctions et objets du COTS,
- « Brak » est le pourcentage du code modifié pour tenir compte des besoins d'intégration,
- « SF » est une échelle de niveau de complexité des produits à intégrer sur 5 niveaux [Abts97] (très bas, bas, moyen, haut et très haut). Dans son exemple, Abts donne la valeur 4 au niveau très bas, 1 au niveau haut et 0 au niveau très haut,
- « EM » est une constante d'effort permettant de prendre en compte le nombre de drivers de COTS. Il existe 14 constantes d'effort [Abts97, Abts00] paramétrables sur 5 niveaux (très bas, bas, moyen, haut et très haut).

A partir du calcul des efforts en homme / mois Boehm introduit la formule permettant d'estimer le coût de l'intégration :

$$COST = PM * (\$/PM)$$

Dans laquelle « \$\$/PM » correspond au salaire moyen par personne et par mois.

Lors de l'utilisation de sa méthode, Boehm constate, entre autre, que la valeur de « Brak » ne peut être obtenue avec précision qu'avec le concours du fournisseur de COTS et des utilisateurs de ce COTS dans d'autres projets.

C'est donc l'expérience accumulée dans divers développement avec ce COTS qui permettra d'affiner l'évaluation des efforts. Cela est confirmé dans l'article de Boehm [Boehm97]. Son étude démontre que pour obtenir avec un bon niveau d'approximation avec treize paramètres multiplicateurs d'efforts (correspondant à son étude), il est nécessaire de confronter les avis de plusieurs participants.

Dans tous les cas, l'estimation des coûts et des efforts d'intégration ne peut être obtenue sans trop d'approximations que si les acteurs déterminant les paramètres multiplicateurs d'efforts de la formule, ont les connaissances suffisantes pour fournir les bonnes valeurs. Dans ce cas, les expériences d'intégration, l'aide des fournisseurs et de la communauté de COTS sont autant de moyens d'obtenir suffisamment d'informations pour déterminer ces paramètres correctement.

Le manque d'information est un risque d'erreur qui peut être pris en compte pour les facteurs de risque. Il existe d'après Madachy [Madachy97], plusieurs catégories de risques qui sont tous reliées aux différentes catégories de coûts de COCOMO. Il utilise pour déterminer le facteur de risque du projet la formule suivante :

$$Pr\ ojectRisk = \sum_{j=1}^{No.categories} \sum_{i=1}^{No.categoryrisks} risklevel_{ij} \times effortmultiplierproduct_{ij}$$

Dans cette formule la valeur de « riskLevel » correspond à une probabilité non linéaire que le risque puisse exister et « effortmultiplierproduct » correspond à une constante multiplicative sur l'effort basé sur 5 niveaux de valeurs. Madachy propose des valeurs comprises entre 0.75 et 1.66 avec une valeur nominale à 1 pour cette dernière.

Il identifie des risques sur la planification, le produit, la plate-forme, le personnel et le processus. Dans toutes ces catégories, on a un dénominateur commun, particulièrement intéressant pour notre démarche, qui décrit les raisons pour lesquelles les coûts d'intégration doivent être revus à la hausse. Ce dénominateur est identifié, dans la majorité des cas, comme liés à un manque d'informations ou à des informations incomplètes. D'où, l'importance de réaliser des prototypes permettant de recueillir le plus d'informations possibles dans chaque catégorie de risques.

En ce qui concerne, l'estimation du coût et des efforts d'intégration ainsi que le facteur de risque, ils ne peuvent être calculés directement à partir d'informations mesurées lors de la réalisation d'un prototype comme le temps de réponse d'un service ou la précision d'un résultat. Pourtant, les informations recueillies, pendant cette phase, contribueront dans la

détermination des valeurs à affecter aux facteurs multiplicateurs nécessaires aux formules d'estimations citées ci-dessus. D'où l'importance, pour le rapport de synthèse, de montrer dans chaque catégorie de risque les informations recueillies dans l'utilisation des prototypes mais aussi auprès des fournisseurs et des utilisateurs de COTS. Ces estimations de risques et de coûts sont des éléments servant à comparer les COTS.

III - 4.4 Les comparatifs

Jusqu'à présent, le rapport de synthèse propose des informations pour l'ensemble des produits utiles pour la réalisation de l'application à base de COTS. Nous avons présenté les informations pour un scénario d'intégration. Pourtant, il est naturel, dans un projet d'intégration de COTS, d'avoir à choisir soit entre plusieurs produits qui proposent des services similaires, soit entre plusieurs scénarii d'intégration. Dans les deux cas, le comparatif doit aider les développeurs dans le choix du scénario à mettre en œuvre lors de la phase d'assemblage. Au même titre que l'estimation des risques et des coûts, il n'est pas envisageable que le rapport puisse fournir de lui-même le classement des scénarii et des produits. C'est donc dans la manière de présenter les informations que l'on peut aider les développeurs : filtre, tri, regroupement, etc... D'autant plus que les seules informations issues du prototype ne sont pas suffisantes pour déterminer un choix comme nous l'avons vu dans le chapitre Chapitre 2 : IV - 3 : La phase d'acquisition et de sélection. Nous proposons de présenter les informations par catégories et familles en fonction des facteurs de risques ou de leur utilisation (outil pour l'intégration, environnement d'exécution, de développement, ...).

III - 4.5 Bilan sur le rapport de synthèse

Le rapport de synthèse propose le recueil des informations pour chaque scénario d'intégration. Le scénario étant un assemblage de produits rentrant dans la composition de l'application à base de COTS. Pour chaque scénario, il est important de présenter les informations de chaque catégorie de données (Produit, Spécification, Qualité, d'estimation). Cela permettra d'avoir une connaissance sur les produits pris séparément à propos des environnements d'exécution, d'intégration et de dysfonctionnement. Mais aussi nous pouvons considérer les interactions, dans ces mêmes domaines, entre produits lorsqu'ils cohabiteront dans l'application à base de COTS. La possibilité de classer, filtrer et trier les informations est un plus qui permettra une meilleure lecture des multiples données recueillies sur chaque produit.

Dans tous les cas, le rapport doit aider les développeurs dans la réalisation des prototypes (choix de l'environnement de développement du prototype et d'exécution des COTS à adapter dans le prototype), la détection des incompatibilités, les services rendus et utiles pour l'application des COTS ainsi que les services restant à développer et enfin pour

la détermination des estimations des coûts et des risques. Pour réaliser un prototype, il est indispensable de connaître les différents environnements nécessaires à l'exécution et à l'intégration. La détection des incompatibilités sera fournie par la description des contraintes et dysfonctionnements entre produits. L'information doit être associée à chaque fiche produit. De plus, le travail d'intégration des services à réaliser et ceux fournis par les COTS permettra de faire l'estimation des coûts et des risques.

Dans le cadre de l'aide à la réalisation d'un prototype, ce rapport, donne surtout les informations sur les moyens à mettre en œuvre pour son développement. Et le plus important, c'est qu'il va détecter facilement les incompatibilités et les dysfonctionnements. Sachant qu'aujourd'hui cette détection se fait pendant la réalisation du prototype, ou pire encore, pendant la phase d'assemblage (remise en cause de tout le processus de validation des COTS), nous permettrons un gain de temps et une diminution des facteurs de risque correspondant au coût de l'intégration mais aussi de planification. C'est donc tout d'abord un gain de qualité, mais aussi, par le fait de ne pas réaliser un prototype qui déterminerait ce dysfonctionnement, un gain de temps qui peut être mis au service du recueil d'informations. Cela devrait donc aller vers une amélioration de la qualité et une réduction supplémentaire des risques.

IV Conclusion

L'utilisation des COTS, notamment dans le cadre de la réalisation d'un prototype permettant de recueillir des informations utiles pour les phases d'évaluation et de sélection des COTS, nécessite la manipulation d'un nombre important de données. Nous les avons classées en fonction de deux activités : les données dont les développeurs ont besoin pour le déroulement du processus de validation et d'acquisition des COTS, et les données nécessaires à la réalisation d'un prototype permettant cette acquisition.

Dans le premier cas, les données ont été classées en quatre familles correspondant chacune à une utilisation. On trouve la connaissance des produits avec les données « Produit », le traitement de la qualité avec les données « qualité », la saisie des spécifications de l'application à base de COTS avec les données de « spécification » et l'estimation des coûts et des risques avec les données « d'estimation ». Ce sont ces données que les développeurs essaient de compléter avec le plus de précision et d'acuité possibles.

Pour ce faire, la deuxième activité (la réalisation du prototype) permettra de compléter et de vérifier l'exactitude des informations recueillies et d'enrichir les données

précédentes. Mais pour développer un prototype, il est là aussi nécessaire d'avoir des informations spécifiques. Nous les avons regroupées en deux catégories : le scénario d'intégration permettant aux développeurs de connaître les moyens de mise en œuvre d'un prototype et de son exécution, puis un ensemble d'informations sur des difficultés connues pour l'utilisation ou l'intégration des produits (les scénarii de dysfonctionnement).

Dans la phase d'estimation des risques et des coûts, nous avons noté la difficulté de fournir des éléments chiffrés à partir des données recueillies précédemment. C'est l'expérience et la maîtrise globale du projet d'intégration et des informations sur les COTS qui permettent aux personnes en charge de faire l'estimation des risques et des coûts, de déterminer aux mieux les différents facteurs multiplicateurs de risques, utilisés dans la formule mis au point par Boehm. C'est pour cela que nous proposons de pouvoir accéder aux données avec la possibilité de les classer, de les trier, de les filtrer et de les comparer entre différents produits ou scénarios d'intégration.

Les différents acteurs de ce processus de développement doivent manipuler un grand nombre d'informations. Ils ont de plus besoin d'aide de la part des différents acteurs du processus de développement, comme par exemple des informations sur les produits nécessaires pour faire les estimations, mais aussi de la part d'autres acteurs externes à ce processus comme les fournisseurs ou la communauté utilisatrices de COTS. Le simple fait d'avoir identifiées les informations manipulées dans chaque phase du processus de développement n'est pas suffisant pour aider les développeurs dans la réalisation de leurs prototypes. Il manque notamment la relation avec les autres utilisateurs de COTS pour capitaliser leurs propres expériences. Plus il y aura de données sur l'intégration et l'utilisation des COTS et plus les sélections et les évaluations des COTS seront efficaces. C'est donc l'enrichissement des informations sur chaque COTS qui aura un rôle moteur dans l'aide apporté aux développeurs car ces autres expériences faciliteront la réalisation des prototypes qui nous intéressent plus particulièrement dans nos travaux. De ce fait, il nous semble indispensable de traiter ces informations à travers un Système d'Information (SI) pour COTS qui permettra de capitaliser de l'information pour la réalisation des prototypes et de mettre en relation les différents acteurs de l'intégration des COTS comme nous le verrons par la suite et notamment avec le démonstrateur que nous avons réalisé.

Chapitre 4 : Le Système d'Information pour COTS

I Introduction

Au travers de l'état de l'art, nous avons abordé les difficultés de réalisation des applications à base de COTS. Notamment, avec l'utilisation des phases spécifiques à ce type de développement et avec des outils pour l'estimation des risques et des coûts, nous essayons aujourd'hui de diminuer les difficultés. Quelque soit la phase du processus de développement, la difficulté majeure que nous avons mise en évidence concerne le recueil et le traitement des informations, en particulier pour la réalisation des prototypes. C'est pourtant à l'aide de ces derniers que les développeurs pourront recueillir les informations manquantes et indispensables pour ce processus.

Pour la manipulation et le traitement des données, la solution idéale consiste dans l'utilisation d'un Système d'Information. Il permet de recueillir, traiter ou transformer puis diffuser de l'information. De plus, il offre la possibilité de mettre en relation plusieurs acteurs et permet ainsi de réaliser des échanges entre eux. Ces derniers sont nécessaires dans le processus de développement à base de COTS. C'est le cas dans les phases où la recherche d'informations est indispensable et surtout pendant la réalisation d'un prototype.

C'est pour cette raison que nous proposons un Système d'Information (SI) spécifique pour le développement d'applications à base de COTS. Il est basé sur l'utilisation d'une

carte d'identité de COTS qui référencera l'ensemble des informations manipulées. Il permettra de réduire la principale des difficultés évoquées ci-dessus : l'obtention des informations. C'est à travers les acteurs, utilisateurs de ce SI, que nous pourrions établir une relation client / fournisseur permettant de recueillir et de capitaliser les informations utiles au processus de développement. Finalement, nous apporterons une aide à la réalisation des prototypes, indispensable à la conduite du processus et à la bonne sélection des COTS. Tout cela devrait nous permettre d'évaluer les gains attendus de l'utilisation du SI pour COTS.

II Le rôle du Système d'Information

II - 1 *Définition du Système d'Information*

Dans l'ATLAS de 2004 [Mével03] le Système d'Information (SI) est défini comme un ensemble organisé d'informations relatives à une entreprise ou à un domaine de connaissance. Cet ensemble d'informations circule entre les acteurs en utilisant des processus : émission, transmission, réception et traitement. Elles sont saisies puis mises en forme pour la transmission vers le récepteur de l'information.

D'après Boulding et Lemoigne [Lemoigne77], un système peut être représenté par ses trois constituants que sont : le système opérant (SO), le Système d'Information (SI) et le système de pilotage (SP). Ainsi, il fait la liaison, entre le système de décision et le système opérant, avec le Système d'Information comme le montre la Figure 29.

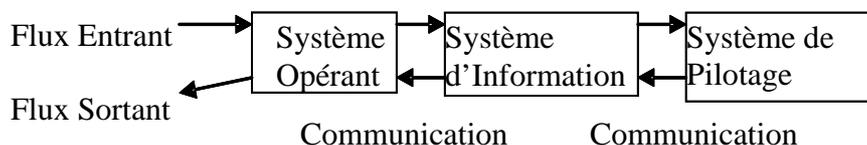


Figure 29 : Décomposition d'un Système d'Information

Le SO permet la transformation de flux entrant en flux sortant, c'est le cas des différentes phases du processus de développement. Elles ont toutes un flux d'informations entrant qui sera transformé pour permettre d'initialiser la phase suivante du processus de développement.

Le SP définit l'activité de décision que nous retrouvons dans notre modélisation du processus de la Figure 10 (Chapitre 2 : IV - 7, page 57)

Nous retrouvons donc cette activité de décision dans :

- la phase de sélection qui permet d'initialiser les évaluations à l'aide de la réalisation d'un prototype,
- la phase de tri et de sélection en utilisant les résultats des phases d'évaluation ou d'adaptation,
- les itérations après la phase d'adaptation ou après la phase d'assemblage et,
- la phase de mise à jour pour la reprise du processus de développement.

Le SI permet aux deux systèmes (SO et SP) de communiquer entre eux en fournissant une image du processus et contenant l'ensemble des informations utiles pour ces systèmes.

C'est à partir des informations fournies d'une part par les concepteurs et la communauté de COTS et d'autre part par les spécifications de l'application à base de COTS que le Système d'Information va pouvoir démarrer son processus. Il doit conduire le développeur à l'évaluation des différents produits à intégrer : cela représente le premier flux entrant.

On trouve d'autres flux entrants avec les informations recueillies pendant toutes les phases du processus de développement qui permettront d'enchaîner les différentes activités à partir des décisions prises suite aux différents résultats obtenus dans chaque phase.

II - 2 Relier les fournisseurs d'informations et les clients

Les concepteurs de COTS sont les premiers acteurs pouvant fournir des informations sur leurs produits. Nous avons défini dans notre modélisation (Figure 12, page 69) que la communauté des COTS fait partie des acteurs ayant aussi la capacité de fournir de l'information. Si les premiers se basent sur le développement du COTS, les seconds vont utiliser leurs expériences de leurs intégrations passées, pour diffuser de l'information utile aux développeurs d'application à base de COTS (acteurs que nous avons nommés utilisateurs).

Le besoin d'information et la difficulté de les obtenir montrent qu'il faut établir dans le processus de développement des liens entre les développeurs (les utilisateurs) en manque

d'informations et les différents acteurs pouvant en offrir (concepteur et communauté de COTS).

Relier les différents acteurs doit permettre aux uns, de demander de l'aide (à l'image des forums de discussion), et aux autres d'en fournir (en enrichissant le SI). Ce sera le rôle du SI que de mettre en relation les différents acteurs, de stocker et de trier les informations qui sont manipulées. C'est sa capacité à présenter les données aux différents acteurs qu'il semble intéressant d'exploiter ici, comme nous le verrons dans le paragraphe suivant.

II - 3 Favoriser le recueil et le traitement des informations

En effet, ces informations pourraient être triées en fonction de leur nature ou de leur utilisation. Par exemple, notre proposition de famille d'informations pourrait servir de classification en relation avec les différentes phases du processus de développement.

En fonction de la phase et des besoins en informations il est possible d'adapter la présentation et le recueil des données. De plus, l'utilisation d'un système de traitement doit pouvoir interagir entre les différentes phases du développement. Les informations d'une phase pouvant servir pour d'autres phases. Par exemple, la définition des besoins en termes de services, de qualité logicielle et de critères de sélection, peuvent interagir avec les comparaisons faites pendant les activités de tri et de sélection des COTS. Lorsque, par exemple, le développeur ajoute un service supplémentaire à son application, le SI doit demander au développeur qu'il sélectionne un service d'un ou plusieurs COTS permettant de le réaliser. Le résultat dans d'autres phases du processus peut être l'identification d'un travail supplémentaire en développement, en évaluation ou en comparaison de COTS entre eux. Aussi, les évaluations et les phases d'adaptation peuvent interagir avec les calculs de coûts et l'estimation de risques. Tout cela peut être utile, lors des phases de décision dans le processus de développement, si chaque phase a une information sur les résultats ou les modifications faites dans les autres phases du processus. De fait, cela permettra de réduire les difficultés d'utilisation des COTS.

III Les acteurs et les informations disponibles pour le Système d'Information (SI)

III - 1 Les acteurs du Système d'Information

III - 1.1 Le fournisseur et le concepteur de COTS (FourCo)

Il est en fait le vendeur ou le distributeur du produit. Mais il est surtout dans notre approche le concepteur du COTS.

Dans un circuit de distribution classique on doit dissocier le fabricant d'un produit et le circuit de distribution ou vendeur de ce produit. C'est effectivement, dans bien des cas, deux acteurs différents. Pourtant, dans la relation que l'on essaye de montrer, cette différence n'apporte rien sur le fait que ce soit l'un ou l'autre qui fournira le produit et les informations dont ont besoin les clients. Seule la différence de services proposés entre divers circuits de vente pourrait nécessiter que l'on définisse deux acteurs séparés. Or ce critère là, s'il est important lors de la sélection du vendeur de COTS, il n'a pas d'influence sur la qualité et le choix du produit. Ce dernier est identique quelque soit le vendeur. Notre préoccupation est focalisée sur les informations dont ont besoin les clients pour mener à bien leurs missions de sélection et d'intégration des COTS. En particulier nous voulons surtout les aider à réaliser un prototype, et pas spécialement dans la sélection d'un vendeur.

Pour cela, nous ne considérerons ici qu'un seul acteur : « FourCo ». Il sera le responsable des informations que le client pourrait lui demander au titre de concepteur et diffuseur de COTS. Mais dans un premier temps, pour pouvoir initier une relation client / fournisseur, il est indispensable que FourCo diffuse de l'information sur le produit qu'il met à disposition de ses futurs clients.

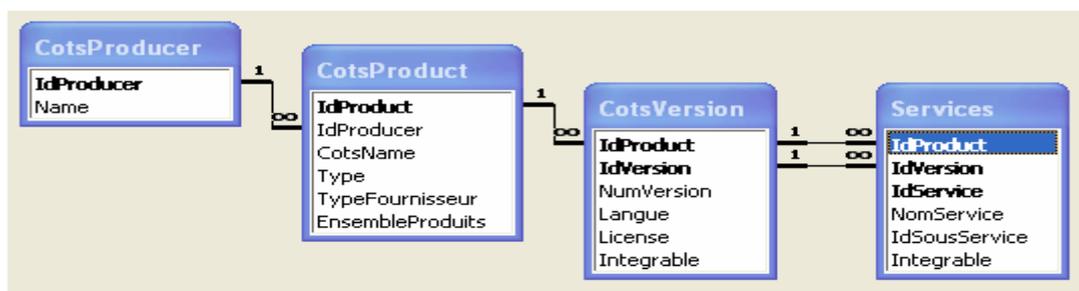


Figure 30 : Les services de COTS

Cette figure correspond à la base de données du SI qui permet de saisir le minimum d'informations que l'on trouve dans sur les sites Web des distributeurs de COTS, à savoir les services proposés par chaque produit.

Les produits sont identifiés dans le SI de manière unique à l'aide d'une clé d'identification que nous avons nommé ici « IdProduct » dans la Table « CotsProduct ». Chaque produit appartient à un fournisseur de COTS que nous identifions dans la table « CotsProducer ». Un fournisseur de COTS pouvant avoir plusieurs COTS, nous retrouvons la relation 1..n entre ces deux tables.

Chaque produit se décline en plusieurs versions identifiées à travers la table « CotsVersion » qui permet de lister différentes informations et en particulier les services proposés par les produits que nous décrivons à l'aide de la table « Services ».

III - 1.2 L'utilisateur de COTS

Un COTS est un logiciel acquis par un « utilisateur » de COTS. C'est pour nous un développeur qui réalise son application. C'est aussi le client vis-à-vis du fournisseur du produit. Le client va utiliser ce COTS en l'intégrant dans une application qu'il va développer à l'aide de sa propre équipe de développement. Pour cela il utilisera un processus comme celui proposé dans la Figure 10 : processus de développement à base de COTS.

Dans celui-ci, il existe un certain nombre d'activités à réaliser et surtout l'utilisateur de COTS devra récupérer de nombreuses informations. Elles sont en partie disponibles chez les fournisseurs de COTS, et pour la partie restante, les développeurs vont devoir les rechercher à travers des évaluations et des tests réalisés avec l'aide des prototypes.

C'est donc « l'utilisateur de COTS » qui, en lançant la première activité du processus de développement (la veille technologique), va initialiser un processus de communication avec FourCo qu'il pourra continuer à utiliser tout au long du processus de développement de son application à base de COTS. Dans le processus de développement à base de COTS (Figure 10, page 57) nous avons introduit un nouvel acteur (la communauté de COTS) qui est en mesure d'aider l'utilisateur de COTS.

III - 1.3 La communauté de COTS

Elle représente l'ensemble des développeurs avec au moins une expérience de développement d'application à base de COTS. Ils ont acquis de l'information pendant leur développement et notamment durant les phases de qualification, d'adaptation et de réalisation de prototypes, mais aussi pendant les évaluations des risques et des coûts. Plus encore, l'historique de leurs développements permettra de comparer les estimations avec la

réalité du coût de l'intégration du COTS. Cela devrait affiner la propre estimation des coûts et des risques du projet mené par l'utilisateur de COTS.

Nous pensons aussi aux informations issues des scénarios d'intégration réussis ainsi que de tous ceux identifiés comme ne permettant pas l'intégration ou ne satisfaisant pas les critères de sélection. En fait, la communauté de COTS possède des informations issues de projets d'intégration passés. Ces dernières peuvent être utiles à d'autres développeurs (utilisateurs de COTS) qui sont demandeurs d'informations.

La communauté de COTS devient au même titre que les concepteurs les fournisseurs d'informations. Le fait de pouvoir compléter le SI, en particulier les cartes d'identité de COTS, facilitera le recueil des informations et leur sélection. Ces acteurs ont un rôle important par le fait qu'ils vont enrichir les fiches descriptives des COTS. Les informations ainsi ajoutées pourront servir à d'autres utilisateurs de COTS qui posséderont une information plus complète sur comment intégrer le COTS.

III - 1.4 La liste des acteurs

Comme nous venons de le voir, nous vous avons proposé trois acteurs : le fournisseur, le client et la communauté des COTS. Nous les avons modélisés en UML (Figure 31), ils nous serviront par la suite pour décrire notre approche.

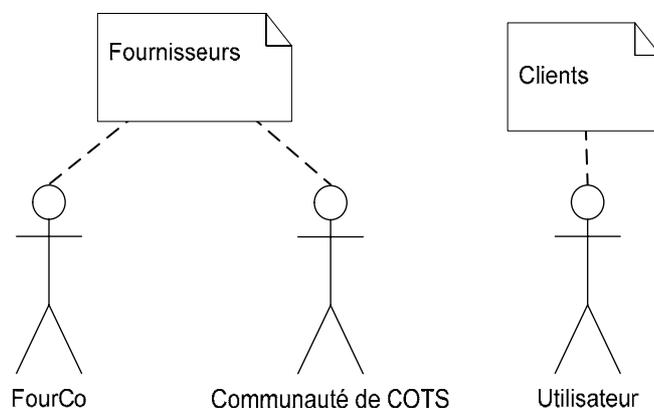


Figure 31 : les acteurs du SI

L'utilisateur de COTS est l'initiateur du processus de développement à base de COTS. A ce titre, les phases de « veille technologique » et de « définition des besoins » nécessiteront des informations contenues dans le SI et qui sont modélisées dans la Figure 32 comme un acteur. Ces dernières ont été préalablement rentrées par le concepteur et fournisseur de COTS. Elles permettent au client de connaître les COTS avec lesquels il va initier son processus de développement comme proposé dans la séquence UML de la Figure 32.

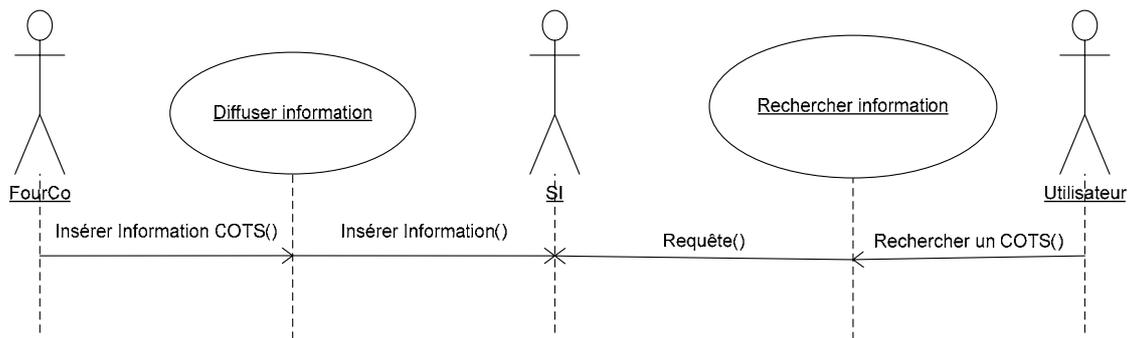


Figure 32: Première séquence pour l'initialisation du fonctionnement du SI

D'après le diagramme de séquence, « FourCo » doit commencer par insérer de l'information à destination des « utilisateurs ». Dans le paragraphe suivant nous allons établir la liste des informations les plus couramment fournies par cet acteur.

III - 2 Les informations disponibles

Dans la première phase du processus de développement à base de COTS (la veille technologique), l'utilisateur de COTS doit recenser les produits pouvant convenir à son domaine d'activité et à de futurs besoins de développement. Dans ce cadre, les recherches se font essentiellement sur Internet. Les sites de distributions ou les fabricants de COTS sont les principales sources d'informations.

Pourtant, lorsqu'on regarde les sites Internet ou les documentations des vendeurs sur les services rendus par les COTS, elles sont très différentes d'un vendeur à l'autre. On relève pourtant des similitudes :

- Le nom du produit,
- Le nom du concepteur,
- Le type de licence ou les prix,
- La catégorie ou la famille à laquelle appartient le produit, souvent définie par le fournisseur de l'information et donc différente d'un site à un autre,
- Une description des services plus ou moins complète (bien souvent succincte),

- Un descriptif des pré-requis logiciels et matériels nécessaires au fonctionnement envisagé et bien souvent sans aucun scénario d'incompatibilité.

Sur le site d'un concepteur de COTS (« SUN²») nous retrouvons ces quelques informations citées ci-dessus comme le montrent ces quelques captures d'écrans.

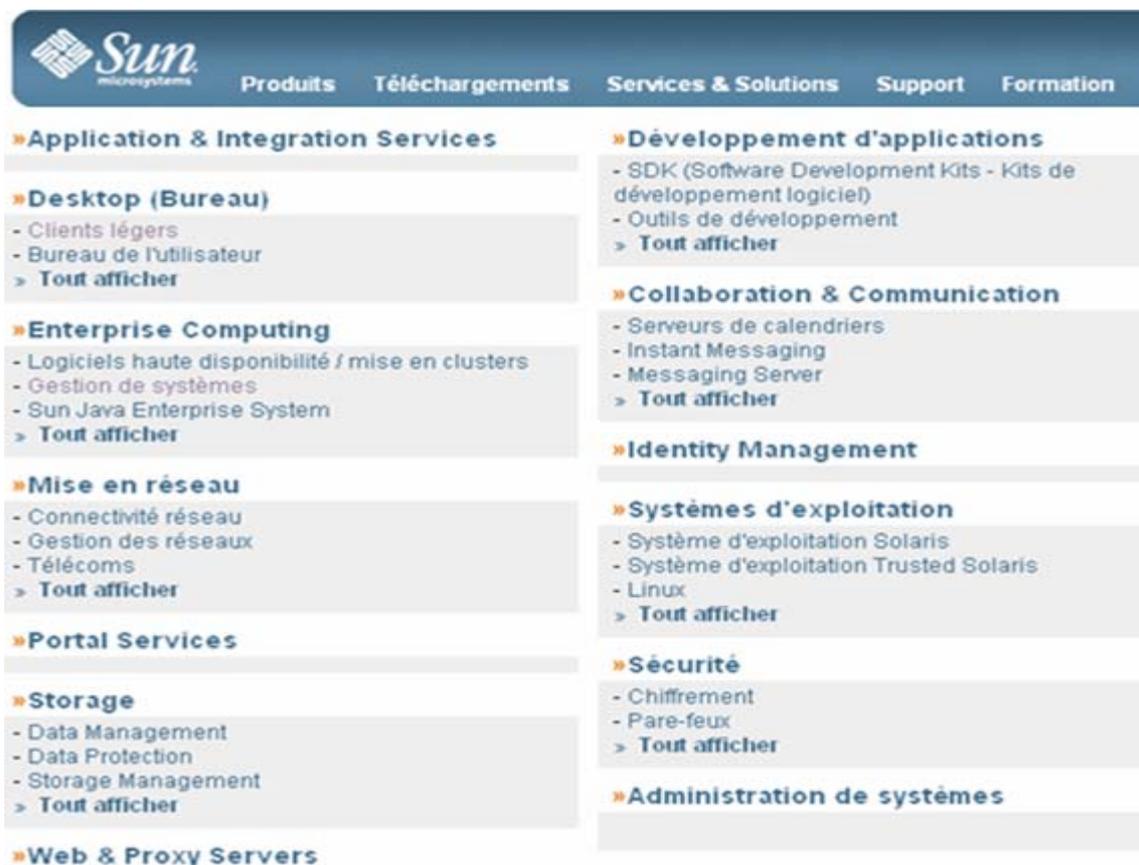


Figure 33 : Logiciels par catégories chez SUN

La décomposition en catégorie de SUN présente en Figure 33 est la propriété de l'éditeur. Chaque fournisseur de contenu choisira le nom des catégories et l'appartenance d'un logiciel à l'une de ces familles. D'où des difficultés pour « l'utilisateur » dans sa recherche et la comparaison des produits et des fournisseurs. Il est donc nécessaire d'établir une norme sur les noms de ces catégories ou familles.

² SUN Microsystems

Lorsqu'on choisit un logiciel dans une catégorie donnée, on trouve généralement un descriptif des fonctionnalités ou services du produit comme dans le cas de la « gestion de systèmes » (Figure 34).



Produit	Description	Prix
» Solaris Resource Manager	<p>Le logiciel Solaris Resource Manager permet de gérer les ressources informatiques d'un système en garantissant une disponibilité et une utilisation optimales. Les administrateurs système souhaitant consolider des utilisateurs ou des applications sur un serveur unique peuvent utiliser ce logiciel pour garantir la bonne allocation des ressources en fonction des priorités qu'ils se sont fixées. Au final, l'administrateur d'un système simple peut gérer un plus grand nombre d'applications ou d'utilisateurs sur un nombre inférieur de serveurs et d'instances de systèmes d'exploitation, optimisant ainsi l'utilisation des ressources matérielles. De plus, Solaris Resource Manager offre la flexibilité nécessaire pour la modification d'allocations de ressources « à la volée ». Ainsi, si une application est plus sollicitée, les administrateurs peuvent, grâce à des politiques définies, permettre à ladite application de disposer des ressources nécessaires pour répondre au pic de demande.</p>	Details »
» Sun Control Station	<p>Sun Control Station est une solution logicielle conçue pour simplifier à l'administrateur système la gestion du serveur de volume et les déploiements du système Sun Java Desktop. Sun Control Station est idéale pour les environnements tels que la fourniture de services, l'informatique technique et les environnements de bureau, dans lesquels il est nécessaire de dimensionner, gérer à distance et maintenir à jour les serveurs de volume avec les dernières applications et mises à jour logicielles.</p>	Details »
» Sun Management Center	<p>Sun Management Center est un système de gestion des éléments destiné à surveiller et gérer l'environnement Sun. Sun Management Center peut être intégré aux principaux systèmes de gestion d'entreprise, afin d'offrir aux clients une infrastructure de gestion unifiée. Le pack de base de Sun Management Center est gratuit et comprend la surveillance du matériel. Les applications avancées (add-ons) étendent les capacités de surveillance du pack de base.</p>	Details »

Figure 34 : Liste de produits appartenant à une catégorie chez SUN

Le descriptif du produit est sommaire. Nous avons le nom du produit sans en connaître sa version et un descriptif succinct de son utilisation. Le détail de ce descriptif fournira les grandes lignes des fonctionnalités du logiciel et une ou plusieurs configurations pour son fonctionnement comme nous pouvons le voir dans les deux captures d'écran des Figure 35 et Figure 36.

Solaris Resource Manager



Le logiciel Solaris Resource Manager permet de gérer les ressources informatiques d'un système en garantissant une disponibilité et une utilisation optimales. Les administrateurs système souhaitant consolider des utilisateurs ou des applications sur un serveur unique peuvent utiliser ce logiciel pour garantir la bonne allocation des ressources en fonction des priorités qu'ils se sont fixées. Au final, l'administrateur d'un système simple peut gérer un plus grand nombre d'applications ou d'utilisateurs sur un nombre inférieur de serveurs et d'instances de systèmes d'exploitation, optimisant ainsi l'utilisation des ressources matérielles. De plus, Solaris Resource Manager offre la flexibilité nécessaire pour la modification d'allocations de ressources « à la volée ». Ainsi, si une application est plus sollicitée, les administrateurs peuvent, grâce à des politiques définies, permettre à ladite application de disposer des ressources nécessaires pour répondre au pic de demande.

Details »

- Permet de contrôler les principales ressources système, telles que le processeur, la mémoire physique, la mémoire virtuelle, le nombre de processus, le nombre de connexions, ainsi que leurs durées. Les responsables de départements informatiques peuvent offrir à leurs utilisateurs un service de meilleure qualité et plus stable.
- Allocation de ressources partagées : le client peut allouer les ressources du processeur en fonction de politiques prédéfinies basées sur un système de parts, par opposition aux pourcentages fixes. Ainsi, le système partage de façon dynamique toutes les ressources disponibles suivant la proportion relative des parts allouées aux utilisateurs actuels. Cette procédure permet de simplifier l'ajout de nouveaux utilisateurs ou de nouvelles applications
- Permet de réaliser de multiples allocations par domaine, d'optimiser l'utilisation des ressources et d'augmenter l'efficacité des périphériques matériels. Un nombre réduit de serveurs et d'instances de système d'exploitation permet de réduire les coûts de gestion.
- Administration des ressources basées sur des politiques ; permet de configurer et de gérer des limites de ressources sans l'intervention de l'administrateur. Les ressources système non utilisées sont automatiquement réallouées afin d'optimiser les

Figure 35 : Description d'un produit chez SUN

- **Configuration système requise:**
 - Espace disque minimum : 6 Mo
 - Mémoire minimum : 3 Ko par utilisateur
- **Plates-formes prises en charge:** Solaris Resource Manager est pris en charge sur tous les systèmes SPARC qui s'exécutent sous les environnements d'exploitation Solaris 2.6, Solaris 7 ou Solaris 8

REMARQUE : la fonction de contrôle de la mémoire physique est uniquement disponible pour Solaris 8 version 1 (6/00) et les versions ultérieures

- **Systèmes d'exploitation pris en charge:**
 - Environnement d'exploitation Solaris 2.6 (SPARC)
 - Environnement d'exploitation Solaris 7 (SPARC)
 - Environnement d'exploitation Solaris 8 (SPARC)

Figure 36 : Configuration requise pour l'exécution d'un produit chez SUN

Dans tous les cas, le détail de tous les services ou des fonctionnalités proposés par le produit, n'est pas exposé dans ces sites. De plus, ils n'indiquent pas si le produit est intégrable ou non. Pourtant, s'il existe un filtre essentiel dans la recherche de COTS, c'est bien celui des possibilités d'intégration que peut avoir un produit (en référence à notre définition des COTS dans le paragraphe Chapitre 2 : II - 1). En effet, tout produit ne permettant pas d'être assemblé, n'intéresse pas notre acteur dans cette phase de veille technologique destinée à établir une liste de produits qu'il pourrait utiliser dans un futur développement. Dans le SI, nous avons introduit un attribut permettant d'identifier cette caractéristique au niveau des tables « CotsVersion » et « Services » (Figure 37). Ainsi, nous pouvons identifier les COTS qui intéressent les développeurs et les services des COTS qui pourront être intégrés et ceux qui ne le sont pas.

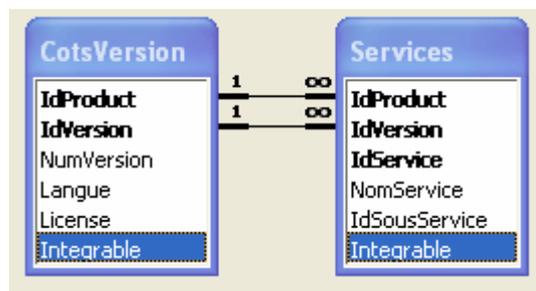


Figure 37 : Attribut pour identifier les COTS et les services intégrables

Sans notre SI, pour obtenir des informations plus techniques, notamment sur les capacités d'intégration, et sur une liste plus détaillée des services mis à disposition par les produits, il va falloir :

- contacter directement le concepteur avec lequel il y aura des possibilités d'échanges d'informations à travers notamment des démonstrateurs présentant les capacités du produit à répondre aux interrogations du client,
- se former auprès des services de formation spécialisés,
- obtenir des documentations techniques chez les fournisseurs ou à partir de la communauté d'utilisateurs du produit,
- contacter les forums de discussion à travers lesquels il est possible d'obtenir des réponses aux questions que « l'utilisateur » peut se poser.

Parmi ces possibilités, les informations recherchées par l'« utilisateur » de COTS doivent répondre à son besoin :

- d'identifier des produits comme étant des COTS (donc pouvant être intégrés),
- d'identifier les fonctionnalités utiles pour de futurs projets de développement ou correspondant avec le domaine d'activité de « l'utilisateur » et,
- d'évaluer leur qualité logicielle.

Cela correspond aux deux familles d'informations que nous avons définies précédemment :

- les « données produit » permettant l'identification de ses fonctionnalités,
- les « données qualité » permettant l'évaluation qualité par « l'utilisateur ».

Celles-ci sont définies pour l'activité de veille technologique et de définition des besoins décrite dans la Figure 17.

Pourtant, lorsqu'on essaye d'obtenir ces informations, le parcours est souvent difficile comme nous le montrons dans l'exemple suivant : l'intégration d'un produit de chez Microsoft « Office Excel 2003 ». Ce produit peut être intégré de plusieurs manières. Le site de l'éditeur propose, dans une rubrique pour les développeurs, des informations techniques : « la base de connaissance Microsoft ». A l'aide d'un moteur de recherche nous pouvons obtenir des articles permettant à « l'utilisateur de COTS » d'extraire des informations. Dans la capture d'écran de la Figure 38, nous avons essayé d'obtenir des informations sur les possibilités d'intégration.



Figure 38 : Recherche de comment réaliser une intégration avec Excel 2003

L'article sélectionné dans la Figure 38, permettra d'obtenir une partie de la réponse. Les captures d'écrans des figures 39 et 40 décrivent en partie les technologies nécessaires à l'intégration (Figure 39) et décrivent des scénarios permettant l'intégration en même temps qu'ils proposent des informations sur d'éventuels cas de figure de dysfonctionnement ou de problèmes liés à cette intégration (Figure 40).

INFO : Éléments à prendre en compte pour l'automatisation côté serveur de Microsoft Office

[Voir les produits auxquels s'applique cet article](#)

Ancien n° de publication de cet article : F257757

Cet article peut contenir des liens vers des informations en langue anglaise (pas encore traduites).

Numéro d'article	: 257757
Dernière mise à jour	: mercredi 20 avril 2005
Version	: 7.0

Résumé

Les développeurs peuvent recourir à l'automatisation dans Microsoft Office pour créer des solutions personnalisées qui tirent profit des capacités et fonctionnalités intégrées dans le produit Office. Toutefois, si ce type de développement par programmation peut être implémenté relativement facilement sur un système client, un certain nombre de complications peuvent apparaître lorsque l'automatisation s'exécute à partir d'un code côté serveur tel que ASP (Active Server Pages), DCOM ou un service NT.

Cet article décrit les complications auxquelles les développeurs peuvent être confrontés, offre des alternatives à l'automatisation susceptibles d'accélérer les performances et suggère des étapes de configuration d'Office si l'automatisation côté serveur est incontournable. Toutefois, les développeurs doivent savoir que les suggestions présentées ci-dessous sont offertes uniquement à titre d'information. Microsoft ne recommande pas et ne prend pas en charge l'automatisation côté serveur de Microsoft Office.

REMARQUE : dans ce contexte, le terme « côté serveur » s'applique également à un code exécuté sur une station de travail Microsoft Windows NT ou Microsoft Windows 2000, à condition que la station Windows ne soit pas celle sur laquelle l'utilisateur a ouvert une session interactive. Par exemple, un code démarré par le Planificateur de tâches sous le compte SYSTEM est exécuté dans le même environnement qu'un code ASP ou DCOM « côté serveur » et est donc confronté pour la plupart aux mêmes problèmes. Pour plus d'informations sur les stations Windows et COM, reportez-vous aux sections « Plus d'informations » et « Références ».

Plus d'informations

Toutes les versions actuelles de Microsoft Office ont été conçues, testées et configurées pour être exécutées par l'utilisateur final sur une station de travail cliente. Elles requièrent un Bureau interactif et un profil utilisateur et n'offrent pas le niveau de sécurité ou de réentrée nécessaire pour répondre aux besoins des composants côté serveur conçus pour s'exécuter sans assistance.

Figure 39 : Informations sur la manière d'intégrer Office Excel 2003

À l'heure actuelle, Microsoft ne recommande pas et ne prend pas en charge l'automatisation des applications Microsoft Office à partir d'une application ou d'un composant client non interactif et sans assistance (y compris ASP, DCOM et les services NT), car Office peut présenter un comportement instable ou entraîner un blocage lorsqu'il est exécuté dans ce type d'environnement.

Si vous créez une solution fonctionnant dans un contexte côté serveur, vous devez dans la mesure du possible tenter de recourir à des composants ayant été conçus pour être exécutés sans assistance de manière fiable ou trouver des solutions de remplacement permettant à au moins une partie du code de s'exécuter côté client. Si vous choisissez d'utiliser une application Office à partir d'une solution côté serveur, vous pouvez constater qu'il lui manque de nombreuses capacités pour s'exécuter correctement et vous risquez de compromettre la stabilité de l'ensemble de votre solution.

Problèmes liés à l'utilisation de l'automatisation côté serveur de Microsoft Office

Les développeurs qui tentent d'utiliser Office dans une solution côté serveur doivent tenir compte de cinq éléments majeurs pour lesquels l'environnement entraîne un comportement imprévu d'Office. Si vous souhaitez exécuter votre code sans problème, ces situations doivent être corrigées et leurs effets minimisés autant que possible. Lorsque vous créez votre application, examinez ces problèmes avec la plus grande attention car il n'existe pas de solution globale capable de tous les traiter. Chaque type de conception vous oblige à les hiérarchiser différemment.

1. **Identité de l'utilisateur** : L'exécution des applications Office nécessite l'identité d'un utilisateur, même si elles sont démarrées par l'automatisation. Elles tentent d'initialiser les barres d'outils, les menus, les options, les imprimantes et certains compléments en fonction de paramètres figurant dans la ruche du Registre de l'utilisateur qui lance l'application. De nombreux services sont exécutés sous des comptes qui ne sont associés à aucun profil utilisateur (tels que les comptes SYSTEM ou IWAM_[nom_serveur]) ; par conséquent, Office risque de ne pas s'initialiser correctement au démarrage et de renvoyer une erreur sur la méthode **CreateObject** ou **CoCreateInstance**. Même si l'application Office démarre, d'autres fonctions risquent de ne pas fonctionner correctement sans profil utilisateur. Si vous envisagez d'automatiser Office à partir d'un service, vous devrez configurer soit votre code, soit Office de manière que le programme s'exécute avec un profil utilisateur chargé.

Figure 40 : Problème lié à l'intégration d'Office Excel 2003

Comme nous le voyons avec ces exemples de deux éditeurs de logiciels, les informations sont en fait, éparpillées dans plusieurs documents. La recherche puis l'étude de nombreux documents est nécessaire pour compiler les informations permettant réellement de réaliser une étude d'évaluation sur les « données produit ». Nous avons résolu ce problème à l'aide de notre SI en associant à chaque produit et à chaque service du produit des scénarios d'intégration qui permettent de compiler l'ensemble de ces données éparpillées dans plusieurs documents et lieux.

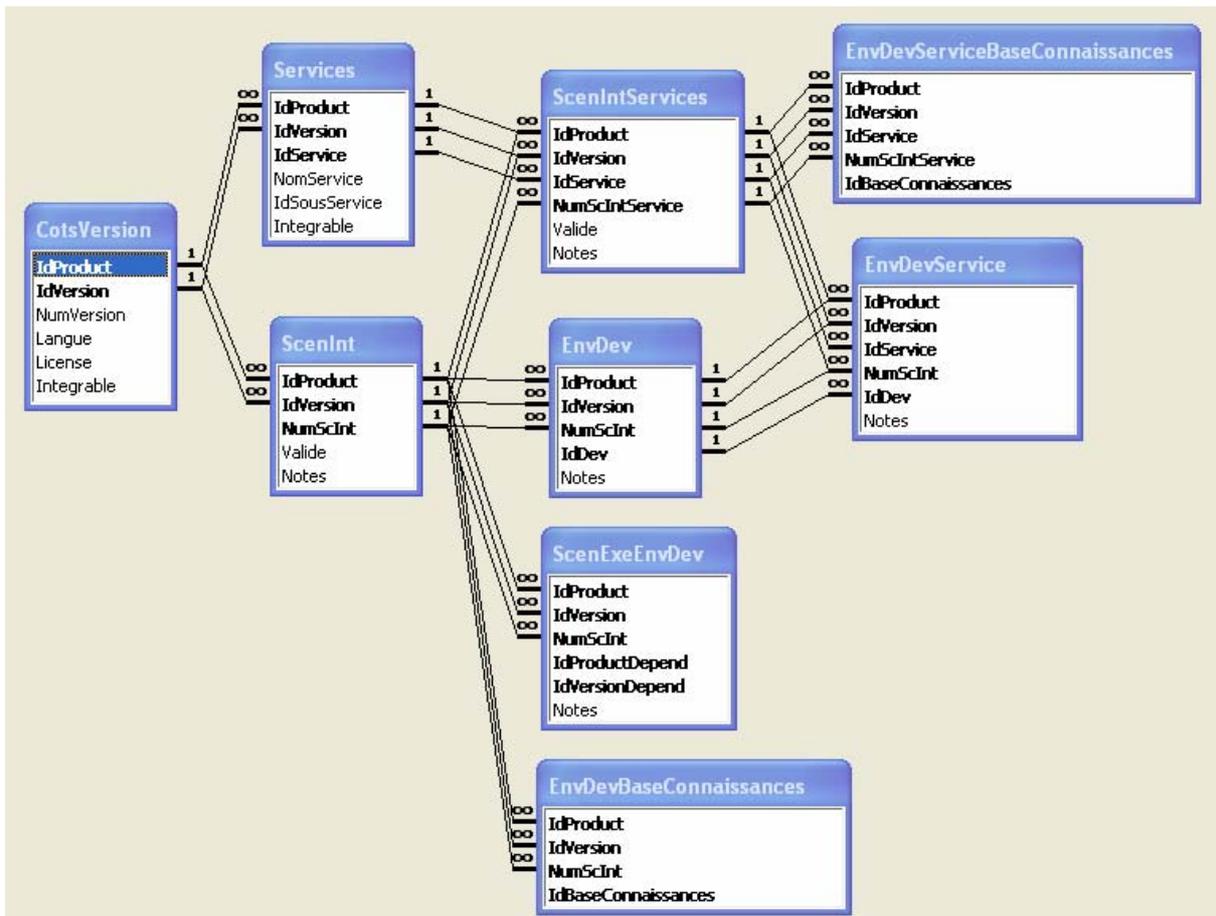


Figure 41 : Tables décrivant le scénario d'intégration

Dans notre SI, le scénario d'intégration est décrit à l'aide de deux tables « ScenInt » (pour Scénario d'Intégration du COTS) et « ScenIntServices » (pour Scénario d'Intégration des Services). Si la première table sert à déterminer le cadre général permettant de réaliser l'intégration de tous les services, la deuxième table (« ScenIntServices ») permet d'informer le développeur sur la manière d'intégrer précisément un service du COTS. Chaque scénario est composé d'une table permettant d'établir la liste des logiciels nécessaires pour l'exécution du prototype intégrant le COTS (« ScenExeEnvDev » pour Scénario d'exécution pour l'Environnement de développement), d'une table permettant d'obtenir une aide spécifique à chaque scénario (« EnvDevBaseConnaissances » et « EnvDevServicesBaseConnaissances » pour Bases de Connaissances de l'Environnement de Développement des Services) et une table permettant de décrire l'ensemble des outils de développement et des langages nécessaires pour l'intégration (« EnvDev » pour Environnement de Développement). A l'aide de notre SI, la documentation est maintenant centralisée, regroupée par services et associée aux différents outils et logiciel nécessaires au développement. Nous avons donc facilité le recueil d'informations et ainsi réduit cette difficulté.

Cependant, si l'on désire obtenir des résultats sans notre SI sur les « données qualité », cela est encore plus difficile. Dans notre recherche chez les deux éditeurs, l'information n'a pas été trouvée. Cela ne veut pas dire qu'elle n'existe pas mais seulement que son accès n'est pas mis au premier plan. Ces données peuvent correspondre aux attributs d'une norme qualité comme l'ISO9126 sur la qualité logicielle [ISO9126]. La plupart de ces informations ne sont pas présentées par les éditeurs. C'est souvent dans la communauté des COTS que l'on trouve des informations utiles au remplissage de ce questionnaire sur la qualité logicielle. Comme exemple, nous proposons le site d'un fournisseur de produits : « 01 Informatique ». Son site (www.telecharger.com) présente une liste de produits classés par système d'exploitation et famille de produits (capture d'écran de la Figure 42).



Figure 42 : Liste de produits classés par système d'exploitation et familles sur le site de « 01 informatique »

Dans notre SI il est facile d'obtenir ces informations (si le fournisseur de COTS les a inscrites dans le Système d'Information) dans la mesure où la liste d'attributs de la norme ISO9126 est présente dans une table ayant une relation 1..n avec chaque COTS et services du COTS.

La description des produits est souvent plus succincte que celle proposée par les éditeurs. Un lien vers le site de l'éditeur permet parfois de la compléter. Avec ces descriptions, il est possible d'accéder aux opinions des autres utilisateurs du produit. Ces informations provenant de la communauté des COTS permettent d'enrichir la base de connaissance de « l'utilisateur » sur le produit, notamment sur la qualité logicielle, mais nous n'avons aucun contrôle sur les informations déposés. Dans l'exemple pris dans le site de « 01 informatique », nous avons choisi la description d'un produit ayant un certain nombre d'opinions de la part des internautes dont voici la capture d'écran apparaît en Figure 43.

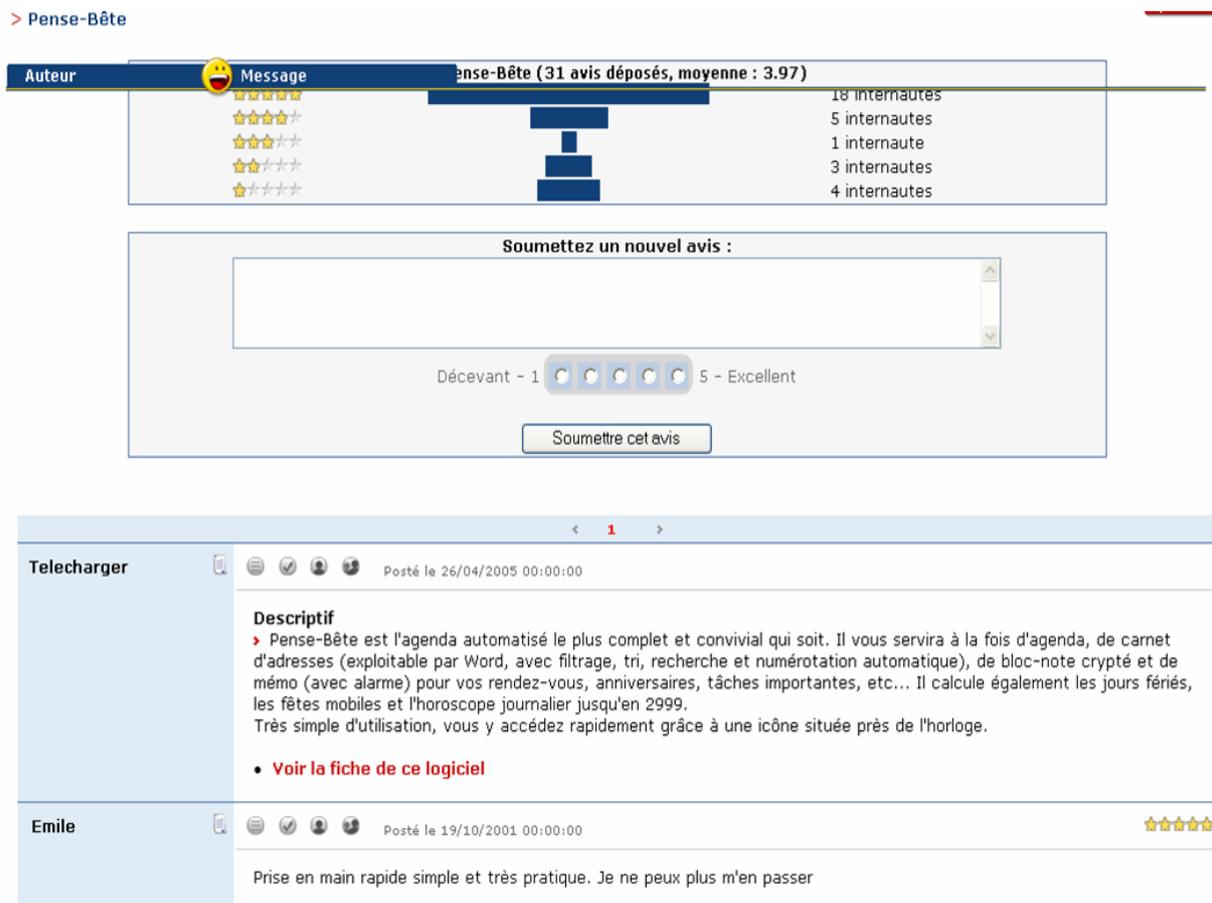


Figure 43 : Opinions sur un produit proposé sur « Télécharger.com »

L'exactitude de ces informations n'est contrôlée par aucun éditeur ou organisme, de ce fait, les informations sont à vérifier. Notre acteur pourra le faire dans sa phase de veille technologique ou dans une phase ultérieure d'évaluation des COTS.

Dans tout les cas, l'information est :

- disséminée dans plusieurs réponses des internautes,
- parfois contradictoire comme dans l'exemple ci-dessus ou des internautes vont dénigrer un produit alors que d'autres vont le définir comme extraordinaire,
- non certifiée et donc elle doit être contrôlée par notre acteur dans la phase de veille technologique ou d'évaluation des COTS.

Pour cela, dans le SI, nous proposons que toutes les informations introduites aient un propriétaire (l'utilisateur responsable de la modification ou de l'ajout d'information dans une fiche). Les données certifiées seront celles du fournisseur du COTS. Si ce dernier modifie ou valide l'information d'un utilisateur, la fiche apparaîtra comme certifiée dans le SI.

III - 3 L'aide à l'intégration des COTS

De tout cela, on constate une difficulté réelle pour obtenir des informations sans notre SI. Surtout, lorsqu'elles sont sommaires et non structurées. Pourtant, les concepteurs de COTS possèdent de nombreuses informations sur leurs produits, à commencer par les spécifications qui ont permis le développement, mais aussi les informations qu'ils ont récupérées pendant les phases de test.

Même si certains concepteurs ne vont pas divulguer la totalité des informations de peur de perdre leur savoir ou de se faire dérober leur technologie, il en existe un certain nombre qu'ils seraient prêts à diffuser sans compromettre leur intégrité.

En fait, le type d'information qui pourrait intéresser un client ne semble pas être établi. Peut être aussi qu'il est difficile d'établir la liste de toutes les informations qui seraient utiles aux différents clients. Pourtant, le problème est ainsi posé : **il faudrait connaître la liste de toutes les informations utiles pour chaque phase du processus de développement et cela, pour chaque type d'application que les clients seraient à même de réaliser avec le COTS du fournisseur.** Cela permettrait au concepteur d'identifier et de sélectionner les informations qu'il voudra bien diffuser, et aux clients d'avoir les données correspondant à leurs besoins. Cependant, les COTS sont parfois utilisés dans une application ou un domaine non prévu par le concepteur. Cela rendra difficile l'identification de tous les cas d'utilisations possibles des produits. De plus, les informations indispensables pour un client, peuvent être optionnelles pour d'autres. Les

spécificités des méthodes de travail ou des contraintes qualité utilisées par chaque client peuvent rendre difficile l'identification de toutes ces informations à fournir.

A la vue de la complexité de la tâche, surtout dans la difficulté de définir une exhaustivité des informations utiles dans chaque cas d'utilisation, **nous proposons d'identifier un nombre suffisant d'informations pour réaliser les opérations les plus usuellement utilisées dans le cadre de la réalisation d'un prototype.** Pour le reste, nous préférons **établir une relation client / fournisseur permettant** de terminer la liste de données manquantes et **d'enrichir une structure d'information existante.**

Cette approche a le mérite d'établir une relation entre client et fournisseur pour la diffusion des informations utiles à l'intégration du COTS et de permettre une capitalisation des informations diffusables pour chaque COTS.

Tout cela nous amène à penser qu'il est possible d'aider l'utilisateur de COTS dans l'intégration des COTS s'il avait à disposition des informations manipulables facilement (format « XML » par exemple car accessible simplement avec un éditeur texte présent dans tous les systèmes d'exploitation). Nous avons vu que les « données produit » sont accessibles en partie et de manière non structurées et disséminées dans plusieurs documents ou lieux. De ce fait, pour réaliser les différentes tâches du processus de développement, les utilisateurs de COTS sont obligés de rassembler les informations, de vérifier parfois leur intégrité pour pouvoir les utiliser dans la comparaison de produits, les phases d'évaluation de coûts et de risques ainsi que dans la réalisation de prototypes.

D'après Leung [Leung02], la réalisation des prototypes est indispensable à plusieurs phases du processus de développement. Donc, nous considérons que pour aider l'utilisateur de COTS il faut pouvoir lui proposer les informations utiles à sa réalisation. Par contre la difficulté est la même : les informations sont toujours aussi pénibles à trouver, non certifiées et donc à contrôler. La difficulté est d'autant plus grande qu'elles sont disséminées dans plusieurs documents sans aucune structure permettant leurs manipulations.

C'est pour cela que notre proposition est basée sur l'utilisation d'un Système d'Information qui, d'une part, met en relation les acteurs de la construction d'une application à base de COTS pour la réalisation d'un prototype, d'autre part, permet de recueillir et de vérifier les informations utiles au processus de développement. C'est dans l'aide à la réalisation du prototype que notre proposition est centrée. C'est aussi la mise en relation des différents acteurs qui est prioritaire dans notre proposition. C'est à l'aide de ces relations client / fournisseur (dans le cadre des informations utiles à ce processus de développement) que l'obtention des informations propres à chaque client reste possible.

Dans le paragraphe suivant nous décrivons la structure du SI permettant :

- d'établir une relation client / fournisseur pour l'échange d'informations,
- de définir les familles d'informations nécessaires à la réalisation d'un prototype (principal objectif de notre proposition),
- d'établir une structuration des données facilement exploitables par exportation et importation des informations,
- une certification de ces informations par les différents acteurs du système,
- de capitaliser les informations à échanger entre les fournisseurs et les utilisateurs de COTS permettant ainsi d'enrichir la structure de données du SI.

IV Les informations manipulées par le SI

IV - 1 La relation Client/Fournisseur

Cette relation est la base de notre système. Elle établit un lien permettant l'échange d'informations entre le demandeur d'information (l'utilisateur de COTS) et les diffuseurs (les acteurs « ForCo » et « Communauté de COTS »).

Cela se traduit par la mise en œuvre de mécanismes d'échanges (Figure 44) :

- la diffusion d'information en l'enregistrant dans une base de données, image informatique du SI,
- la recherche et la récupération des informations dans le SI,
- la demande d'informations complémentaires (donc manquantes) afin qu'elles soient ajoutées dans le SI,
- le traitement de ces demandes d'informations par les fournisseurs « FourCo » et « Communauté de COTS » qui ajouteront les données manquantes dans le SI.

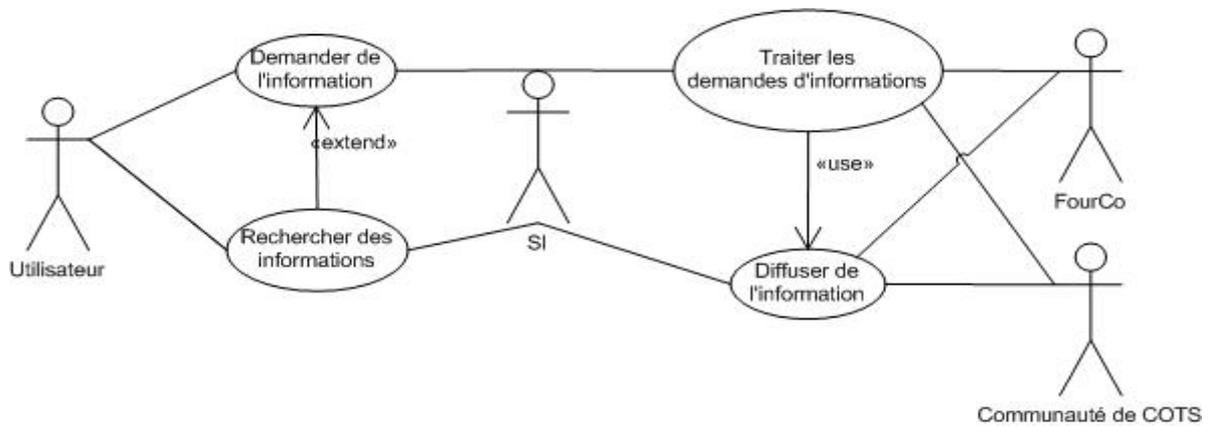


Figure 44 : Echange d'information entre acteurs

A travers ces échanges et le SI, il sera possible d'avoir une base de connaissance des COTS, avec la possibilité :

- d'extraire des données pour que les « utilisateurs de COTS » puissent réaliser des traitements comme par exemple des comparaisons entre produits ou pour l'utilisation dans des outils comme COCOMO ou COCOTS [ESI01],
- de l'enrichir par des informations complémentaires ou manquantes comme par exemple des scénarios de dysfonctionnement ou des résultats de tests d'évaluation.
- et surtout d'identifier les moyens nécessaires à l'intégration des COTS dans un prototype.

Dans ce dernier cas, qui nous intéresse plus particulièrement, les moyens sont constitués de deux environnements différents : celui de l'exécution des produits et celui de leur intégration. C'est à partir de ces informations que nous faciliterons la réalisation des prototypes.

IV - 2 Structure d'information pour la réalisation de prototypes

Le développement d'un prototype nécessite un environnement permettant sa conception. Il est composé d'outils logiciels et matériels qui doivent être configurés de manière spécifique pour chaque COTS qui sera intégré dans le prototype.

Un prototype logiciel est une application qui nécessite deux types d'environnement : celui avec lequel il sera possible de réaliser le développement et celui permettant son exécution. Dans le premier cas, c'est l'ensemble des logiciels qui s'exécutent sur un ou plusieurs ordinateurs et qui sont paramétrés pour permettre l'exécution du prototype. Nous appellerons cette configuration un scénario d'exécution. A tout scénario de fonctionnement il existe des contraires, les scénarios de dysfonctionnement. Ceux-ci, décrivent un ensemble logiciel et les paramétrages de ces derniers ne permettant pas partiellement ou totalement l'exécution du prototype. Par exemple on identifie que le produit X est incompatible avec les versions de Windows XP.

L'environnement de développement est composé des outils logiciels, des langages de programmation et des connaissances nécessaires à la mise en œuvre du développement logiciel du prototype intégrant le COTS. Cet environnement est spécifique à chaque COTS. Dans le cas des outils logiciels, on trouve par exemple le compilateur d'un éditeur. Pour chaque outil, il faudra préciser en plus le langage de programmation préconisé. Associées à ces outils de développement et ces langages, les fournisseurs de COTS proposent des informations sur la manière d'intégrer leurs produits à l'aide d'exemples sous forme, bien souvent, d'extraits de programmes écrits spécifiquement pour chaque outil et langage. C'est une partie des connaissances que l'on associe à l'environnement de développement.

De ce fait, nous proposons d'identifier le projet de réalisation d'un prototype avec les deux types d'environnement : exécution et développement. Pour chacun d'entre eux, nous définirons le scénario de fonctionnement et de dysfonctionnement. Dans la Figure 45 nous proposons notre représentation des informations nécessaires à la réalisation du prototype.

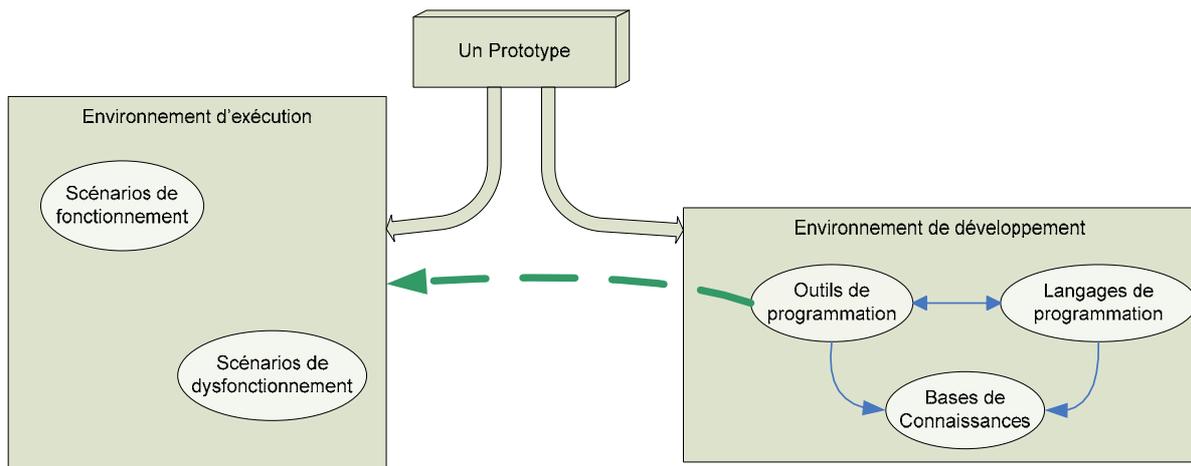


Figure 45 : Informations nécessaires pour la réalisation d'un prototype

Cette représentation nous servira pour définir les informations nécessaires à l'intégration d'un COTS. On pourra se rendre compte de certaines similitudes dans le paragraphe suivant avec l'intégration d'un COTS.

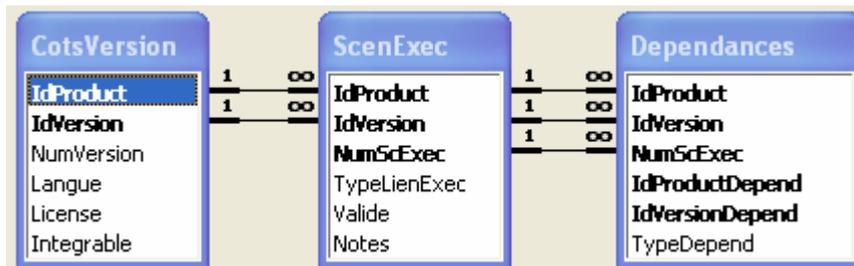


Figure 46 : Tables décrivant le scénario d'exécution

Dans notre SI nous utilisons une Table « ScenExec » (pour Scénario d'exécution) qui permet à l'aide de son attribut « Valide » de différencier le scénario de fonctionnement et de dysfonctionnement. D'autre part, il établit une relation 1..n avec l'ensemble des références aux logiciels qui sont nécessaires pour chaque scénario que nous retrouvons dans la table « Dependances » avec les attributs « IdProductDepend » (comme Identifiant du Produit) et IdVersionDepend » (comme Identifiant de la Version). Cela permettra d'identifier tous les logiciels nécessaires au bon fonctionnement (ou au dysfonctionnement). Chaque référence à un logiciel peut amener d'autres scénarios qui compléteront la liste des outils nécessaires à l'exécution du COTS.

IV - 3 Structure d'information pour l'intégration d'un COTS

En partant de la Figure 45 et de la description des informations nécessaires à l'intégration d'un COTS dans le paragraphe Chapitre 3 : III, page 88, nous pouvons décrire les informations indispensables pour réaliser cette intégration. Nous les décomposons en deux groupes principaux qui sont l'environnement d'exécution et l'environnement d'intégration. Le premier environnement est le même que celui défini dans le paragraphe précédent pour la réalisation d'un prototype. Dans le cas de l'environnement d'intégration, il doit proposer les moyens nécessaires au développement d'un prototype et à l'intégration d'un COTS. Pour cela, il est nécessaire de décrire des scénarios qui le permettent et de connaître les scénarios qui posent des problèmes d'intégration. Si le premier cas définit la marche à suivre, le second cas doit identifier des incompatibilités ou des difficultés entraînant un dysfonctionnement partiel ou total.

Dans tous les cas ces scénarios sont liés à un environnement de développement identique à celui présenté dans le paragraphe précédent. Par le biais de cet environnement, nous pouvons aussi avoir un lien entre les outils de programmation et l'environnement d'exécution comme présenté dans la Figure 47. Car ces outils sont des applications qui nécessitent toutes un environnement d'exécution qu'il faut définir.

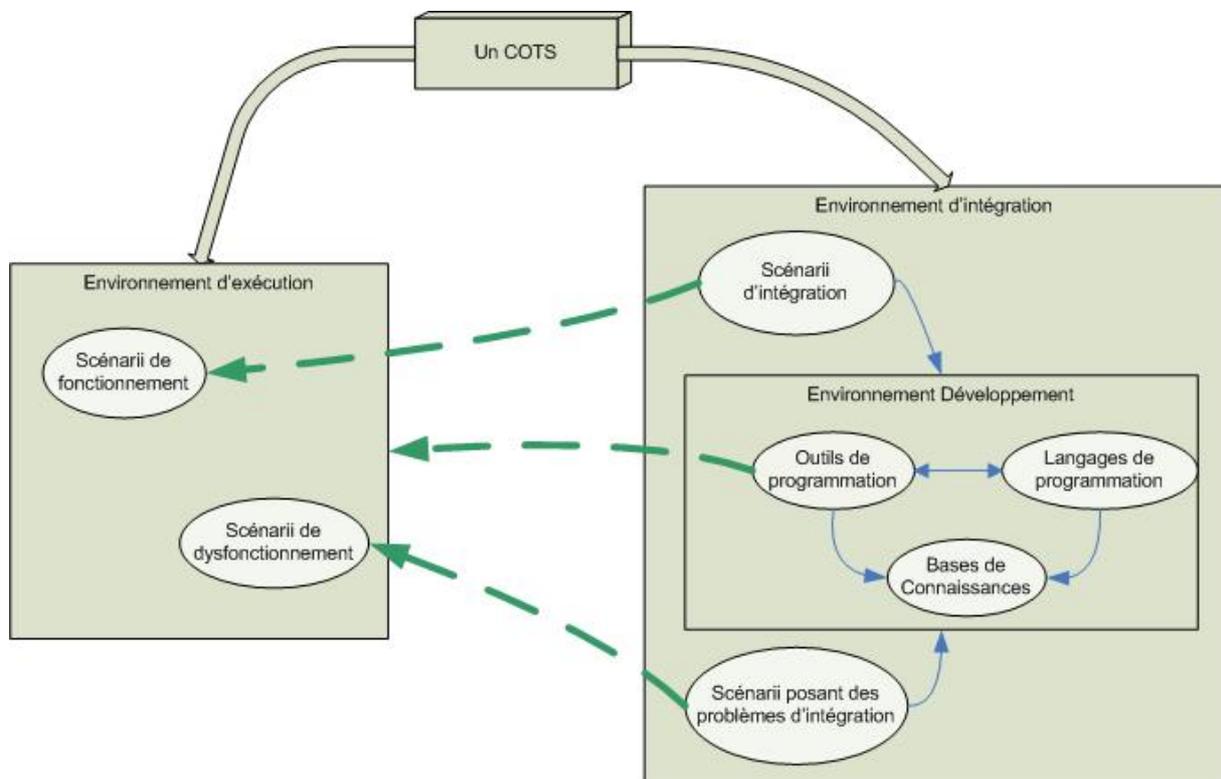


Figure 47 : Informations nécessaires pour l'intégration d'un COTS

Ce lien est le même que celui présenté pour la réalisation du prototype. Il permet d'établir l'environnement d'exécution des outils de programmation. Celui-ci peut être différent de celui de l'exécution du COTS. C'est le cas par exemple d'un COTS pour PDA dont l'environnement d'exécution est celui du PDA. Par contre l'outil de programmation a besoin d'un environnement d'exécution différent pour intégrer le COTS. Chaque scénario d'intégration a besoin pour sa réalisation d'un environnement spécifique à l'intégration comme dans l'exemple du PDA qui nécessite d'utiliser un émulateur. Le but de ces liens est de définir tous les éléments logiciels qu'il va falloir utiliser pour arriver à concevoir un prototype intégrant le COTS.

Dans la Figure 48, nous avons la structure de données permettant à notre SI de décrire tout cela. Nous avons ajouté aux tables déjà décrites dans ce chapitre, les tables « LangProg » (pour langage de programmation) et « OutilsDev » (pour Outils logiciels de développement). La table « langage de programmation » permet d'identifier pour un environnement de développement le langage de programmation à utiliser pour intégrer le COTS ou le service du COTS. D'autre part la table « OutilsDev » permet d'identifier, via la référence à un produit (attributs « IdProduct » et « IdVersion »), l'outil permettant de réaliser l'intégration dans un environnement donnée (table « ScenExeEnvDev ») et avec une documentation permettant d'aider le développeur dans son intégration (table « EnvDevBaseConnaissances »).

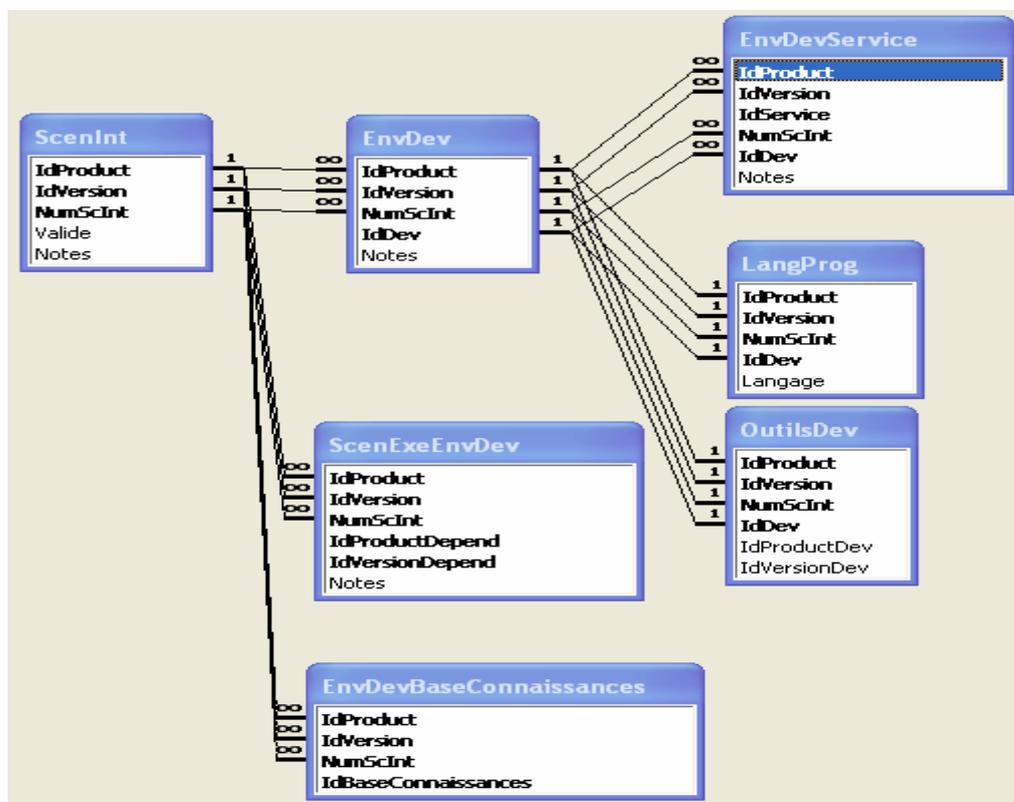


Figure 48 : Tables décrivant l'environnement d'intégration

Nous représentons ici l'ensemble des tables nécessaires à notre SI pour décrire les scénarios d'intégrations du COTS comprenant l'environnement de développement, les différents produits nécessaires à son exécution et l'ensemble des connaissances permettant la réalisation de son intégration. Le scénario d'intégration est l'un des éléments nécessaires pour l'utilisation d'un COTS dans un prototype.

Nous proposons pour cela la Figure 49 qui regroupe l'ensemble des besoins du prototype et des COTS.

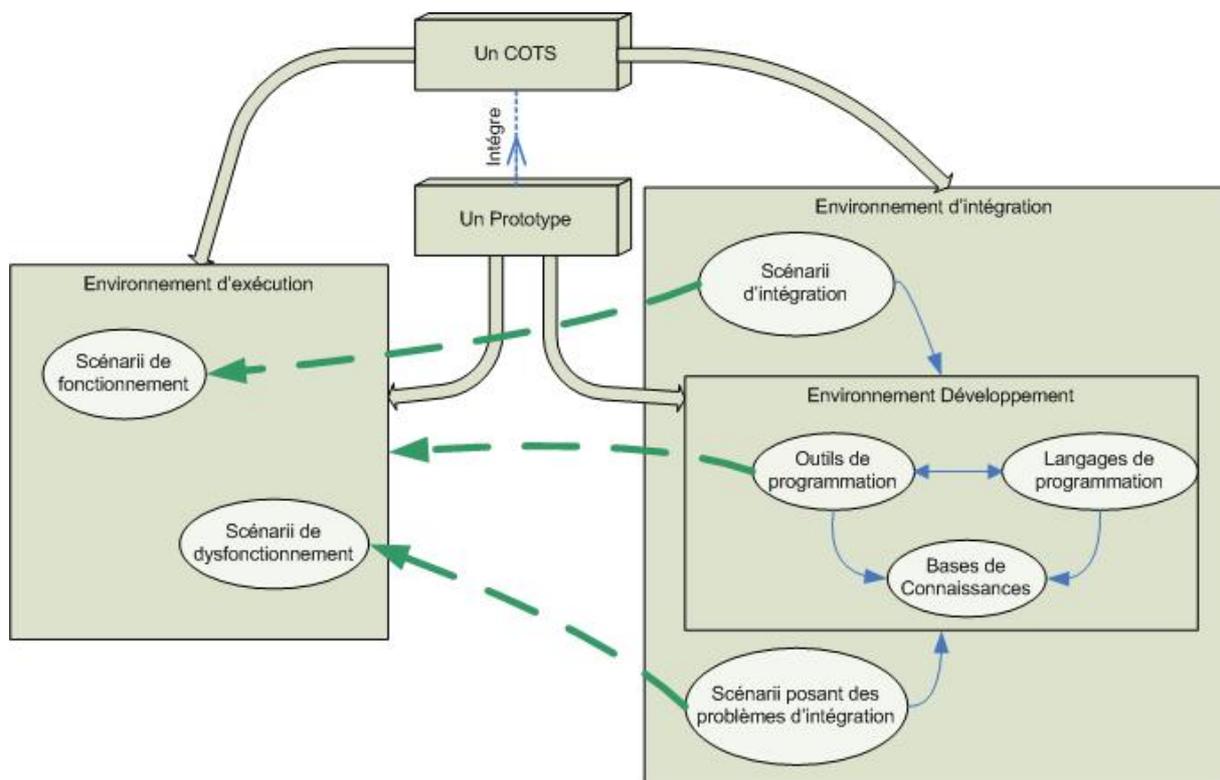


Figure 49 : L'ensemble des informations pour intégrer un COTS dans un prototype

Ces informations sont fournies par les acteurs « FourCo », complétées par les acteurs appelés les clients durant le processus de développement d'une application à base de COTS. En fait, elles doivent toutes être rattachées aux produits et donc à la carte d'identité des COTS (ICOTS – [Michel05]). C'est à partir de cette carte d'identité que nous affectons les informations concernant les environnements d'intégration et d'exécution. De ce fait, lorsque les fournisseurs de COTS ont besoin de décrire un environnement pour leur produit ils peuvent faire référence à une autre carte d'identité qui elle-même décrira son propre environnement. Ainsi, par références successives, il est possible d'obtenir la liste de tous les éléments logiciels nécessaires pour un environnement donné et donc de détecter des incompatibilités. Par exemple si dans un prototype nous assemblons un « COTS A » et un « COTS B ». Le premier a besoin pour son exécution d'un produit « Y » qui utilise un

produit « Z » mais incompatible avec l'utilisation du produit « X ». Le second a besoin d'un produit « V » utilisant le produit « X ». Dans cet exemple nous pourrions conclure que l'utilisation simultanée du COTS A et B n'est pas possible : c'est donc une incompatibilité que le développeur devra résoudre. Pour cela, il peut choisir d'utiliser l'un ou l'autre des COTS et de développer les services non remplis par le COTS manquant, ou alors de remplacer l'un d'entre eux.

De fait, si l'enrichissement de ces informations est fait par tous les acteurs du développement de COTS et d'application à base de COTS, nous devons nous poser la question de la validité de l'information. Une solution serait de la certifier comme détaillé ci-après.

IV - 4 La nécessité de certifier les informations

Dans notre démarche, les informations sont manipulées par le fournisseur et le client. Si le fournisseur peut garantir l'exactitude des informations qu'il introduit dans le SI, on ne peut pas demander la même chose de la part des clients qui constituent la « communauté des COTS ». De ce fait, les informations manipulées dans le SI doivent être identifiées en deux catégories : celles certifiées par un fournisseur ou organisme certificateur et celles apportées par les acteurs « utilisateur ». Ces dernières sont donc à vérifier car la source de l'information n'est pas obligatoirement fiable.

Nous considérons que les informations doivent avoir une identification sur la cohérence des données. Celle-ci peut être apportée par le fournisseur de COTS, un organisme de certification et par la cellule de certification de l'équipe en charge du développement de l'application à base de COTS. Le fournisseur et les organismes de certification peuvent, après vérification, intégrer de nouvelles informations que des utilisateurs de COTS auraient proposées. De la même manière les équipes de développement, du fait même des processus utilisés, vont devoir réaliser des évaluations et des qualifications. A partir de leurs résultats, il est possible de certifier les informations pour une utilisation interne à l'équipe de développement. Toutes ces informations ainsi certifiées (Figure 50) ne nécessiteront pas ou peu d'investigation pour les qualifier.

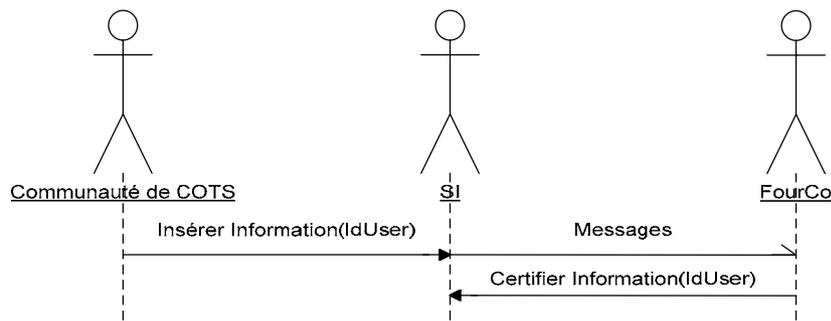


Figure 50 : Séquence de certification de l'information par "FourCo"

De plus, cette certification par les équipes de développement va dans le sens de la capitalisation de l'information. En effet si dans chaque projet d'intégration d'un COTS les développeurs remplissent les informations concernant les scénarios pour l'exécution et l'intégration (fonctionnement ou dysfonctionnement), les informations sur les produits peuvent être enrichies.

V L'utilisation du SI

V - 1 Présentation du Système d'Information

Le SI va permettre la diffusion, l'enrichissement des informations sur les moyens et les services qui pourront être intégrés dans une application, ainsi que la mise en relation du client demandeur d'informations et des acteurs pouvant en proposer. La base de cette relation d'échange d'informations a été décrite dans la Figure 44 sur laquelle nous revenons.

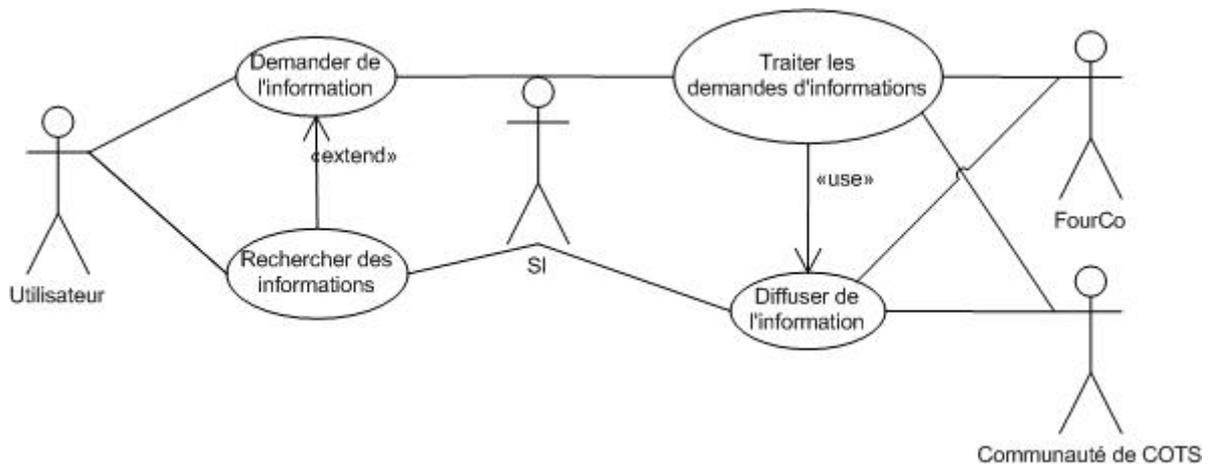


Figure 51 : Echange d'information entre acteurs

A partir de ce schéma, nous allons définir l'utilisation du SI par les deux types d'acteurs : les fournisseurs (à droite du schéma) et les clients (à gauche du schéma). Ils vont manipuler des informations contenus dans le SI et que ce dernier se propose de traiter pour les présenter en fonction des phases du processus de développement ou pour mettre en relation les acteurs entre eux.

V - 2 Les fournisseurs d'informations sur les COTS

Nous avons identifié deux acteurs (le fournisseur et le concepteur de COTS) que nous avons rassemblés en un seul « FourCo » puis « la communauté des COTS » (Figure 31) qui correspondent à des utilisateurs de COTS pouvant fournir des informations issues de leurs expériences.

Ces deux acteurs ont un rôle similaire dans le cadre de l'aide qu'ils peuvent apporter à l'utilisateur de COTS dans sa réalisation d'un prototype. Ils vont fournir de l'information pour sa réalisation et notamment dans l'environnement d'exécution du COTS mais aussi dans l'environnement de développement et de l'aide sur des éléments de programmation pour l'intégration des COTS.

Lors de la création du COTS, « FourCo » va devoir établir une fiche d'identification correspondant à son produit, à laquelle on associera les « données produit ». C'est à partir de cette carte d'identité que d'autres informations pourront être rattachées.

Cette fiche, nous l'appelons la carte d'identité de COTS (ICOTS) [Michel05]. Elle permet l'identification unique d'un produit comme on peut le trouver dans les cartes

réseaux avec l'adresse MAC (ou adresse matérielle). Il est donc nécessaire de standardiser le codage de cet identifiant pour que cela puisse être possible. A ce jour, et dans la mesure où cet identifiant n'existe pas encore, nous avons attribué un code, dans notre SI, construit à partir du nom du concepteur de COTS et de l'identifiant du produit de ce dernier.

Carte d'identité de COTS : ICOTS	
Identifiant Unique :	Microsoft123604
Nom du concepteur :	Microsoft
Nom du Produit :	Microsoft Outlook
Forme Binaire :	Exécutable
Catégorie du Produit :	Client de messagerie
Principaux Services	
Envoi et réception de courrier	
Carnet d'adresses	
Agenda	
Autres services ...	
Licence :	Payante
Plateforme :	Windows

Figure 52 : Exemple de carte d'identité de COTS (ICOTS)

Avec cet ICOTS (Figure 52) nous proposons d'identifier des informations assez générales mais qui sont utilisées dans la première phase du processus de développement à base de COTS : la veille technologique. Dans l'ICOTS, en plus de son identité (nom du produit, concepteur, forme binaire), nous identifions le produit dans une catégorie ou famille, son type de licence et un descriptif de son utilisation. Ce sont des informations que l'on retrouve classiquement dans des sites présentant des produits en téléchargement comme nous l'avons déjà décrit dans le paragraphe III - 2, page 110.

Il est donc indispensable lorsqu'un produit est développé de créer la carte d'identité du produit et d'associer à celle-ci les informations à diffuser. En particulier, en ce qui nous concerne, les informations d'aide à la réalisation d'un prototype, nous les avons décrites dans le paragraphe précédent et elles concernent les différents environnements nécessaires à l'exécution et à l'intégration du produit. Lorsqu'on fournit la référence à un produit, on a donc besoin de cet identifiant. C'est d'ailleurs à l'aide de cet ICOTS que par exemple l'environnement d'exécution de l'environnement d'intégration est identifié. Ce dernier possède à son tour les informations concernant son environnement d'exécution qui peut faire référence à un autre ICOTS et ainsi de suite. D'après la Figure 44 page 123,

« ForCO » et « La communauté des COTS » sont reliés à l'activité « diffuser de l'information ». Cette dernière est l'œuvre d'une démarche spontanée ou le résultat d'une demande d'information de la part des utilisateurs de COTS que les fournisseurs d'informations ont traités par l'activité appelée « Traiter les demandes d'informations ».

Cette dernière est donc la conséquence de la mise en relation du client et du fournisseur par le SI. C'est en fait l'utilisation du SI par le client qui permet d'établir ce lien entre ces deux types d'acteurs (client/fournisseur), au même titre que la diffusion par le fournisseur de COTS de la carte d'identité a permis d'amorcer la relation entre ces acteurs.

V - 3 L'utilisateur de COTS

V - 3.1 Dans la phase de veille technologique et de spécification

En charge du développement d'une application à base de COTS, le développeur a deux activités principales à réaliser : la recherche d'informations correspondant à de la veille technologique et la définition des besoins correspondant à de la spécification du produit à réaliser.

Dans la première activité, il va devoir rechercher des informations sur les produits existant dans le commerce. Pour cela, il va vouloir télécharger les cartes d'identité des COTS. Ces dernières peuvent être proposées par chaque fournisseur de COTS ou par des portails Web sur des produits logiciels. Avec les ICOTS, les développeurs devraient pouvoir mener leur veille technologique sur les services habituellement utilisés dans les applications correspondant au domaine ou métier ciblé par le développement qui doit être réalisé.

Dans la seconde activité, les développeurs vont devoir définir cette application et en particulier les services rendus. C'est donc à partir de cette définition que le développeur va devoir créer la fiche du produit en cours de développement : la carte d'identité ICOTS. Dans celle-ci, le développeur renseigne sur les informations concernant le produit au fur et à mesure de son avancement dans chaque phase du processus de développement de son application. Il va donc utiliser les familles de données que nous avons introduites dans le chapitre 2 de ce mémoire. Par exemple, dans la première phase du processus, il est proposé de renseigner les « données de spécifications ». C'est en fait la définition du produit présenté dans la Figure 16.

De ce fait, nous pouvons résumer ces activités par la récupération des ICOTS pour de la veille technologique et la création de l'ICOTS de l'application à base de COTS. Dans celle-ci, le développeur définit les services rendu qui sont introduits dans un espace personnel du SI. Celui-ci correspond à une base d'informations propre et non diffusable dans l'état aux autres utilisateurs de COTS, ni vers les fournisseurs de COTS. Pour faire la différence entre les informations du SI appartenant aux fournisseurs et celles attribuées à l'utilisateur et dont il se sert pour la réalisation de son application, nous décomposons l'acteur via UML appelé SI en deux acteurs que nous nommerons « SI Web » pour le premier cas et « SI local » pour l'ensemble des informations manipulées au niveau de l'équipe de développement.

Nos schémas ci-dessous correspondent à notre proposition en UML de l'utilisation du SI par l'utilisateur de COTS dans le cadre de la veille technologique (Figure 53) et dans le cadre de la spécification des besoins (Figure 54).

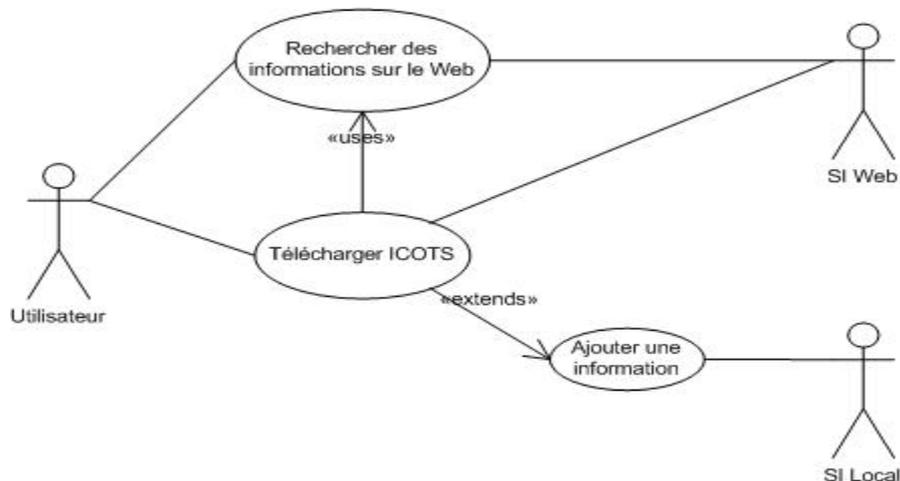


Figure 53 : Utilisation du SI dans la phase de veille technologique par les utilisateurs

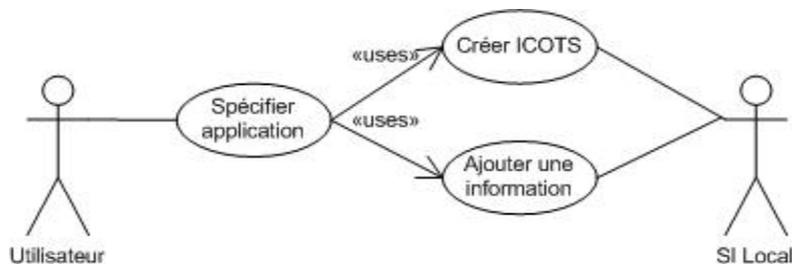


Figure 54 : Utilisation du SI dans la phase de spécification des besoins par les utilisateurs

Cette phase se poursuit avec la phase de sélection et d'acquisition des COTS que nous explicitons maintenant.

V - 3.2 Dans la phase de sélection et d'acquisition

Dans cette phase, les développeurs vont devoir associer les services de COTS (Figure 55) dont ils ont téléchargé l'ICOTS avec les services définis lors de la phase de spécification des besoins. Cette première mise en relation des services attendus avec des ceux qui sont rendus par des produits issus de la veille technologique permettra d'identifier au vu des informations recueillies si le ou les COTS sélectionnés peuvent passer à la phase suivante d'évaluation et de qualification.

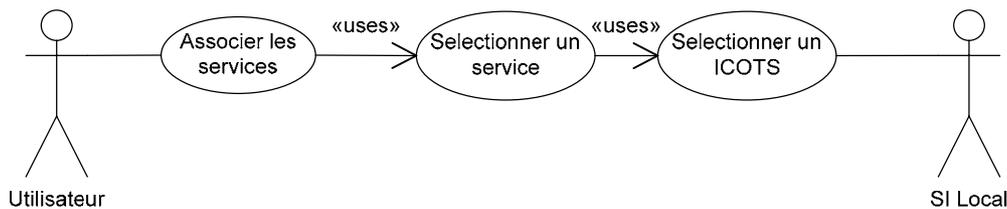


Figure 55 : Utilisation du SI dans la phase d'acquisition et de sélection par les utilisateurs

En fait, le nombre de services d'un COTS qu'il est possible d'utiliser dans une application est l'un des critères de sélection. La comparaison des informations du type plateforme cible ou correspondant à la qualité logicielle sont d'autres critères qui permettent le tri entre plusieurs candidats. Toutes ces informations sont portées par l'ICOTS dans le SI. L'utilisateur doit simplement parcourir la fiche du produit. Si toutefois l'information n'était pas présente, il pourrait demander auprès des fournisseurs de compléter l'ICOTS avec les informations manquantes.

Lorsqu'un COTS est sélectionné pour être évalué, il appartient aux développeurs dans les phases suivantes que nous avons regroupées dans le niveau décisionnel, de réaliser un prototype. Là aussi l'ICOTS et le SI vont pouvoir fournir des informations comme nous allons le voir ci-dessous.

V - 3.3 L'activité de réalisation d'un prototype

La réalisation d'un prototype permet de récolter des informations pour le compte des différentes phases du niveau décisionnel (évaluation, sélection, estimation des coûts et des risques, tri et sélection). Pour sa réalisation, le SI et les ICOTS vont aider le développeur dans son premier travail qui consiste à rechercher le plus d'informations possibles sur la manière de procéder pour réaliser l'intégration de tel ou tel COTS. Cette tâche qui est fastidieuse aujourd'hui car difficile à obtenir, devient plus facile dans la mesure où ces informations sont présentes dans l'ICOTS. Si ce n'était pas le cas, il est possible à l'aide du SI de les demander auprès des fournisseurs. Ces derniers peuvent répondre en mettant à

jour l'ICOTS. Nous proposons, modélisé en UML (Figure 56.), cette utilisation du SI pour la réalisation du prototype.

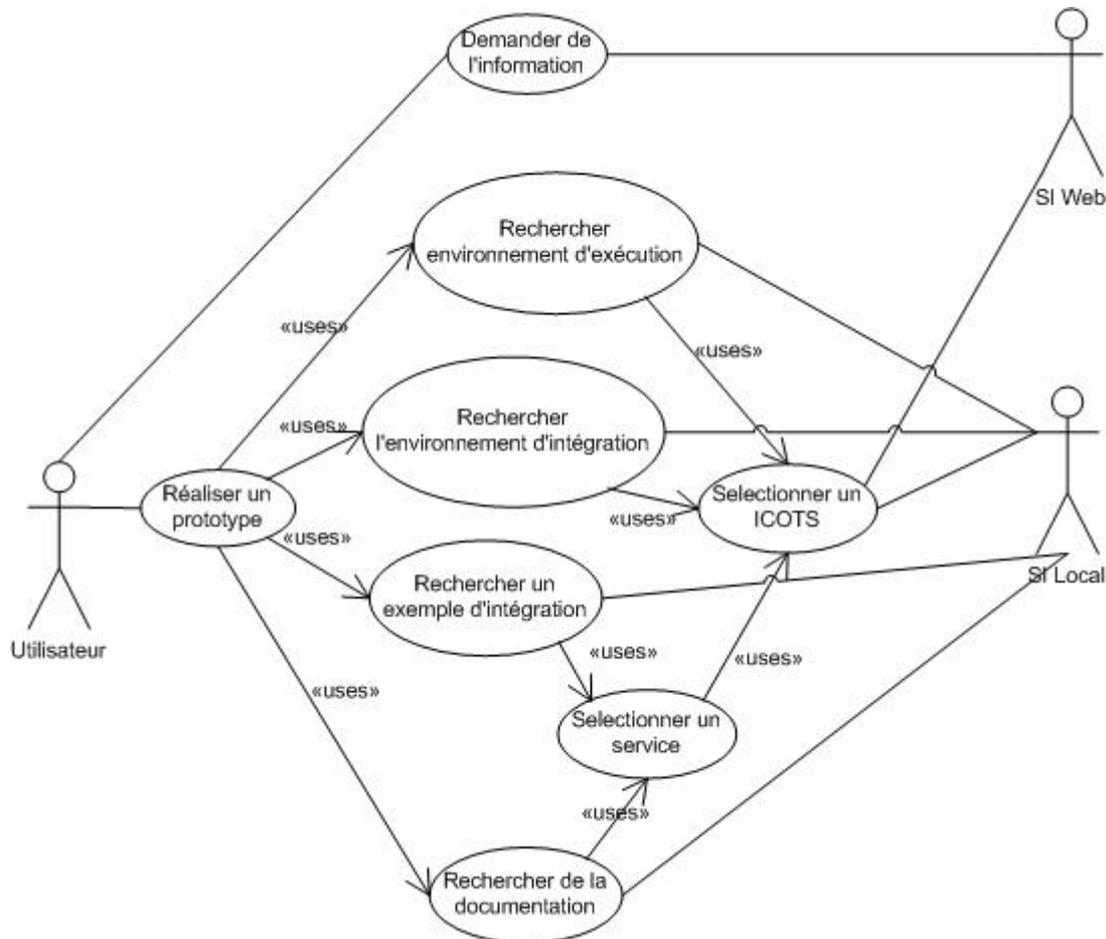


Figure 56 : Utilisation du SI dans le cadre de la réalisation du prototype

En référence à la Figure 45 qui décrit les informations nécessaires (environnement d'exécution et d'intégration) pour la réalisation du prototype, la Figure 56 montre l'utilisation du SI pour l'obtention de ces informations.

La réalisation du prototype utilise quatre activités de recherches qui sont :

- l'environnement d'exécution,
- l'environnement d'intégration,
- les exemples permettant l'intégration
- et la documentation.

Ces quatre activités correspondent à celles citées dans le chapitre III - 2, page 90. Elles permettent d'identifier une manière d'utiliser (environnement d'exécution et la documentation) et d'intégrer le COTS (environnement d'intégration, exemples et documentation).

Ces informations sont créées par les fournisseurs mais aussi par les utilisateurs de COTS au cours de la réalisation d'applications à base de COTS. C'est à partir de ces projets que les informations recueillies peuvent compléter les ICOTS à partir desquels les informations sont ensuite utilisées par un nouvel utilisateur de COTS ou dans le cadre d'un développement d'applications à base de COTS.

V - 3.4 L'activité de capitalisation des informations

Il est donc nécessaire que la définition d'un projet de réalisation d'applications à base de COTS commence par l'identification du produit et le renseignement des informations concernant la description des différents services, de l'environnement d'exécution (voire des scénarios de dysfonctionnement) et de l'environnement d'intégration si le produit est un COTS lui-même. C'est donc en renseignant le SI sur les caractéristiques du produit, la manière de l'intégrer, les résultats d'évaluation et de qualification que les développeurs vont capitaliser toutes les informations ainsi recueillies pour de futurs utilisateurs de COTS.

Cette expérience peut aussi intéresser les fournisseurs de COTS afin d'intégrer des scénarios complémentaires dans la description de leur produit. Comme nous l'avons précisé dans le paragraphe IV - 4, l'information pourra être certifiée par le fournisseur et servira directement à de futurs utilisateurs de COTS qui récupéreront plus d'informations qu'ils ne le font actuellement. De plus, le fournisseur propose un produit plus attractif car sa fiche sera particulièrement riche en informations.

V - 3.5 Les rôles des rapports de synthèse

Les objectifs principaux des rapports de synthèse sont d'identifier :

- des difficultés ou des dysfonctionnements,
- les moyens nécessaires pour l'intégration,
- les environnements d'exécution qui sont disponibles pour l'application et l'ensemble des COTS qui la composent.

L'ensemble de ces rapports aidera le développeur à maîtriser les multiples informations issues des différents produits qui vont composer son application. Certaines l'aideront dans la réalisation de l'intégration (exemple du scénario d'intégration) alors que d'autres l'avertiront sur des configurations d'exécution ou de développement ayant déjà causées des problèmes à d'autres utilisateurs de COTS (Figure 57). Il est évident que cela ne peut que favoriser l'utilisation des prototypes pour évaluer, qualifier et estimer les coûts et les risques. Nous pensons que cela permet un meilleur choix des COTS car les informations récoltées seront plus importantes, notamment pour la réalisation de ce prototype et car son utilisation améliore les résultats des évaluations et des qualifications.

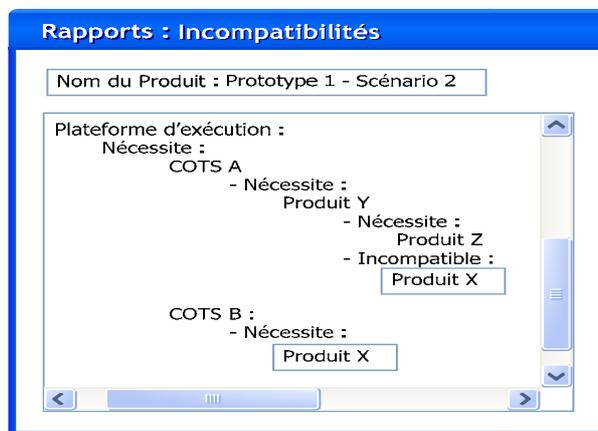


Figure 57 : Exemple de rapport montrant une incompatibilité

La description des différents environnements nécessaires à l'intégration des COTS (Figure 58) aidera le développeur à établir l'ensemble des moyens nécessaires à son développement.

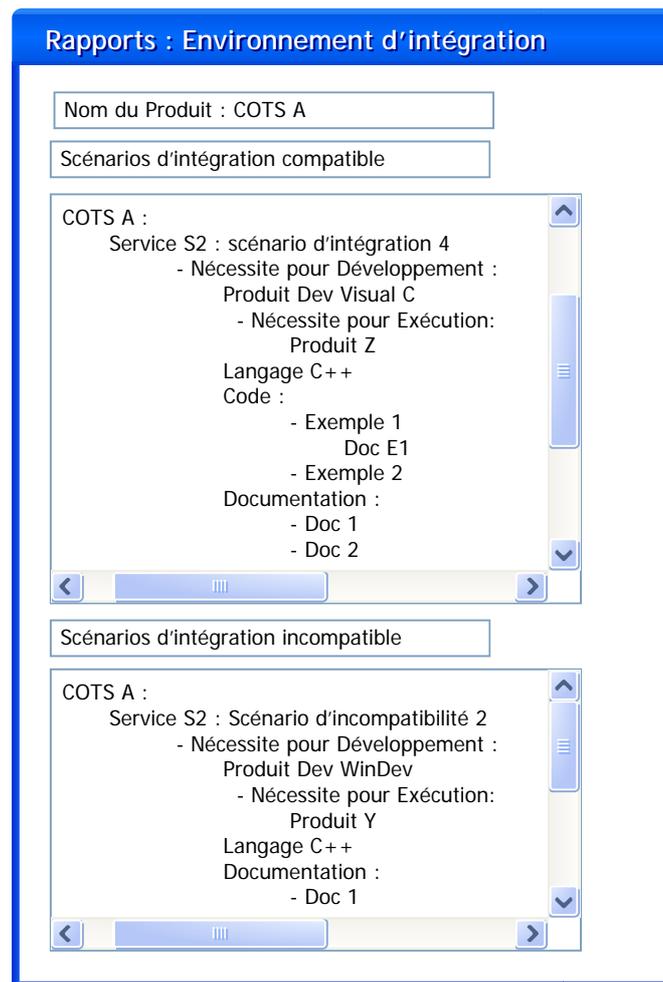


Figure 58 : Exemple d'information sur l'environnement d'intégration

Cet exemple montre les outils nécessaires, les langages qui sont proposés pour réaliser l'intégration d'un service spécifique du COTS « A » avec des exemples de code et la documentation associée. A partir de ces éléments et des scénarios d'incompatibilité qui montrent l'impossibilité de réaliser l'intégration avec le produit de développement « WinDev », les développeurs auront les moyens de réaliser plus facilement les prototypes puisqu'ils connaîtront l'environnement nécessaire à leur développement et auront à disposition la documentation relative à chaque service du COTS à intégrer.

VI Les données du SI

VI - 1 *Les acteurs*

Nous avons décrit les acteurs en trois catégories : utilisateurs, fournisseurs et communauté de COTS. Cela nous a permis de comprendre les différentes approches et les rôles de chacun. Par contre, dans la base de données qui va nous servir à gérer les différents acteurs, nous avons besoin de les identifier et de leur octroyer des droits d'accès différents en fonction du rôle qu'ils auront dans l'utilisation du SI. C'est notamment le cas pour la certification des informations rentrées dans le système. Il va de soi que l'on ne peut pas accorder le même crédit à une information provenant d'un utilisateur de COTS qu'à une autre validée par le fournisseur. Donc, pour cette action particulière ou pour ajouter des fonctionnalités à un COTS, il est évident qu'il faut associer à chaque acteur un rôle lui permettant d'avoir accès à des actions différentes du Système d'Information.



Figure 59 : Les tables pour gérer les acteurs

La Figure 59 correspond aux tables nécessaires pour identifier les acteurs (table « Acteurs »), pour identifier les produits sur lesquels les acteurs peuvent avoir un droit de modification et de certification des informations (table « Acces ») et une table permettant d'identifier les différents produits logiciels manipulés (table « CotsVersion »).

En fait, tous les acteurs pourront avoir le rôle correspondant à la « communauté de COTS ». Ils pourront répondre à des demandes d'informations de la part des « utilisateurs de COTS » en renseignant les fiches des COTS mais leurs informations seront dans un état non validé par le fournisseur, car il est le seul à pouvoir confirmer ou supprimer des informations relatives à son produit. Et pour obtenir ce droit, son identifiant sera associé à son produit dans la table « Acces ». De cette manière, seuls les produits identifiés avec des acteurs associés dans cette table permettront à ces derniers d'intervenir sur la fiche produit en modification, en suppression et en validation.

VI - 2 La carte d'identité ICOTS

Les produits sont identifiés par l'intermédiaire de la carte d'identité des COTS (ICOTS). Celle-ci est composée de plusieurs tables (Figure 60).

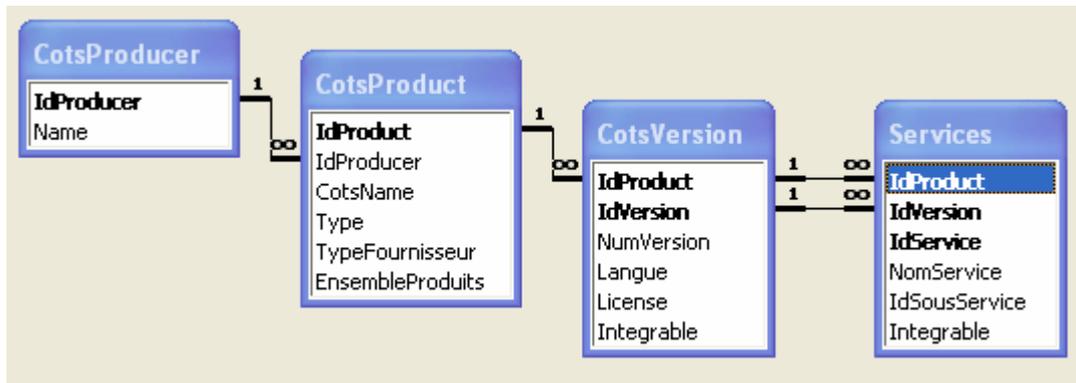


Figure 60 : Les tables permettant d'identifier un COTS

Les produits que nous allons manipuler dans le SI doivent être identifiés de manière unique à l'aide d'une clé d'identification que nous avons nommé ici « IdProduct » dans la Table « CotsProduct ». Chaque produit appartient à un fournisseur de COTS que nous identifions dans la table « CotsProducer ». Un fournisseur de COTS pouvant avoir plusieurs COTS, nous retrouvons la relation 1..n entre ces deux tables.

Chaque produit se décline en plusieurs versions. La table « CotsVersion » permet d'identifier les différentes variantes de manière à pouvoir rattacher à cette table les différentes informations ou fiches dont on a déjà parlé dans ce document (par exemple les services fournis, les « données qualité », les « données d'estimation » ou les « données de spécification »).

C'est le cas dans la Figure 60, avec la table « Services » qui permet d'identifier les services fournis par chaque version d'un COTS. En établissant une relation 1..n avec la table « CotsVersion », il est possible de rattacher plusieurs informations à la carte d'identité de COTS en les décrivant dans les trois tables « CotsProducer », « CotsProduct » et « CotsVersion ».

VI - 3 Informations associées à la carte d'identité

Il est possible d'associer plusieurs informations à la carte d'identité de COTS. Dans notre cas nous allons nous limiter aux informations utiles pour aider les développeurs à construire le prototype permettant d'évaluer les COTS à intégrer. En relation avec cet objectif, nous avons défini dans le Chapitre 4 : Le Système d'Information pour COTS, l'ensemble de ces informations (environnement d'exécution, d'intégration et de développement). Nous les retrouvons dans la Figure 61 sous forme de plusieurs tables et relations.

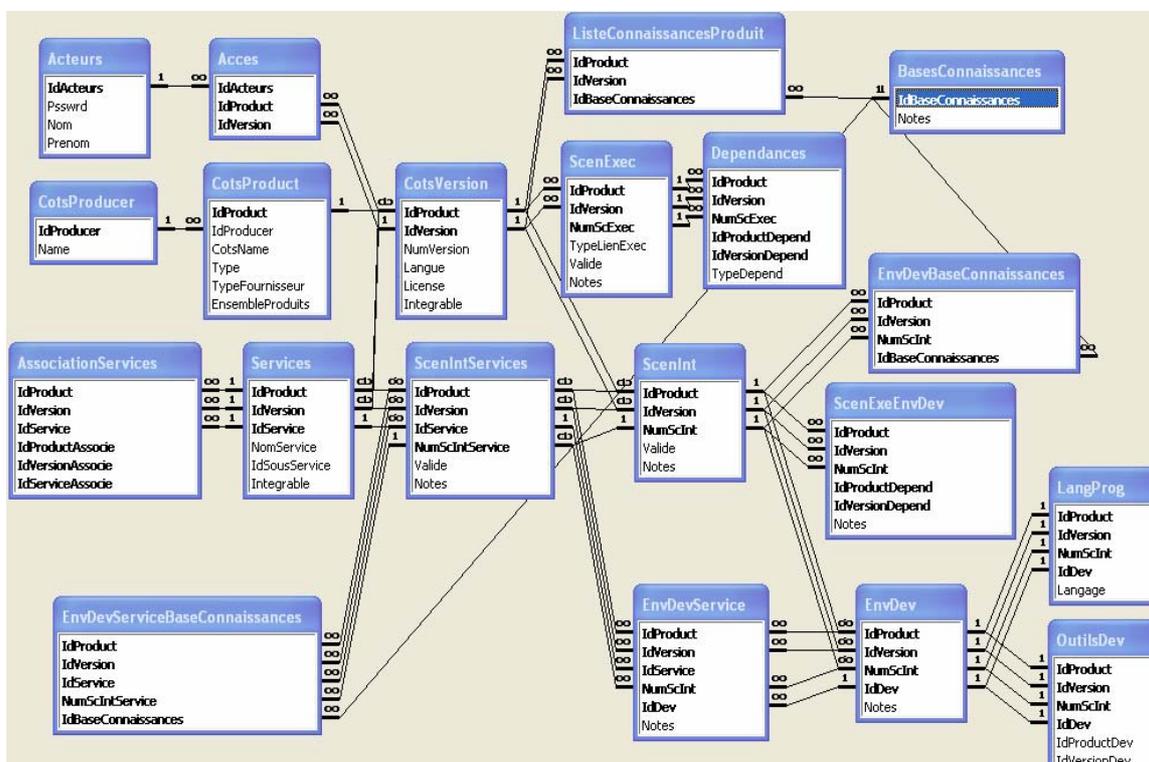


Figure 61 : Les tables pour l'intégration d'un prototype

Elles composent la base de données de notre SI. C'est à partir de cette structure que nous avons construit notre démonstrateur. Il permet de décrire les services des COTS qui sont intégrables, les moyens nécessaires à son fonctionnement et à son intégration puis il établit les rapports de synthèse que nous avons présentés dans le paragraphe V - 3.5 de ce chapitre.

VII Conclusion

Notre proposition de Système d'Information des COTS passe par l'utilisation de notre carte d'identité des COTS que nous avons appelé ICOTS. Cette dernière identifie le produit et fournit un descriptif de l'application permettant de donner un aperçu de son utilisation et de ses fonctionnalités. Son rôle au-delà de l'identification unique du produit doit permettre d'associer plusieurs informations utiles notamment dans un processus de développement d'applications à base de COTS. C'est dans les phases amont à ce processus (spécification des besoins) et dans les activités de qualification et d'adaptation que nous avons voulu apporter notre contribution. En effet, la réalisation d'un prototype est indispensable à ces activités. L'un des objectifs étant d'obtenir des informations qui sont utiles à la bonne marche du processus de sélection et d'intégration d'un COTS. Et comme nous l'avons vu dans l'état de l'art, c'était une difficulté majeure qui ne permettait pas la bonne réussite de l'intégration d'un COTS.

C'est donc avec l'utilisation d'un SI qui permet de rassembler les ICOTS de différents produits et de mettre en relation les fournisseurs et les clients que nous proposons d'améliorer la récupération des informations qui fait tant défaut. Le fait de rattacher de multiples informations autour de l'ICOTS, puis d'enrichir ces informations avec les résultats d'autres développements, doit permettre de réduire considérablement la difficulté de recherche d'informations. De plus, lorsque les fournisseurs décrivent un environnement d'exécution ou d'intégration d'un COTS (Figure 46 : Tables décrivant le scénario d'exécution, page 125), ils le font par le biais d'une référence à un autre ICOTS qui porte lui-même les informations du fournisseur du produit.

Nous pensons que la possibilité de manipuler des informations standardisées sous un format tel que « XML » favorisera le traitement des informations. Le support de ces informations est un format texte lisible depuis toutes les plates-formes informatiques ce qui le rend compatible et exportable. Le SI va ainsi aider les développeurs dans la réalisation de prototypes. Cela permettra de faciliter la récupération d'informations et réduira les difficultés d'utilisation des COTS que nous avons relevées dans l'état de l'art. Il favorisera aussi la mise en relation des fournisseurs et des clients.

Au-delà de la proposition d'utiliser un format « XML » pour l'échange de données entre acteurs, le SI que nous avons présenté dans ce chapitre permet d'aider essentiellement les développeurs à réaliser leurs prototypes. Pourtant, les informations manipulées sont aussi utiles dans d'autres phases du processus de développement comme la phase de sélection ou d'évaluation. L'utilisation que nous présentons de notre SI ne prend pas en charge la totalité des besoins spécifiques au déroulement de ces phases mais peut fournir des réponses partielles. Par exemple, il propose des réponses par rapport à des dysfonctionnements entre COTS mais en revanche il ne tient pas compte des critères de

sélections de ces mêmes COTS que nous avons définis dans les « données de spécification ». En fait, l'ensemble des familles de données (données produit, spécification, évaluation, estimation) présentées dans le « Chapitre 3 : Les informations pour l'utilisation des COTS » ne sont pas reprises dans notre proposition car elles ne sont pas indispensables à la réalisation du prototype. Cependant elles pourraient être utilisées dans le cadre d'une extension de notre SI pour aider les développeurs dans la totalité du processus de développement. Le démonstrateur que nous allons présenter dans le chapitre suivant pourrait guider la saisie et le traitement des informations relatives à toutes les phases du processus.

Chapitre 5 :

Le démonstrateur

I Présentation

I - 1 Les services rendus

Nous vous avons présenté notre Système d'Informations pour COTS qui a pour objectif d'aider les développeurs dans la réalisation des prototypes indispensables pour l'évaluation, la qualification et la sélection des COTS. Dans ce chapitre nous décrivons notre démonstrateur qui permet :

- L'enregistrement de la carte d'identité et des informations associées (pour les COTS et la future application à base de COTS),
- Le téléchargement des cartes d'identité pour visualiser les données contenues,
- L'identification des services des COTS utilisés dans l'application à base de COTS,
- L'identification pour chaque COTS des environnements d'exécution et d'intégration dans le but d'aider le développeur dans la réalisation de son prototype,

- L'accès aux informations relatives à l'intégration des COTS et ses services associés (outils de développement, exemples de codes, ...),
- L'identification des dysfonctionnements éventuels comme par exemple une non concordance entre des spécifications de l'application à base de COTS et les COTS employés.
- La mise en relation des clients et des fournisseurs pour obtenir ou demander des informations complémentaires et permettre la capitalisation des informations,

Le SI est supporté par une base de données qui détient l'ensemble des informations manipulées (Identification des logiciels, environnement d'exécution, environnement d'intégration) par les trois types d'acteurs que nous avons déjà mentionné dans le Chapitre 4 : III - 1 (« FourCo », « Communauté de COTS » et « Utilisateur de COTS »).

I - 2 Les objectifs

Ce démonstrateur est un outil nécessaire pour appliquer les propositions de nos travaux. Il reprendra de fait les informations nécessaires à la description des COTS dans l'objectif d'aider les développeurs dans la réalisation des prototypes. Nous avons présenté ces données dans le Chapitre 4 : Le Système d'Information pour COTS et notamment avec la Figure 49 : L'ensemble des informations pour intégrer un COTS dans un prototype. On y trouve les environnements d'exécution permettant le fonctionnement du prototype et de ses COTS, ainsi que l'environnement de développement permettant d'intégrer et d'utiliser les COTS.

Le démonstrateur permettra la saisie et la lecture des fiches ICOTS contenant ses propres informations. C'est un guide à la description des fiches correspondant à notre objectif d'aide aux développeurs. C'est aussi un moyen d'obtenir efficacement et rapidement un ensemble d'informations jusqu'ici difficiles d'accès. C'est d'ailleurs l'une des raisons pour lesquelles nous avons réalisé ces travaux. Le démonstrateur proposera des synthèses sous forme de rapports qui permettront une lecture plus efficace des nombreuses informations recueillies par le développeur en réalisant un travail automatique de compilation et d'analyse des données difficile à réaliser manuellement.

Ces rapports sont aussi l'un des objectifs de ce démonstrateur. On en présentera trois : sur l'environnement d'exécution, sur l'environnement d'intégration et sur les dysfonctionnements prévisibles. Le premier doit identifier l'ensemble des produits indispensables au fonctionnement du prototype et de ses COTS. Il identifie en même temps l'ensemble des logiciels identifiés comme étant incompatibles avec l'utilisation de ce

prototype. Le second permet d'accéder aux informations permettant la réalisation des développements conduisant à l'intégration des COTS. C'est par exemple le code nécessaire pour intégrer un service de COTS dans un langage de programmation donné et un outil de développement spécifique. On pourra aussi trouver la liste des outils, librairies et bibliothèques de programmation indispensables à cette intégration. L'objectif étant de fournir le plus d'informations possibles pour faciliter la réalisation de l'intégration. Enfin, le troisième et dernier rapport permettra d'identifier la liste des dysfonctionnements identifiables à l'aide des informations recueillis à partir de chaque COTS participant à la constitution du prototype ou de l'application à base de COTS. Il est ainsi un moyen de recueil d'informations qui peuvent être utiles pour les phases de sélection des COTS.

Un autre objectif que l'on propose avec ce démonstrateur, c'est la capitalisation des évaluations provenant de la réalisation d'autres prototypes. Chaque fois qu'une intégration est faite, il est possible d'indiquer avec quels environnements d'exécution et d'intégration les COTS ont été conçus. Mieux encore, il est intéressant de décrire des scénarios de dysfonctionnement, c'est-à-dire des environnements ne permettant pas l'exécution ou l'intégration du prototype et de ses COTS. L'enrichissement des fiches ICOTS permettront notamment d'établir le rapport de dysfonctionnement.

Enfin, nous pensons qu'il est intéressant de décrire les informations citées dans ce rapport. Leurs diffusions, leurs capitalisations et l'utilisation du démonstrateur sont un outil d'aide à la réalisation des prototypes. C'est en ce sens que nous pensons réduire la difficulté du recueil et du traitement des informations pour la réalisation de développements basés produits sur étagère.

II Utilisation du démonstrateur

II - 1 En fonction des acteurs

Nous avons présenté dans nos travaux, les trois acteurs du Système d'Information dans le Chapitre 4 : Le Système d'Information pour COTS et avec la Figure 31 : les acteurs du SI. Nous avons les fournisseurs d'informations avec « FourCo » et la « communauté des COTS » et l'utilisateur des COTS appelé « Utilisateur ».

Le premier accède au démonstrateur pour renseigner la fiche ICOTS correspondant au COTS qu'il met à disposition des utilisateurs. Il renseigne ainsi les différents

environnements proposés dans cette thèse. Puis il publie ces fiches pour que les utilisateurs accèdent facilement à l'information comme nous l'avons proposé dans la séquence UML de la Figure 32 page 110. Le démonstrateur et notre proposition servent ici à guider le fournisseur dans la description des informations indispensables à l'utilisateur de COTS. Le deuxième acteur (« communauté des COTS ») accède à ces mêmes fiches pour compléter les informations en ajoutant des scénarios complémentaires ou en enrichissant des scénarios existants. Par cette action l'acteur va capitaliser ses propres expériences qui serviront à de futurs utilisateurs de ces COTS. C'est un point important pour notre démarche car plus il y aura d'enrichissement de ces fiches et plus pertinentes seront les informations pour les futurs utilisateurs de COTS.

Enfin le troisième acteur utilise directement le démonstrateur pour récupérer les fiches ICOTS des fournisseurs de COTS, pour décrire le prototype en termes de services à rendre et d'environnement d'exécution cible. Ce démonstrateur lui permet ainsi d'accéder aux informations distribuées par les fournisseurs de COTS notamment pour savoir comment réaliser l'intégration des COTS. Il va aussi pouvoir obtenir des synthèses sur l'environnement d'exécution, l'environnement de développement et les dysfonctionnements possibles. Cela implique qu'il doit renseigner le démonstrateur sur les services qu'il désire intégrer en sélectionnant les services de chaque COTS impliqués dans son développement de prototype. La relation qu'il va ainsi établir entre les services d'un COTS et son application permettront au démonstrateur d'établir les rapports cités ci-dessus. Le démonstrateur permettra la manipulation des informations et pourra proposer des synthèses comme celles publiées dans des rapports (sur l'environnement d'exécution, sur l'environnement d'intégration et sur les dysfonctionnements). Il servira aussi de guide pour utiliser la démarche nécessaire à l'obtention de ces rapports.

II - 2 La démarche pour les utilisateurs de COTS

Nous présentons ici l'utilisation de l'outil par l'acteur « Utilisateur de COTS » car lui seul a besoin de réaliser le prototype permettant l'évaluation des COTS à intégrer. Nous donnons de ce fait la priorité à cet acteur puisque notre étude porte essentiellement sur l'aide que l'on peut lui apporter dans la réalisation de son prototype. Pour cela, après son identification, l'acteur va devoir suivre quelques étapes pour utiliser le démonstrateur comme nous pouvons le voir dans le schéma de la Figure 62. Elles représentent la démarche d'utilisation du démonstrateur pour les utilisateurs de COTS.

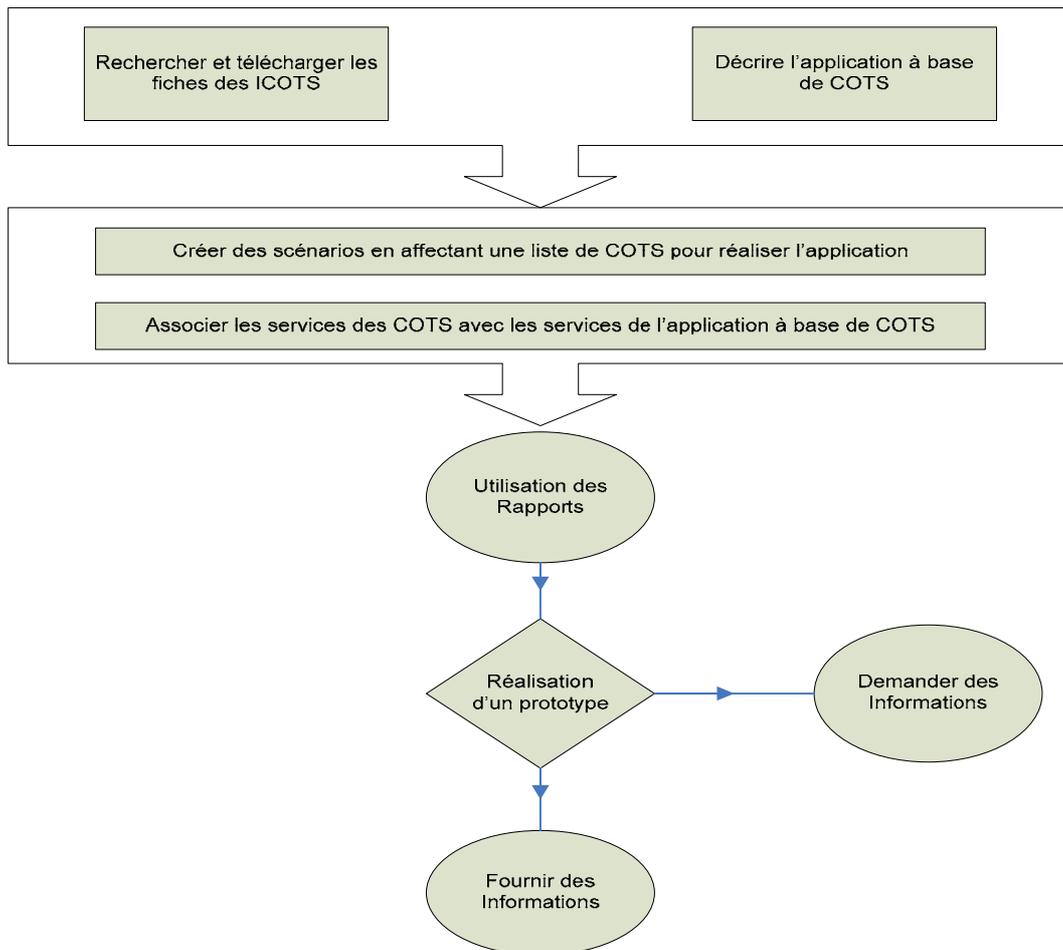


Figure 62 : Les principales étapes d'utilisation du démonstrateur

Dans les paragraphes suivants, nous allons détailler ces différentes étapes.

II - 2.1 La récupération des cartes ICOTS

La première de ces étapes (Rechercher et télécharger les fiches ICOTS) peut se faire en parallèle de celle que l'on va décrire par la suite (Décrire l'application à base de COTS). La récupération des fiches correspond en fait à une petite partie des informations récupérées dans la phase de veille technologique du processus de développement. Les informations sont celles qui intéressent l'intégration des COTS. On y trouve donc les différents environnements déjà cités et les services de chaque COTS.

La recherche s'effectue via Internet. Nous avons prévu que les fournisseurs qui désirent proposer ces ICOTS permettent le téléchargement des fiches utiles aux développeurs. Ces fiches peuvent par la suite être mises à jour pour la récupération d'informations complémentaires et notamment celles correspondantes à des enrichissements de la part de la communauté des COTS.

Cette première étape comme la suivante sont inspirés des deux premières phases du processus de développement d'applications à base de COTS. Elles représentent un moyen pour obtenir les informations sur les COTS mais nous avons aussi pensé que c'était un moyen de préparer notre outil à une utilisation dépassant son objectif premier centré sur les développeurs du prototype. C'est pour cela que l'on retrouve cette similitude avec le processus de développement à base de COTS. Le démonstrateur doit pouvoir servir aussi de guide dans la saisie et la manipulation des informations relatives aux autres phases de ce processus.

II - 2.2 La description de l'application à base de COTS

C'est une étape indispensable si l'on désire que le démonstrateur puisse proposer des rapports de synthèse. Elle n'est pas utile si les développeurs se contentent de lire les informations structurées proposées par les fournisseurs de COTS. Ce qui en soi répond à notre objectif de réduire la difficulté d'accès à l'information notamment pour la réalisation des prototypes. Mais le fait de ne pas faire cette étape réduit une partie de l'aide que l'on peut apporter aux développeurs en ne présentant pas ces rapports. En fait, nous estimons que le volume important d'informations et les interdépendances entre différents logiciels peuvent rendre difficile l'analyse de ces informations. C'est pourquoi nous préconisons cette étape afin de pouvoir utiliser les rapports de synthèse qui seront capables traiter la totalité des informations utiles et de détecter rapidement et efficacement certaines informations. Par exemple, la liste complète des logiciels nécessaires à l'utilisation ou à l'intégration du prototype et de ses COTS. C'est aussi la possibilité de détecter au plus vite les difficultés prévisibles dues à des incompatibilités entre logiciels constituant le prototype, ses COTS et les logiciels nécessaires au fonctionnement de l'ensemble.

La description de l'application à base de COTS est là aussi pour identifier les services rendus et l'environnement d'exécution cible comme nous l'avons présenté dans le Chapitre 4 : V - 3.1. La saisie des services du prototype permettra de le relier aux COTS qui doivent être intégrés. Par cette étape que l'on détaillera par la suite, nous allons pouvoir au minimum identifier les services réalisés par chaque COTS et ceux restant à développer. La saisie de l'environnement cible permettra d'identifier si les COTS à intégrer sont susceptibles de fonctionner avec ces environnements. Il permettra d'identifier les environnements communs à l'ensemble des COTS, ceux qui ne sont pas compatibles et ceux qui ne sont pas renseignés dans les fiches. Pour ces derniers, ils inciteront le développeur à se poser la question de la nécessité de réaliser une évaluation de ces environnements.

Tout cela aidera le développeur dans la réalisation des prototypes notamment via la lecture des rapports. Mais aussi il est possible que les informations récoltées à l'aide de l'outil et en particulier au sein de la phase d'évaluation, intéresseront les autres phases du

processus de développement : la phase de sélection en identifiant les services fait par les COTS et ceux restant à faire, la phase d'évaluation en identifiant des expérimentations à faire avec le prototype comme l'exemple identifié ci-dessus : les environnements non renseignés dans les fiches ICOTS.

II - 2.3 La description des scénarios

Cette étape n'est pas indispensable à la validation de notre travail. Elle nous semble cependant intéressante dans le cadre de l'utilisation du démonstrateur pour la sélection et l'évaluation des COTS. Elle permet de décrire plusieurs variantes pour l'intégration des COTS au niveau de l'environnement d'exécution ou/et de l'intégration mais aussi sur la combinaison de COTS à utiliser dans l'application à développer. En fait, on doit pouvoir utiliser ce démonstrateur pour décrire l'application dans son ensemble et non seulement le prototype et quelques COTS définis par la phase d'évaluation. Si dans cette phase on peut avoir besoin d'évaluer les COTS les uns indépendamment des autres, dans le cas d'une vision générale de l'application, il nous semble intéressant de pouvoir décrire la totalité de l'application et des COTS la constituant. Pour des besoins de comparaison de scénarios, la possibilité de les décrire permettrait sûrement d'identifier les différences et donc de détecter des informations utiles pour ces phases du processus de développement.

Cette étape est surtout une proposition qui n'est pas implémentée dans le démonstrateur mais qui reste accessible au niveau du modèle entité / association que nous proposons.

II - 2.4 L'association des services des COTS avec les services de l'application à base de COTS

Cette étape est indispensable pour l'exploitation des rapports de synthèse que nous présenterons ci-après. C'est à l'aide de l'association entre les services de COTS et les services que celui-ci remplace dans l'application à base de COTS que le démonstrateur identifiera tous les COTS (et chacun de ses services) impliqués dans le développement de l'application à base de COTS. A partir cette association l'ensemble des informations relatives aux environnements d'exécution et d'intégration pourra être pris en compte par notre démonstrateur. Il établira à partir de cette étape une synthèse sur ces environnements. Nous les présentons à l'utilisateur sous forme de deux rapports correspondant à chaque environnement (exécution et intégration).

L'intérêt pour le développeur est de filtrer les informations recueillies pour l'ensemble des logiciels utilisés dans le cadre de la réalisation du prototype. Cela favorise la synthèse et l'accès aux informations que l'on trouve dans les rapports que nous présentons ci-dessous.

II - 2.5 L'utilisation des rapports de synthèse

Les objectifs principaux des rapports de synthèse sont d'identifier :

- les environnements d'exécution qui sont disponibles pour l'application et l'ensemble des COTS qui la composent.
- les moyens nécessaires pour l'intégration,
- des difficultés ou des dysfonctionnements,

Pour chacun de ces objectifs il est défini un rapport spécifique.

II - 2.5.1 Le rapport sur l'environnement d'exécution

Les développeurs utilisent le rapport sur les environnements d'exécution pour déterminer les éléments communs qui permettront à l'application et à l'ensemble de ses COTS, de fonctionner correctement (Figure 63).

Figure 63 : Rapport sur les environnements d'exécution

En plus, le rapport peut mentionner si cet environnement est compatible avec celui (ou ceux) défini(s) pour l'application à base de COTS lors des spécifications et correspondant à l'environnement cible. Dans tous les cas, ce rapport va permettre de lister l'ensemble des produits nécessaires qu'ils soient directement demandés par un COTS appartenant à cette application ou qu'ils soient indispensables aux fonctionnements des COTS par le fait qu'eux-mêmes peuvent avoir besoins d'environnements d'exécution spécifiques. En fait,

par la lecture de toutes les arborescences de dépendances successives le rapport établit la liste complète des logiciels nécessaires (du COTS dans l'application, du COTS utilisé dans le COTS de l'application, etc.).

II - 2.5.2 Le rapport sur les moyens nécessaires à l'intégration

A partir de ce rapport (les moyens nécessaires pour l'intégration) le développeur pourra établir l'environnement qu'il décide d'utiliser pour le développement de son prototype. La Figure 64 ci-dessous est un exemple d'environnement pour l'intégration de « Microsoft Outlook » et « Microsoft Project ».

The screenshot displays a software integration report tool with three tabs: "Environnement d'exécution", "Moyens Necessaires à l'Intégration", and "Difficulté et Dysfonctionnements". The "Moyens Necessaires à l'Intégration" tab is active, showing a tree structure of COTS and development tools.

Liste des COTS

- Project/V. 2003/Française
 - Alternatif
 - PcSoft
 - Autres Possibilité pour Gestion Proj
 - 2003fr
 - Standard
 - Microsoft
 - Integre Gestion Projet
 - Integre planif des tâches
 - 1.1fr
 - ADD2.6
 - Outlook/V. 2003/Française
 - par Microsoft
 - Visual C#
 - Acces aux tâches
 - Acces au contacts
 - 11fr
 - Outlook ObjectLibrary Vers. 11.0, française
 - Data Access Components Vers. 2.6, Fr
 - .Net Frameworks Vers. 1.1, française
 - Project Serveur Vers. 2003, française

Outils de développement

- Visual Studio Vers. 2003, française
 - Gestion de projet dans (Project Vers. 2003, Française)
 - C#
 - code C++ Exemple
 - Gestion des Contacts dans (Outlook Vers. 2003, Française)
 - C#
 - Doc Lire Contacts Outlook.txt
 - Windev Vers. 10, française
 - Gestion de projet dans (Project Vers. 2003, Française)
 - C++

Liste des Services A Intégrer

- Gestion de projet
- Gestion des Contacts

Liste des Services A Développer

- Gestion des utilisateurs
- Gestion de Projet
- Messagerie
- Manipulation Grille GRAI

Informations sur : Gestion des Contacts dans (Outlook Vers. 2003, Française)

On the Project menu, click Add Reference.
 On the COM tab, click Microsoft Outlook 11.0 Object Library, and then click Select.
 In the Add References dialog box, click OK. If you are prompted to generate wrappers for the library that you selected, click Yes.
 In the Code window, replace the default code with the following code:
 Imports System.Reflection

```

Module Module1

    Sub Main()
        ' Create Outlook application.
        Dim oApp As Outlook.Application = New Outlook.Application

        ' Get namespace and Contacts folder reference.
        Dim oNS As Outlook.NameSpace = oApp.GetNamespace("MAPI")
        Dim cContacts As Outlook.MAPIFolder =
        oNS.GetDefaultFolder(Outlook.OLDefaultFolders.olFolderContacts)
  
```

Figure 64 : Rapport sur les moyens d'intégration d'un COTS

Dans ce deuxième rapport, le développeur a la possibilité de se renseigner sur l'environnement, les exemples et la documentation pour l'intégration de chaque service de COTS dans l'application. Pour chaque COTS, le rapport indique les outils qui vont permettre de réaliser une intégration. La liste peut ne pas être exhaustive mais a le mérite de fournir une réponse sur les moyens nécessaires pour le développement. Si le développeur ne trouve pas l'outil désiré (ou des exemples de codes et de la documentation), il peut demander aux fournisseurs ou à la communauté des COTS si l'outil de développement qu'il a prévu d'utiliser permet l'intégration du COTS. Si tel est le cas, cette information pourra être mise à jour dans la fiche de l'ICOTS et ainsi le développeur aura la réponse à sa demande. Celle-ci pourra être vue par d'autres utilisateurs de COTS qui bénéficieront ainsi du complément d'information. Si personne n'avait la réponse, le développeur devra lui-même établir et prouver cette faisabilité. L'issue de cette évaluation pourra être renseignée dans l'ICOTS et diffusée à son tour comme l'ont fait précédemment les « fournisseurs » et la « communauté de COTS ».

Ce rapport est donc essentiel à notre démarche car il va permettre de réunir sous une même interface l'ensemble des informations utiles pour le développeur du prototype. Nous pouvons seulement regretter que ce rapport ne soit pas aussi utilisable pour des phases de sélection. Nous pensons par exemple que les choix effectués par le développeur du prototype peuvent avoir une incidence sur une phase de sélection. Ceci est d'ailleurs une proposition que l'on peut faire pour l'utilisation de notre travail et de ce démonstrateur dans le cadre de l'ensemble des phases du processus de développement.

II - 2.5.3 Le rapport de dysfonctionnement

Dans le même ordre d'idées, l'exemple de rapport de dysfonctionnement de la Figure 65 doit aider le développeur du prototype dans la détection des difficultés possibles mais aussi pour pouvoir guider le travail d'évaluation.

The screenshot shows a web-based report interface with three tabs at the top: 'Environnement d'exécution' (highlighted in yellow), 'Moyens Necessaires à l'Intégration' (highlighted in green), and 'Difficulté et Dysfonctionnements' (highlighted in red). The main content area is orange and contains two sections:

- Synthèses des résultats sur les difficultés et dysfonctionnements**: A white box containing the text: 'Aucun produit Incompatible avec les spécifications', 'Aucune Incompatibilité détectée entre produits', and '1 Produit identifié de type OS comme incompatible'.
- Liste des Produits Non Conformés**: A white box containing a list of non-compliant products:
 - Produits Non Compatibles avec les spécifications
 - Produits Non Compatibles avec d'autres produits du projet
 - Liste de Produits Non Compatibles identifiés dans les cots utilisés
 - Windows 2000 Vers. Service Pack 2, Française

Figure 65 : Rapport sur les difficultés et dysfonctionnements

II - 2.5.4 Bilan

Dans ce dernier rapport et dans l'exemple de notre démonstrateur, nous identifions un élément de dysfonctionnement (Figure 65). Le produit « Windows 2000 SP2 » est incompatible avec l'utilisation d'un autre logiciel : « Ms Project ». L'ensemble de ces rapports doit aider le développeur à maîtriser les multiples informations issues des différents produits qui vont composer son application. Certaines vont l'aider dans la réalisation de l'intégration (exemple du rapport précédent) pendant que d'autres vont l'avertir sur des configurations d'exécution ou de développement ayant déjà causées des problèmes à d'autres utilisateurs de COTS. Il est évident que cela ne peut que favoriser l'utilisation des prototypes pour évaluer, qualifier et estimer les coûts et les risques. Nous pensons d'une part que cela permettra un meilleur choix des COTS car les informations récoltées seront plus importantes, notamment pour la réalisation de ce prototype, et d'autre part son utilisation améliorera les résultats des évaluations et des qualifications.

III Les bénéfices obtenus

A partir de ce démonstrateur il est possible de vérifier les gains obtenus par son utilisation. Tout d'abord, nous améliorons le recueil et la diffusion d'informations ciblées pour les développeurs en charge de la conception des prototypes d'intégration. Ils ont à leur disposition l'ensemble des informations utiles pour l'utilisation des COTS (environnement d'exécution) et pour le développement d'une application (environnement d'intégration) correspondant à notre proposition faite dans la Figure 49 page 128. Cela permet un gain de temps très important par rapport à la situation actuelle mais surtout une meilleure qualité dans le travail de sélection et d'évaluation de la part des développeurs d'applications à base de COTS. Ces bénéfices peuvent être obtenus par la facilité de récupération des informations reliées à la carte d'identité. Surtout il est possible, pour l'ensemble des utilisateurs de COTS ou des fournisseurs de COTS, de compléter la carte d'identité avec l'historique d'autres projets d'intégration. Ce rôle d'enrichissement des fiches ICOTS nous semble particulièrement intéressant pour les futures réalisations de prototypes. Toutes ces informations peuvent être reliées à chaque service des COTS ce qui améliore encore l'accès à la connaissance.

Un autre volet de l'utilisation de ce démonstrateur permet lui aussi d'obtenir des gains : l'utilisation des rapports de synthèse. A travers eux, le développeur obtient une vue synthétique de certaines difficultés rencontrées pendant les phases de développement du prototype et de l'intégration des COTS. En effet, la détection de conflit entre COTS incompatibles entre eux est une source particulièrement intéressante. Le fait de connaître

cette information avant même de lancer le développement du prototype ou l'évaluation du COTS fera gagner un temps précieux qui sera utilisé pour améliorer la qualité et la richesse des informations.

Pour cela, le démonstrateur permet la mise en relation des utilisateurs de COTS avec la communauté ou les fournisseurs de COTS. Ces derniers pourront les aider en proposant des réponses à leurs questions. Celles-ci seront enregistrées dans le SI et reliées à la carte d'identité du COTS. Le démonstrateur ne propose pas pour l'instant une certification des informations. Pourtant elle permettrait d'enrichir les données de la carte d'identité et surtout améliorerait la qualité et la richesse des informations que les utilisateurs de COTS vont recueillir.

IV Les limites et perspectives du démonstrateur

La limite principale de notre démarche et de ce démonstrateur sont pour l'essentiel liées au périmètre d'étude que nous avons retenu. On a essentiellement apporté de l'aide dans la réalisation des prototypes servant au recueil d'informations pour l'évaluation et la sélection des COTS. Les informations manipulées dans notre démarche (et dans le démonstrateur permettant de la valider) sont parfois communes à celles utiles à ces autres phases du processus de développement d'application à base de COTS. Nous avons présenté plusieurs familles de données dans le Chapitre 3 : Les informations pour l'utilisation des COTS (données produit, spécification, évaluation, estimation). Dans certains cas, comme pour le rapport de dysfonctionnement, nous pouvons imaginer que ces informations pourraient être traitées directement au sein de la phase de sélection sans avoir à réaliser un prototype. Pourtant, du fait du périmètre de notre étude, notre proposition n'intègre pas cette dimension. Nous en avons tenu compte pour que notre travail puisse dans l'avenir servir de support pour ajouter les informations dédiées à chaque phase du processus de développement et les traitements appropriés sous forme de rapports favorisant leur analyse.

Nous pensons que la frontière est souvent floue entre aider la réalisation d'un prototype et le travail de sélection d'un COTS. Pourtant si les informations que nous proposons peuvent servir directement cette phase, nous avons omis un nombre non négligeable d'autres données relatives à ce travail. Par exemple, la sélection doit tenir compte des « données de spécifications » dont nous avons parlé dans le Chapitre 3 : Les informations pour l'utilisation des COTS. Or ces dernières ne peuvent être saisies dans notre démonstrateur. Pourtant, si notre proposition est incomplète à ce niveau là, la carte

d'identité de COTS permettra d'ajouter ces informations et d'améliorer le démonstrateur pour qu'il puisse aussi viser d'autres objectifs.

Enfin, dans la démarche, il est possible de demander des informations complémentaires ou d'en fournir. L'un des intérêts de ce Système d'Information, c'est d'établir cette relation indispensable entre clients et fournisseurs comme nous l'avons identifié dès le début de l'état de l'art. C'est surtout le moyen d'enrichir avec les réponses, les différentes fiches des COTS. Cela servira forcément à d'autres utilisateurs lorsqu'ils récupéreront les cartes d'identité de COTS. Cependant, si nous avons identifié le besoin de valider ces informations complémentaires dans le « Chapitre 4 : IV - 4 La nécessité de certifier les informations ». Cette étape n'a pas été réalisée dans le démonstrateur car elle est surtout une information vitale pour la phase d'évaluation qui doit définir la validité des informations recueillis. C'est elle aussi qui fait le choix des informations à utiliser pour la réalisation du prototype. Nous pensons qu'il est donc important dans le cas où le démonstrateur est utilisé aussi pour la phase d'évaluation (même partiellement) de traiter le problème de la certification des données pour ne pas remettre en cause les résultats de ce démonstrateur.

Chapitre 6 : Conclusion

L'utilisation des COTS dans les développements d'applications informatiques doit permettre d'améliorer les temps de conception, la fiabilité, la qualité et les coûts des applications. Pourtant, les difficultés de mise en œuvre peuvent mettre en doute l'intérêt même de l'utilisation de ces COTS. Les efforts nécessaires pour leurs intégrations sont parfois la cause d'une dérive des coûts et des temps de développement, a contrario de ce qui était escompté. En fait, les développeurs veulent éviter une dérive qui remettrait en cause l'utilisation même de ces COTS. Des modèles de processus comme le dernier de Boehm s'attachent à établir l'évaluation et le choix des COTS à intégrer. Ce type de modèle fait apparaître une spirale qui à chaque tour affine l'évaluation mais en contrepartie incrémente le temps et le coût de l'étude. C'est d'ailleurs dans cette phase de sélection que l'on situe la plus grande difficulté de l'utilisation des COTS. Le choix des produits doit être en accord avec des critères de sélection arrêtés dans une phase de définition des besoins et avec des évaluations des risques et des coûts les moins éloignés de la réalité du futur développement. A cet égard, l'état de l'art de ce mémoire a permis de recenser de nombreuses méthodes qui traitent plus particulièrement de cette phase de sélection : PORE, OTSO, CDSEM, CISD, CAP, CRE, CEP, CBA, STACE, PECA. Tous ces raisons majeurs nous ont amenés à préconiser l'utilisation de prototypes pour rechercher des informations indispensables à toutes ces démarches ou méthodes de sélection que l'on vient d'énoncer. D'après beaucoup d'auteurs, c'est le seul moyen de s'assurer du recueil, de la validité des informations et de minimiser les risques et les coûts. La pauvreté et la faible disponibilité des informations sur les COTS sont les principaux problèmes à la bonne utilisation de ces méthodes de sélection et d'évaluation. Nous avons dans cet esprit montré tout au long de ce mémoire la difficulté de traiter le problème du fait du volume très important des informations à manipuler et de leurs interdépendances. C'est en partant du constat que l'utilisation d'un prototype est un bon tremplin pour recueillir les informations nécessaires aux évaluations mais aussi pour réduire les risques et les coûts, que nous avons décidé d'aider le développeur à le concevoir. A travers notre proposition, nous facilitons le recueil, la capitalisation et le traitement des informations, notamment

dans le cadre de la réalisation d'un prototype permettant l'évaluation de l'intégration de plusieurs COTS.

La proposition est articulée autour de l'utilisation d'un Système d'Information pour COTS. Son rôle est de permettre la mise en relation des clients et des fournisseurs de COTS ainsi que de la communauté des utilisateurs de COTS. A partir de là, nous favorisons l'échange et le recueil d'informations. Elles sont toutes basées sur l'utilisation de notre carte d'identité de COTS, permettant d'identifier tous les produits logiciels de manière unique. C'est un ensemble d'informations qui permet d'identifier le produit par son nom, son concepteur et les services principaux qu'il fournit. Nous l'utilisons pour associer les informations utiles pour le développement du prototype. Toutes ces données sont inscrites dans le SI. Nous y trouvons l'environnement d'exécution de chaque produit. Il est composé des différents logiciels indispensables à son bon fonctionnement que nous renseignons par l'intermédiaire d'une référence à leurs cartes d'identité. Dans chacune d'entre elles nous pouvons aussi trouver les produits nécessaires à leur fonctionnement. Avec ce principe, on est assuré d'avoir la liste complète des dépendances et donc des logiciels nécessaires au fonctionnement de chaque logiciel inscrit dans le Système d'Information. Nous complétons les cartes d'identité avec un deuxième environnement dédié à l'intégration. Cet environnement doit permettre d'identifier les moyens à mettre en œuvre pour réaliser l'intégration du COTS, moyens comprenant un outil de développement, un langage, de la documentation et des logiciels permettant le réaliser l'intégration du COTS. Dans le SI, ils sont présentés sous forme de scénarios afin de proposer plusieurs variantes possibles à l'intégration d'un COTS. C'est donc au développeur de sélectionner l'environnement de développement qui correspond le mieux à son besoin. Ces deux environnements sont complétés avec des scénarios de dysfonctionnement. Ils permettent pour chacun d'entre eux d'identifier des configurations ne permettant pas l'utilisation ou l'intégration d'un COTS.

A l'aide de notre Système d'Information, nous favorisons l'échange et la capitalisation d'informations mais nous aidons aussi le développeur dans la réalisation de son prototype. La relation entre le fournisseur de COTS et le développeur d'applications à base de COTS s'établit à travers l'utilisation des cartes d'identité. Elles contiennent les informations diffusées par le premier et utilisées par le second. Ce dernier peut demander que la carte d'identité soit complétée avec des données supplémentaires dont le développeur aurait besoin pour la sélection, l'évaluation des COTS ainsi que pour la réalisation du prototype. Par la même, la carte d'identité sert à capitaliser ces informations. Tous les compléments d'informations seront utilisés ultérieurement. Ils faciliteront l'utilisation des COTS par rapport à la situation présente car les informations sur les COTS seront plus riches qu'actuellement. Cela permettra un gain de temps mais surtout une meilleure analyse des COTS. Il y aura inévitablement un gain sur la qualité des sélections, et la réalisation des prototypes sera facilitée. Cette richesse d'information est complétée par le renseignement des scénarios de dysfonctionnement au niveau des environnements d'exécution et

d'intégration des COTS. Ce sont des données supplémentaires qui permettent de détecter au plus vite certaines difficultés qui n'apparaissent aujourd'hui qu'avec l'utilisation des prototypes. C'est donc là aussi un gain de temps et une précision sur l'analyse des COTS qui ne peut entraîner qu'une amélioration générale de l'utilisation des COTS dans le développement d'applications.

L'utilisation de notre Système d'Information est centrée principalement sur l'aide au développement du prototype permettant l'analyse des COTS. Les informations manipulées sont actuellement recueillies tout au long du processus de développement d'une application à base de COTS. Cela démarre par la définition des besoins qui permet d'inscrire dans le SI, la carte d'identité de l'application à développer. Cela se poursuit lors des phases de veille technologique et d'acquisition des COTS. Dans ce cas, c'est l'utilisation des différentes cartes d'identité des COTS proposées par les fournisseurs qui contribuent à la réalisation du processus de développement en apportant son lot d'informations. C'est aussi dans les phases importantes de choix et de décision avec les évaluations et la qualification des COTS que le SI apporte une aide réelle aux développeurs. Il permet la détection précoce des conflits à l'aide des rapports de dysfonctionnement, il établit les informations permettant de réaliser l'intégration avec les scénarios du même nom et enfin il permet d'établir l'environnement d'exécution de l'application finale en intégrant tous les COTS.

Globalement, notre proposition est centrée sur l'aide à la réalisation d'un prototype intégrant un ou plusieurs COTS. Même si notre démonstrateur est en mesure de détecter des dysfonctionnements qui pourraient remettre en cause une sélection de COTS ou une composition d'un ensemble de COTS, l'étude ne porte pas sur les informations nécessaires aux autres phases du processus de développement, notamment dans le cadre de la sélection. En effet, lorsque le développeur réalise son prototype la sélection des COTS est déjà faite. Au même titre, si dans l'état de l'art, nous avons détecté des informations utiles en fonction des différentes phases du processus de développement, celles-ci ne sont pas implémentées dans le Système d'Information car elles n'étaient pas utiles directement pour la réalisation des prototypes. De la même manière, nous avons abordé la problématique de la validité de l'information, notamment celle fournie par la communauté des COTS, et nous avons proposé de le traiter par le biais de l'identification du créateur de l'information. Cependant, le démonstrateur, aujourd'hui ne permet pas de traiter cette partie car d'une part elle ne remet pas en cause notre proposition et d'autre part, nous pensons que la mise en œuvre de cette partie dépend de la standardisation qu'il est nécessaire de faire sur l'ensemble de notre Système d'Information. Pour cela, il faudra étudier les standards existant sur les formats et les échanges de données. Il sera nécessaire de trouver une solution adaptée à notre proposition et qui en même temps soit acceptée par le plus grand nombre d'utilisateurs de notre SI.

Tout cela reste possible si en plus, nous arrivons à obtenir les fiches ICOTS de la part de tous les fournisseurs de COTS. Pour ce faire, nous préconisons au minimum deux conditions à respecter :

- la possibilité que chaque fournisseur de COTS reçoive son identifiant unique permettant de le dissocier d'un autre fournisseur de COTS,
- la possibilité que le format des fiches ICOTS soit standardisé pour favoriser l'échange de données préconisé dans notre travail.

Dans le premier cas, un identifiant unique pour chaque fournisseur de COTS existe : il suffirait d'utiliser le numéro d'identification international des entreprises. Mais encore faut-il décider que cet identifiant est bien celui représentant chaque fournisseur de COTS. Il permettrait aussi l'identification des auteurs des modifications des fiches ICOTS. Dans le deuxième cas leur format peut être celui de « XML » par exemple, car c'est un format d'échange standard sur Internet. Cependant, il doit être accepté et utilisé de manière unique par tous les acteurs que nous avons définis dans notre système d'information. Un besoin de spécification et de normalisation des fiches manipulées par notre démonstrateur semble indispensable à l'utilisation de notre approche. Cette standardisation doit permettre l'évolution de notre proposition vers une utilisation tout au long du processus de développement et dans le cadre des différentes difficultés d'obtention et de manipulation des informations associées à chaque COTS.

La facilité avec laquelle on peut réunir les informations recueillies à travers les fiches d'ICOTS fait penser que l'on pourrait utiliser le système dans toutes les étapes du processus de développement à base de COTS. En particulier, il est possible d'établir les processus qui permettront de relier les activités et les expertises de chacune de ces phases. Nous pensons pour cela aux « workflow » qui permettent de gérer les flux d'informations au sein d'une organisation. Dans notre SI, les flux d'informations correspondent aux procédures de saisies, à l'enchaînement des phases du processus de développement en transmettant les informations précédemment saisies et aux traitements de ces informations issues de chaque phase du processus. Les informations manipulées sont principalement les fiches ICOTS de l'application à base de COTS que le développeur doit réaliser et les informations qu'il doit recueillir dans chaque phase du processus.

Enfin nous pourrions modéliser l'organisation du Système d'Information en fonction des différents processus de développement à base de COTS existants. Le fait d'ajouter à notre travail une dimension plus large incluant les différentes méthodes d'évaluation et de sélections doit permettre de contrôler le déroulement de la méthode sélectionnée qui garantit au développeur un choix optimal des COTS et une maîtrise des coûts et des risques. La modélisation de l'organisation dans le SI ajoutera de la rigueur et une aide dans le pilotage des méthodes de sélection ou d'évaluation implémentées.

Dans le cadre d'un pilotage du processus à l'aide du SI nous pensons pouvoir traiter les interconnexions entre différentes informations en provenance de différentes phases du processus de développement de l'application à base de COTS. Aujourd'hui, il est admis que les résultats d'une phase d'évaluation ont une incidence sur la phase de sélection et

vice-versa. Les cycles d'itération décrits dans différentes méthodes de sélection ont à notre avis le défaut de ne pas traiter l'interdépendance des informations de chaque phase de l'itération. Il en ressort qu'il est nécessaire d'attendre la fin de l'itération pour prendre une décision. Dans le cas où le résultat d'une évaluation aurait pour conséquence le changement de COTS à évaluer, la détection précoce de l'information qui déclenche l'arrêt de l'itération me semble particulièrement intéressante. Nous pensons que cela permettrait de réduire le temps de traitement du cycle de la procédure de sélection / évaluation. Le temps de traitement était une des contraintes ou une des critiques à l'encontre de l'utilisation des COTS dans les développements de logiciels. Cette étude sur l'interdépendance des résultats pour chaque phase du processus de développement devrait amener à notre avis un gain de temps et une qualité de la sélection des COTS.

Références

- [Abts97] Chris Abts, Boehm ; COTS/NDI Software Integration Cost Estimation & USC-CSE COTS Integration Cost Calculator V2.0 ; User Guide Rev1.0 ; 30 September 1997.
- [Abts00] Chris Abts, Boehm, Clark ; COCOTS: A COTS Software Integration Lifecycle Cost Model - Model Overview and Preliminary Data Collection Findings.
- [Albert02] Cecilia Albert, Lisa Brownsword ; Evolutionary Process for Integrating COTS-Based Systems (EPIC) Building, Fielding, and Supporting Commercial-off-the-Shelf (COTS) Based Solutions ; TECHNICAL REPORT, CMU/SEI-2002-TR-005, ESC-TR-2002-005 ; November 2002.
- [Alves01] Alves, Castro ; “CRE: A Systematic Method for COTS Components Selection” ; XV Brazilian Symposium on Software Engineering(SBES) Rio de Janeiro ; Brazil ; October 2001.
- [Bertoa03] Bertoa, M.F., Troya, J., and Vallecilo, A ; “A Survey on the Quality Information Provided by Software Component Vendors.” ; In Proc. Of the 7th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2003) ; Darmstadt ; Germany ; July 2003.
- [Basili01] V.Basili, B.Boehm ; “COTS-Based Systems Top 10 List” ; IEEE Computer 34(5) ; pp 91-93 ; May 2001.
- [Boehm81] Cocomo, 81,Boehm, Barry W. ; *Software Engineering Economic* ; Prentice Hall ; 1981.
- [Boehm86] Boehm, B. ; “A Spiral Model of Software Development and Enhancement” ; *Computer 21* ; pp 61-72 ; 5 May 1988.
- [Boehm89] Boehm, B.W. ; “Software Risk Management” ; IEEE Computer Society Press ; ISBN 0818689064 ; 1989.

-
- [Boehm97] Boehm, Abts ; "COTS Software integration, Cost Modeling Study" ; USC-CSE ; tech. Report 98-250 ; 1997.
- [Boehm02] Barry Boehm, Dan Port, Ye Yang, Jesal Bhuta, Chris Abts ; "Composable Process Elements for Developing COTS-Based Applications" ; 2002
- [Brownsword00] L.Brownsword, T.Oberndorf, C.Sledge. "Developing New Processes for COTS-Based Systems" ; IEEE Software ; pp. 48-55 ; July/August 2000.
- [Botella02] Pere Botella, Xavier Burgués, Juan P. Carvallo‡, Xavier Franch, Carme Quer ; Using Quality Models for Assessing COTS Selection ; WER pp 263-277 ; 2002.
- [Carney97] Carney D ; Assembling Large Systems from COTS Components : Opportunities, Cautions and Complexities ; SEI Monographs on Use of Commercial Software in Gouvernement Systems ; Software Engineering Institute ; Pittsburg ; USA ; June 1997.
- [Carney03] David J. Carney, Patricia A. Oberndorf, Patrick R.H. Plac ; A Basis for an Assembly Process for COTS-Based Systems (APCS) ; Technical Report ; CMU/SEI-2003-TR-010, ESC-TR-2003-010 ; *May 2003*.
- [Comella02] S. Comella-Dorda, J. C. Dean, E. Morris, and P. Oberndorf ; "A process for COTS Software Product Evaluation" ; Proceedings of ICCBSS ; pp 86-92 ; Orlando ; Florida ; USA ; February 2002.
- [Dean00] Dean, J. Vidger, M. ; "COTS Software Evaluation Techniques" ; Proceedings of The NATO Information Systems Technology ; Symposium on Commercial Off-the-shelf Products in Defence Applications ; Brussels ; Belgium ; April 2000.
- [ESI01] ESI; Introduction to cost estimation models for software projects (cocomo, cocomo II & cocots) ; ESI-2001-CODE/V1 ; July 2001.
- [FAR96] Federal Acquisition regulations ; Washington, DC ; General Services Administration; 1996.
- [Heineman01] G. Heineman, W. Council ; Component-Based Software Engineering: Putting the Pieces Together ; Addison-Wesley ; 2001.
- [ISO9126] ISO9126 ; Qualité logicielle; <http://lil.univ-littoral.fr/%7Eoumoumsack/qualite/ISO9126.html>.
- [ISO99] ISO/IEC-15408 (1999) ; Common Criteria for Information Technology Security Evaluation, v 2.1 ; Nat'l Inst. Standards and Technology ; Washington, DC ; August 1999 ; <http://csrc.nist.gov/cc>.
- [JeanRenaud94] JeanRenaud and P. Romanazzi ; "Software Product Evaluation: A Methodological Approach" ; In Software Quality Management II:Building Software into Quality ; pp59-69 ; 1994.

-
- [Kontio95] J. Kontio ; “OTSO: A Systematic Process for Reusable Software Component Selection” ; CSTR-3478 ; University of Maryland Technical Reports. University of Maryland; College Park, MD ; 1995.
- [Kunda99] D. Kunda and L. Brooks; “Applying Social-Technique approach for COTS selection” ; Proceedings of 4th UKAIS Conference ; University of York ; McGraw Hill ; April 1999.
- [Lemoigne77] J.J. Lemoigne ; *La Théorie du système général* ; Théorie de la modélisation ; Presses Universitaires de France ; 1977.
- [Leung02] Karl R.P.H. Leung, Hareton K.N. Leung ; On the efficiency of domain-based COTS product selection method ; Information and Software Technology pp 703-715; 44 (2002).
- [Madachy97] Raymond J. Madachy ; Heuristic Risk Assessment Using Cost Factors; IEEE Software; p51-59; May, June 1997.
- [Mével03] Mével Jean Pierre ; dictionnaire atlas ; Hachette ; ISBN : 2012805531 - EAN : 9782012805538 ; November 2003.
- [Michel05] Michel R., Roose P., Barbier F. ; « Information System for Evaluation of COTS »; 3rd International Conference on Software Engineering Research; Management and Application (SERA2005); Central Michigan University ; Mount. Pleasant ; Michigan ; USA ; Published by the IEEE Computer Society ; Août 2005.
- [Morisio00] Maurizio Morisio, Nancy Sunderhaft ; Commercial-Off-The-Shelf (COTS): A Survey A DACS State-of-the-Art Report ; December 2000.
- [Morisio02] M. Morisio, C.B. Seaman, V.R. Basili, A.T. Parra, S.E. Kraft, S.E. Condon ; COTS-based software development: Processes and open issues ; The Journal of Systems and Software ; pp 189–199 ; 61 (2002).
- [NCube99] NCube, Maiden, N.A.M. (1999) ; “PORE: Procurement-Oriented Requirements Engineering Methode for the Component Based Systems Engineering Development Paradigm”; International Workshop on Component-Based Software Engineering ; 1999.
- [Obendorf97] T.Oberndorf. ; “COTS and Open Systems - An Overview” ; 1997, available at <http://www.sei.cmu.edu/str/descriptions/cots.html#ndi>.
- [Obendorf00] Tricia Oberndorf, Lisa Brownsword, Carol A. Sledge, PhD ; An Activity Framework for COTS-Based Systems ; Technical Report CMU/SEI-2000-TR-010 ESC-TR-2000-010 ; October 2000.

-
- [Ochs01] Ochs, M.; Pfahl, D.; Chrobok-Diening, G.; Nothelfer-Kolb, B. ; "A Method for Efficient Measurement-based COTS Assessment and Selection - Method Description and Evaluation Results" ; Pro. of the Seventh International Software Metrics Symposium METRICS 2001, London, pp. 285-297 ; April 2001.
- [Phillips02] B.C. Phillips and S. M. Polen ; "Add Decision Analysis to Your COTS Selection Process" ; Software Technology Support Center Crosstalk ; April 2002.
- [Port04] Dan Port, Zhihao Chen ; "Assessing COTS Assessment: How Much is Enough?" ; University of Southern California ; Los Angeles ; CA; ICCBSS 2004.
- [STSC03] "Software Risk Management: Back to Basics – The Top 10 (or so) Software Risks" ; Software Technology Conference ; 2003; <http://www.stc-online.org/stc2003proceedings/PDFFiles/pres1483.pdf>
- [Torchiano02] Marco Torchiano, Letizia Jaccheri, Carl-Fredrik Sørensen, and Alf Inge Wang ; COTS Products Characterization ; Department of Computer and Information Science (IDI) ; Norwegian University of Science and Technology (NTNU) ; Copyright 2002 ACM 1-58113-556-4/02/0700 ; SEKE '02 ; Ischia; Italy ; July 15-19; 2002.
- [Tran97] Vu Tran and Dar-Biau Liu. ; "A Risk-Mitigating Model for the Development of Reliable and Maintainable Large-Scale Commercial-Off-The-Shelf Integrated Software Systems" ; in Proceedings of the 1997 Annual Reliability and Maintainability Symposium ; pp361-67 ; Jan 1997.
- [Vigder96]: M.Vigder, M.Gentleman, J.Dean. ; "COTS Software Integration: State of the Art" ; Technical Report NRC No. 39190 ; 1996.
- [Vidger97]: M.Vidger, J.Dean. ; "An Architectural Approach to Building Systems from COTS Software Components" ; In Proceedings of the 1997 Center for Advanced Studies Conference (CASCON 97) ; Toronto ; Ontario ; 10-13 November ; 1997 ; available at <http://seg.iit.nrc.ca/English/abstracts/NRC40221abs.html>.

Annexe

I Exemple d'utilisation du démonstrateur

I - 1 Présentation

Nous allons aborder dans cette section l'utilisation du démonstrateur dans le cadre d'un exemple de développement d'une application à base de COTS qui se nomme IPPOP. Celle-ci aura besoin d'intégrer deux COTS : Ms Outlook et Ms Project. Comme nous l'avons vu dans ce mémoire la phase de sélection des COTS est déjà faite et le développeur doit évaluer les produits sélectionnés. Pour cela il réalise un prototype qui intègre les différents services des deux COTS nécessaires pour l'application IPPOP. Nous décrirons l'utilisation du démonstrateur à travers deux utilisateurs de notre Système d'Information : Le fournisseur de COTS « Microsoft » qui va créer les fiches ICOTS et l'utilisateur des COTS qui réalise l'application IPPOP et s'aidera du démonstrateur pour développer le prototype.

Dans le premier cas, avec le fournisseur nous décrirons la saisie des fiches ICOTS. C'est ainsi que l'on découvrira l'implémentation des environnements d'exécution et d'intégration proposés dans ce mémoire. Par la suite le deuxième acteur déroulera la démarche d'utilisation du démonstrateur. A travers celle-ci nous présenterons les trois rapports proposés par dans ce mémoire. Nous récupérerons aussi les informations utiles pour réaliser l'intégration de chaque service des COTS.

Le démonstrateur se présente sous une forme identique quelque soit l'utilisateur identifié (Fournisseur de COTS, utilisateur de COTS ou communauté de COTS). L'utilisateur de COTS est toujours identifié comme « Perso_RM ».

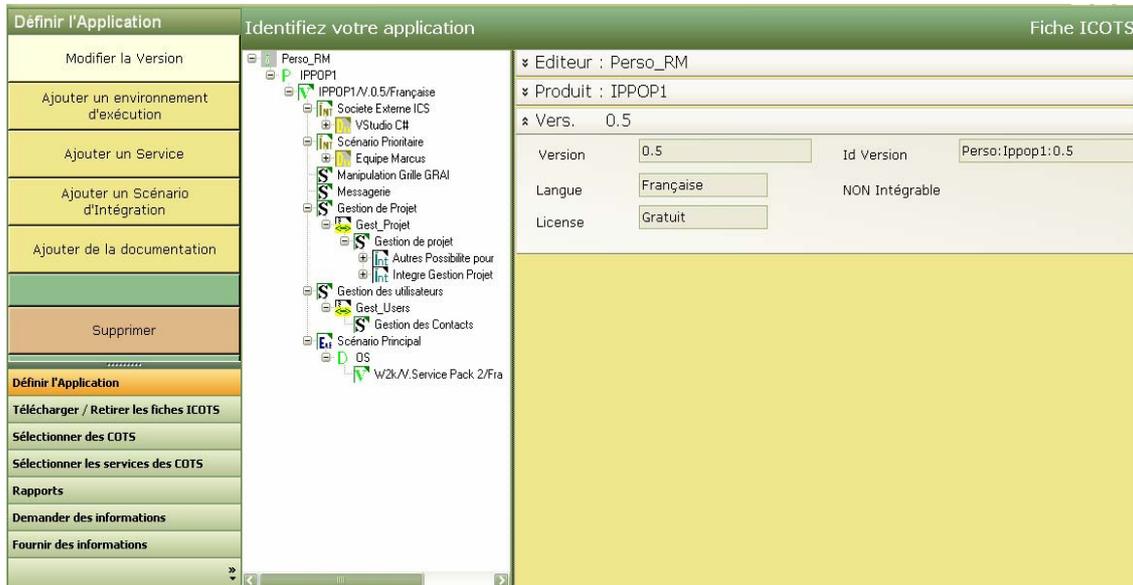


Figure 66 : Ecran principal du démonstrateur

A partir de cet écran l'utilisateur accède aux fonctionnalités principales du démonstrateur qui se situe en bas et à gauche de l'IHM :

- Définir l'application,
- Télécharger / Retirer les fiches ICOTS,
- Sélectionner des COTS,
- Rapport,
- Demander des informations,
- Fournir des informations.

C'est à travers ce menu que nous aborderons la démarche d'utilisation du démonstrateur. Nous retrouvons des actions déjà citées comme la création de l'ICOTS avec le menu « Définir l'Application », la sélection et la recherche de produits pour l'application à développer avec le menu « Télécharger les fiches ICOTS » ainsi que la possibilité d'échanger des informations avec d'autres utilisateurs par l'intermédiaire des menus « Demander des informations » et « Fournir des informations ».

Les informations relatives à chaque produit sont affichées sous forme de fiches dans la partie droite de l'écran. C'est donc à cet endroit que les utilisateurs pourront visionner ou compléter les données relatives aux produits.

Surtout nous avons dans la partie centrale une colonne (Figure 67) décrivant sous forme arborescente les informations relatives aux COTS.

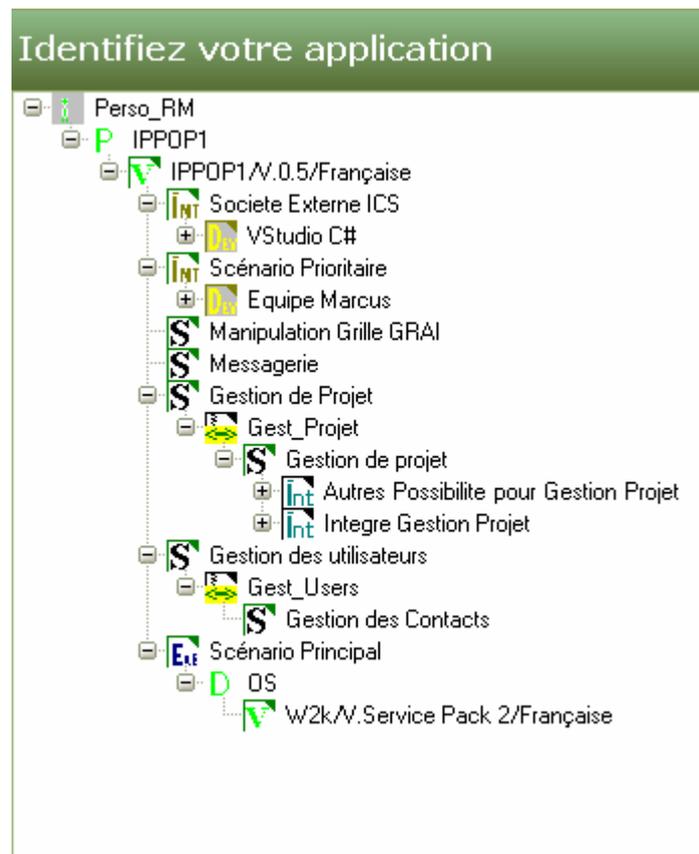


Figure 67 : Arborescence du SI

Les informations sont fournies par défaut à partir des concepteurs de COTS. Pour chacun, il existe une liste de produits qu'il diffuse. Pour chaque produit, nous inscrivons les différentes informations que nous manipulons et qui permettront d'aider le développeur dans sa réalisation d'un prototype. C'est essentiellement la représentation informatique des données relatives à la Figure 47 page 126. On y retrouve les différents environnements d'exécution et d'intégration avec des exemples de codes, de la documentation, des langages et des outils de développement.

	Acteur (Utilisateur, FourCo)
	Produit logiciel
	Version
	Services
	Scénario d'exécution et de dysfonctionnement
	Dépendances
	Scénario d'intégration et de non intégration
	Environnement de développement
	Documentation
	Langage de programmation
	Outil de développement

Figure 68 : Dictionnaire des Icônes

Dans l'exemple de la Figure 67, nous avons développé les informations relatives au produit Outlook de Microsoft avec la signification de chaque icône utilisé dans la Figure 68. On y retrouve un scénario d'exécution demandant l'utilisation d'un OS « Windows 2000 service pack 2 ». On a aussi deux scénarios d'intégration et la liste des services de l'application. Parmi eux deux seront réalisés par intégration de COTS (Gestion de projet et gestion des utilisateurs).

Ci-dessous nous allons décrire les manipulations réalisées par les différents acteurs.

On présentera la création de la fiche ICOTS de MS Outlook. Celle de ms Project sera considérée comme établis avant de passer à la phase suivante de l'utilisation du démonstrateur par l'utilisateur de COTS. Celui-ci décrira son application IPPOP qui intégrera deux COTS Ms Project et MS Outlook. Cela nous permettra de découvrir les informations recueillies et les manipulations à réaliser avec le démonstrateur.

I - 2 Le fournisseur de COTS

I - 2.1 Son rôle

Le fournisseur de COTS propose ses produits et utilise notre Système d'Information pour diffuser les fiches ICOTS. Il a donc besoin de créer ces fiches et de les renseigner avec notamment l'environnement d'exécution et d'intégration. Le premier environnement doit permettre aux utilisateurs de COTS d'identifier l'ensemble des logiciels nécessaires à son fonctionnement. De même dans la fiche ICOTS le fournisseur doit aussi proposer des environnements de dysfonctionnement qui avertiront l'utilisateur d'une non-conformité des COTS avec ces environnements. Dans le deuxième cas (l'environnement d'intégration) le fournisseur doit aider l'utilisateur du COTS à intégrer son produit dans une application. Il fournit ainsi les détails de l'environnement d'intégration décrit dans ce mémoire. Il propose aussi de la documentation pour réaliser cette intégration.

I - 2.2 Créer une fiche ICOTS

Le démonstrateur permet de créer une fiche ICOTS. L'utilisateur du démonstrateur s'identifie comme l'utilisateur « Microsoft ». A partir de cet identifiant le démonstrateur lui propose l'ensemble des fiches lui appartenant sous forme d'une arborescence (Figure 67). En sélectionnant un élément de l'arbre le démonstrateur proposera sur la partie gauche de son interface les menus correspondant à sa sélection. Pour la création de la fiche ICOTS d'Outlook le fournisseur déroule les menus suivant rassemblés dans la Figure 69.

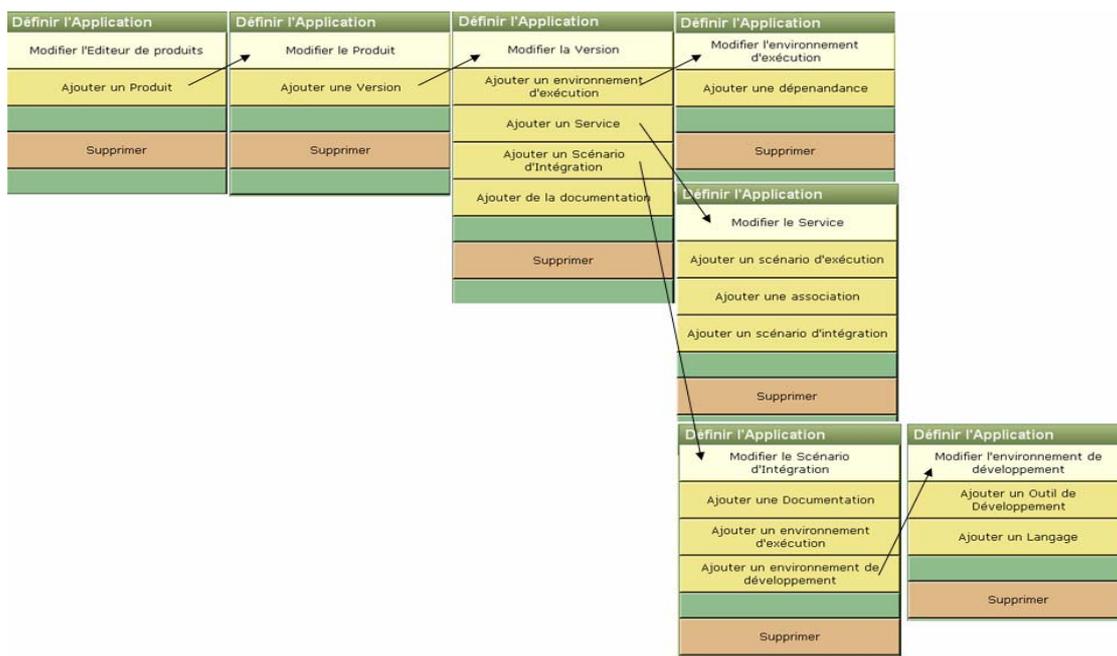


Figure 69 : Les menus du démonstrateur

Le fournisseur de COTS commence donc par identifier son produit par son nom et son identifiant (Figure 70).

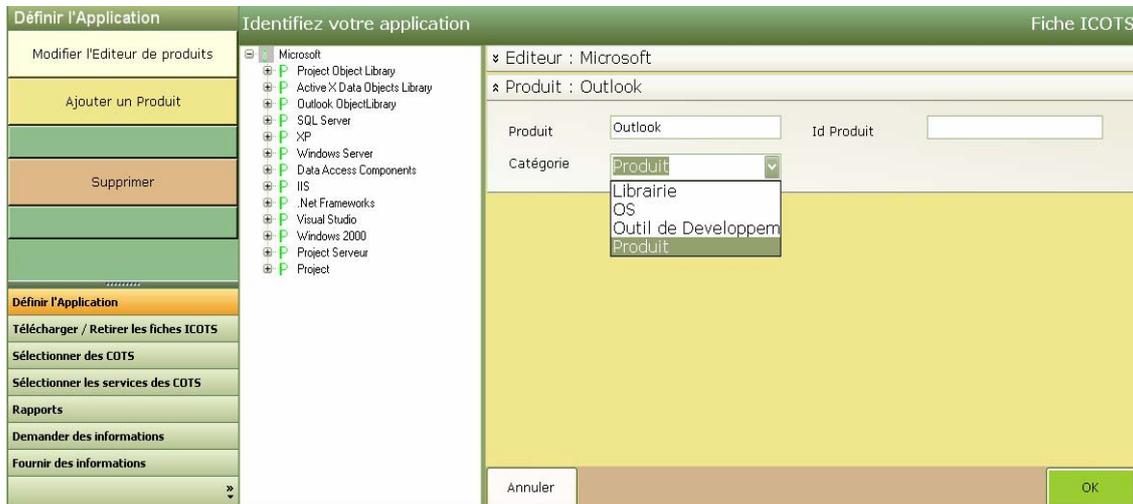


Figure 70 : Création de la fiche produit Outlook

Celui-ci appartient à l'une des quatre catégories suivantes : Librairie, OS, Outil de développement et Produit. Ces catégories serviront à filtrer les logiciels en fonction des besoins du démonstrateur. C'est ainsi que les « outils de développement » servent à filtrer les logiciels pour le choix de l'environnement de développement. Le filtre correspondant aux « L'OS » permet de différencier des produits lors de la présentation des rapports. Enfin, la catégorie « librairie » sert à identifier les logiciels ne pouvant être exécutés seuls.

Puis il identifie la version du produit (Figure 71).

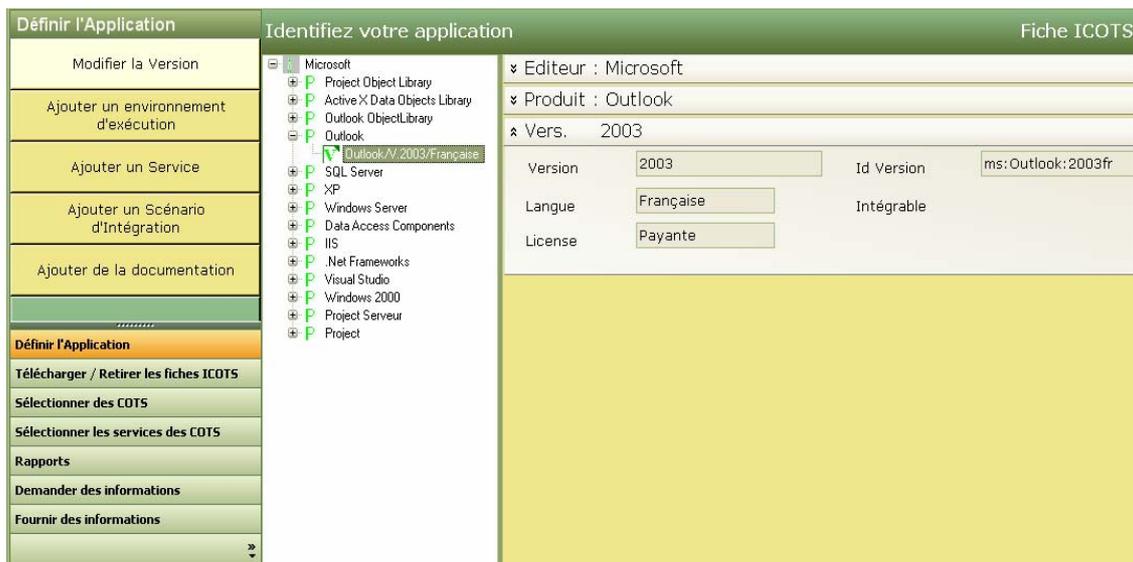


Figure 71 : Créer la version d'un produit

Dans cette fiche nous pouvons ajouter des informations spécifiques à chaque version. Nous avons inséré, dans cet exemple du démonstrateur, trois informations dont une nous semble importante : Le produit est-il intégrable ou pas. Cette information permettra de filtrer les produits qui peuvent fournir des services à une application à base de COTS comme IPPOP. C'est aussi un point important de l'identification du produit comme étant un COTS.

A ce stade quatre possibilités s'offrent à l'utilisateur du démonstrateur : ajouter un environnement d'exécution, ajouter un service, ajouter un scénario d'intégration ou ajouter de la documentation.

Nous présentons donc la première de ces quatre possibilités : l'ajout de l'environnement d'exécution.

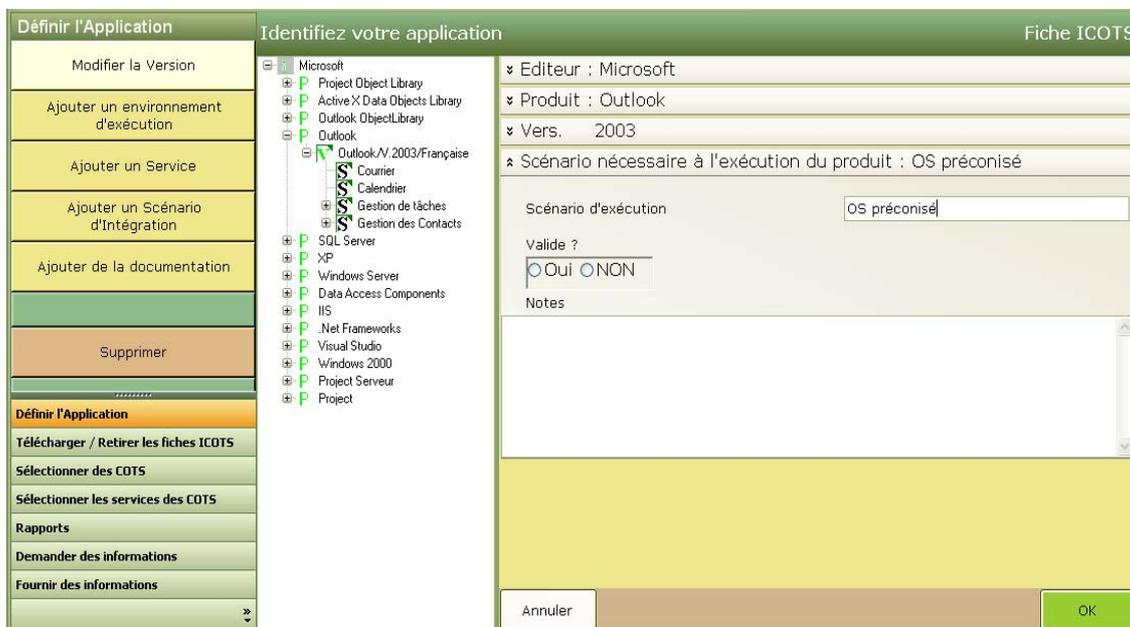


Figure 72 : Ajout de l'environnement d'exécution

Cet environnement permettra d'associer la liste des OS compatibles et des produits indispensables au fonctionnement d'Outlook comme dans l'exemple ci-dessous de l'OS « Windows XP SP2 ». Il permet aussi de décrire de la même manière la liste des logiciels incompatibles avec le COTS. Dans l'exemple d'IPPOP on découvrira que « Ms Outlook » a besoin pour l'intégration de « Outlook Object Library V11 ». Or ce dernier possède un scénario de dysfonctionnement avec un autre produit : « Active X data Object Library V2.6 ». Ce scénario de dysfonctionnement aura une incidence sur la cohabitation de « Ms Outlook » et de « Ms Project » lors de l'intégration. L'inscription de ce scénario de dysfonctionnement se fait au niveau du produit « Active X data Object Library V2.6 » dans un environnement d'exécution qui sera marqué comme non valide.

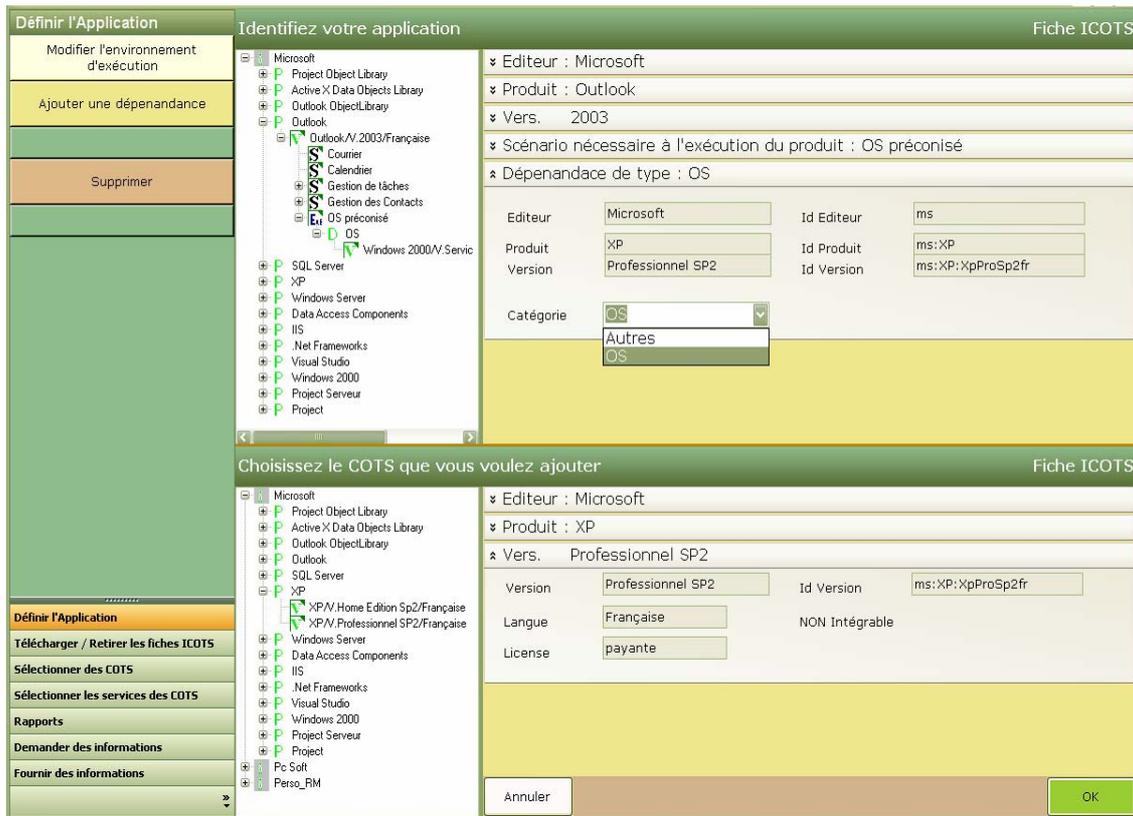


Figure 73 : Ajout d'un OS dans l'environnement d'exécution

Après avoir indiqué la version du produit nous avons une autre information à renseigner : la liste des services de l'application. Dans la Figure 74 on vous présente la description du service « Gestion des Contacts » de « Ms Outlook ».

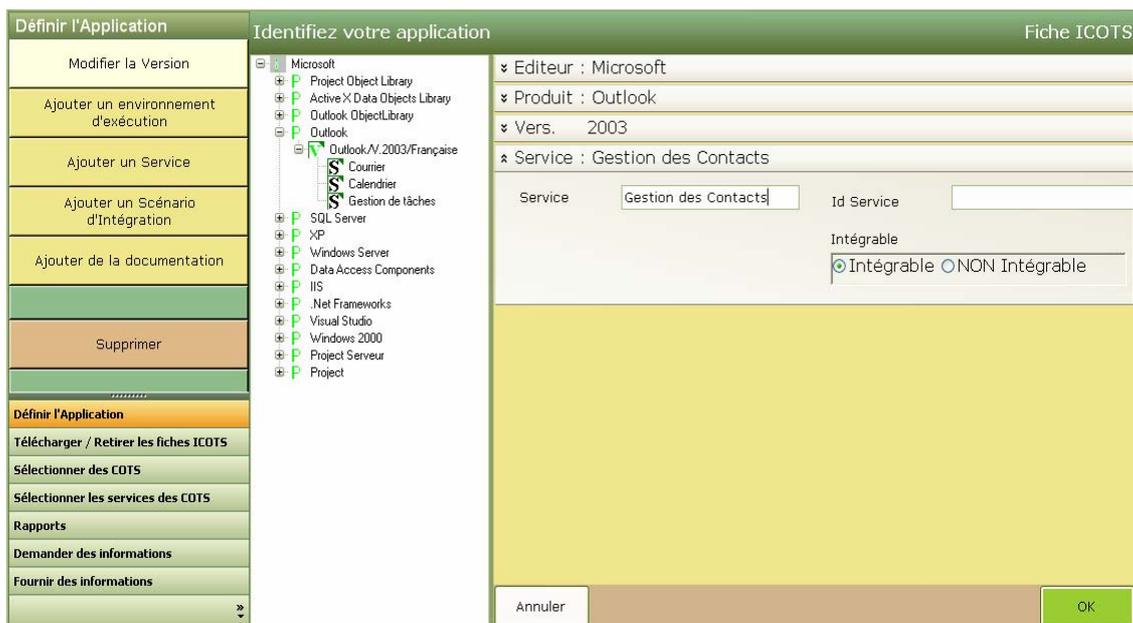


Figure 74 : Ajout de services dans la fiche ICOTS

De la même manière que précédemment nous indiquons pour chaque service si celui-ci est intégrable. L'objectif de cette information est la même que pour la description de la version du produit.

Pour chaque service il est possible maintenant de réaliser deux opérations : ajouter un scénario d'exécution et ajouter un scénario d'intégration.

L'ajout du scénario d'exécution permettra d'associer pour chaque service son propre environnement d'exécution. Il permettra notamment de spécifier des environnements différents en fonction des services utilisés par l'application. Dans le cas de « MS Outlook » cela ne sera pas nécessaire. Nous considérons que le scénario d'exécution de tous les services est par défaut et dans l'absence de toute autre information le premier scénario établi.

L'écran correspondant à cette étape serait donc le suivant (Figure 75).

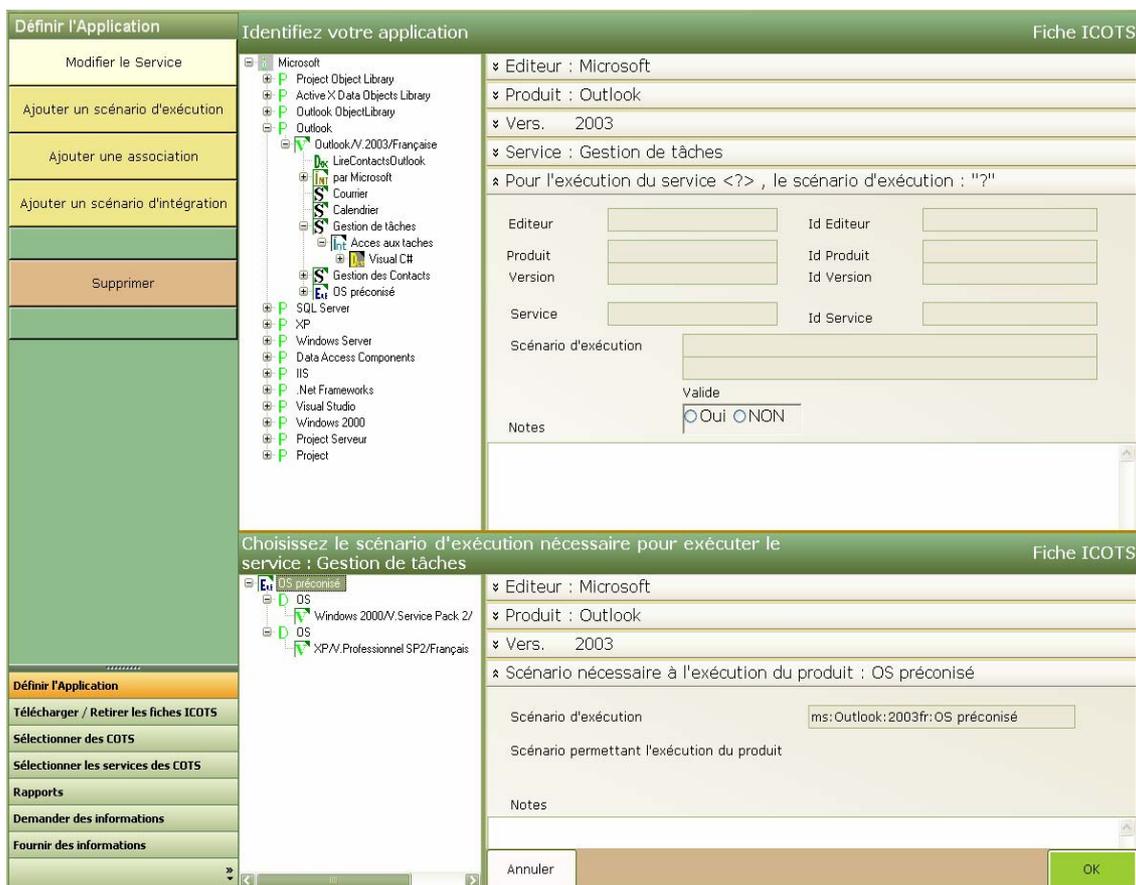


Figure 75 : Ajout d'un scénario d'exécution associé à un service de l'application

Autre possibilité pour le service, l'ajout d'un scénario d'intégration spécifique au service (Figure 76).

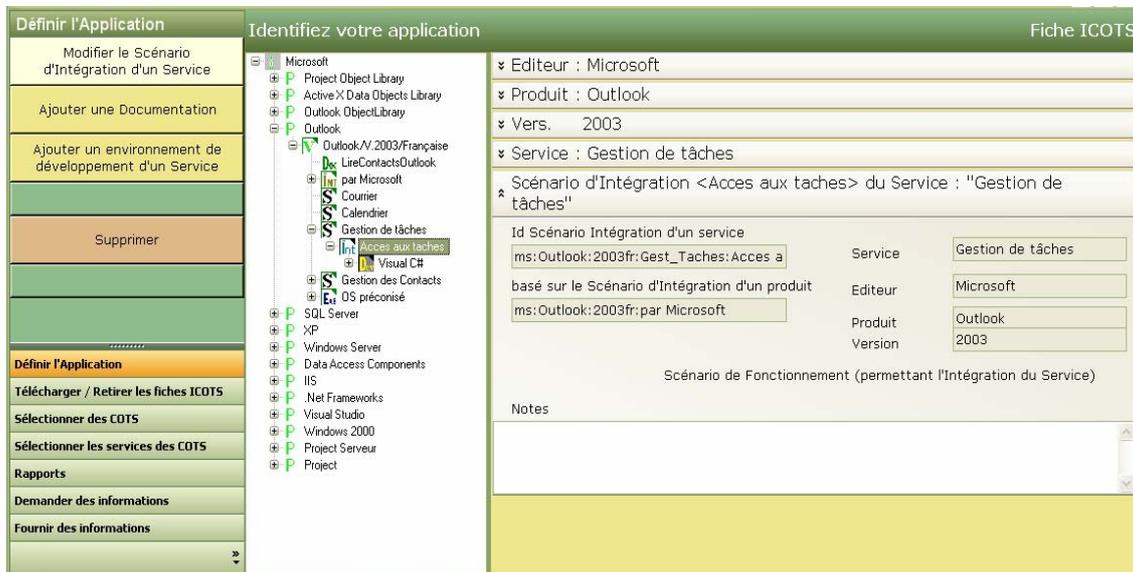


Figure 76 : Ajout d'un scénario d'intégration associé à un service

Dans ce cas, nous avons ajouté un scénario d'intégration appelé « Accès aux tâches » permettant d'intégrer le service « Gestion des Tâches » de « Ms Outlook ». Il est composé d'un environnement de développement : « Visual C# ». Celui-ci détermine l'outil de développement et le langage de programmation associé à cet environnement : « Visual Studio » et « C# ». C'est un scénario dit de « Fonctionnement » car il décrit un scénario permettant l'intégration du COTS. Par contre il est possible aussi avec la même interface utilisateur de décrire un scénario dit de dysfonctionnement. Celui-ci permettra dans le cadre des rapports de dysfonctionnement de prévenir le développeur du prototype que les éléments constituant ce scénario posent un problème pour l'intégration du COTS. Dans le cas de « Ms Outlook » nous n'avons pas d'exemple à proposer.

Revenons au niveau de la description du produit et continuons les différentes actions qui nous sont proposées. Il nous manque de décrire l'ajout du scénario d'intégration. Celui la même que nous venons d'associer au service de « Ms Outlook ». Voici donc la fiche de description de cet élément (Figure 77).

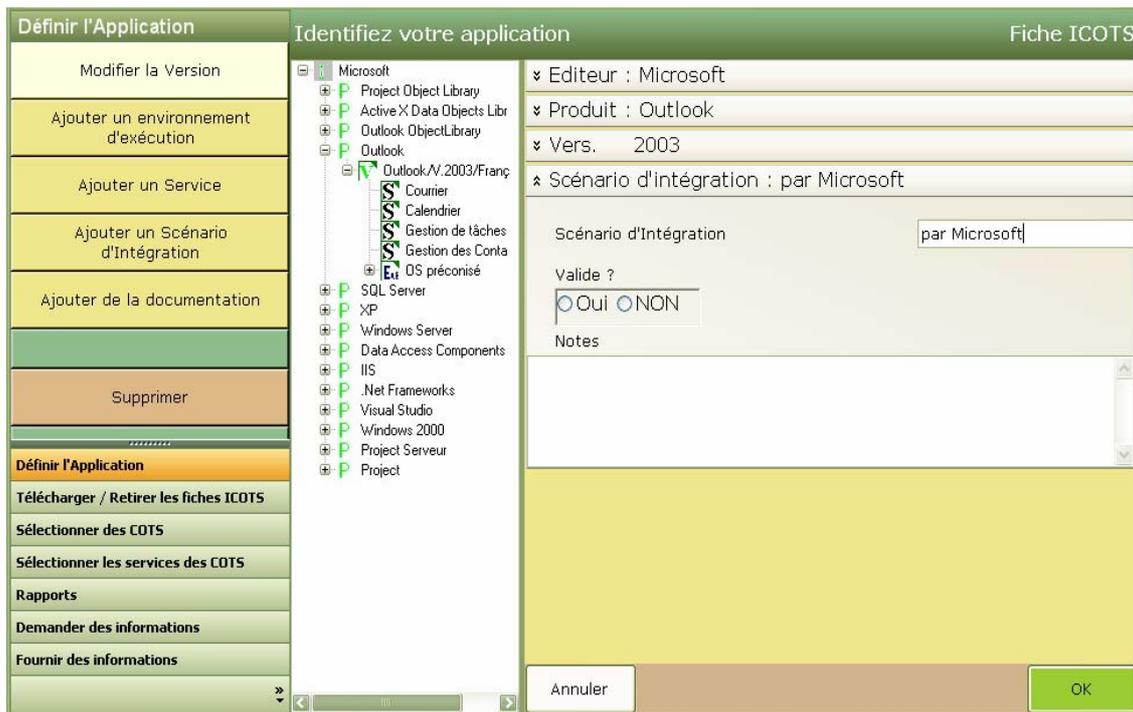


Figure 77 : Ajout du scénario d'Intégration

Ce scénario va permettre d'identifier un environnement de développement, un environnement d'exécution associé à ce scénario d'intégration et une documentation permettant d'expliquer l'intégration de « Ms Outlook ».

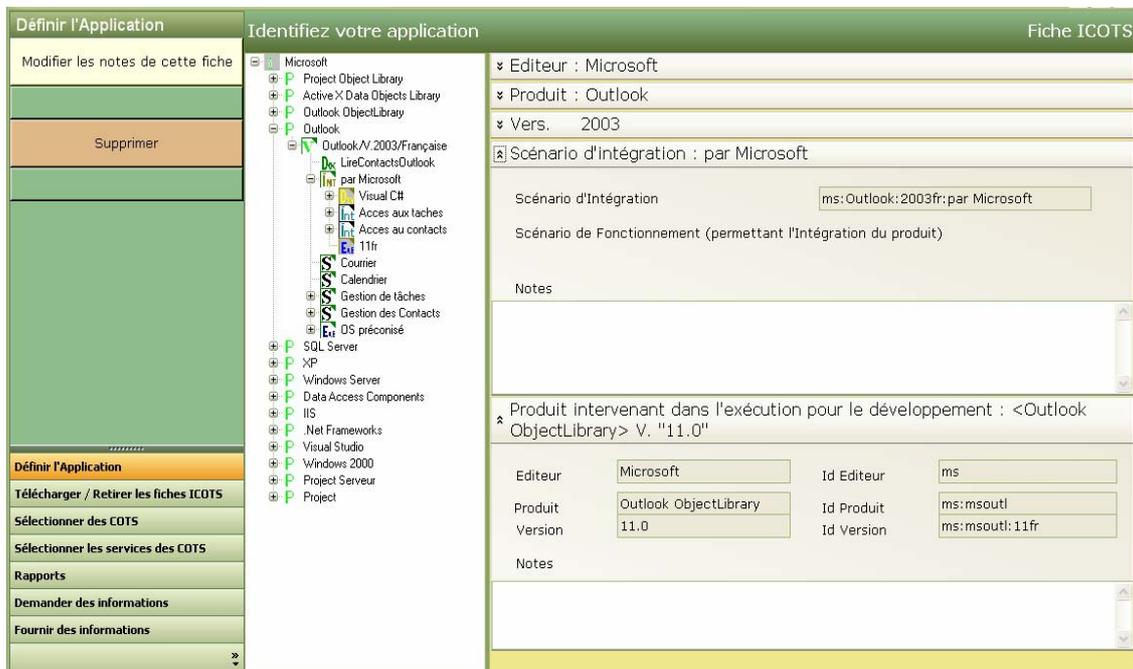


Figure 78 : Ajout d'un produit nécessaire pour l'environnement de développement

Dans le cas de « Ms Outlook » et de ce scénario d'intégration il est nécessaire d'utiliser la librairie « Outlook ObjectLibrary Version 11.0 » (Figure 78).

Dans le scénario d'intégration on spécifie l'outil de développement : « Visual Studio » (Figure 79).

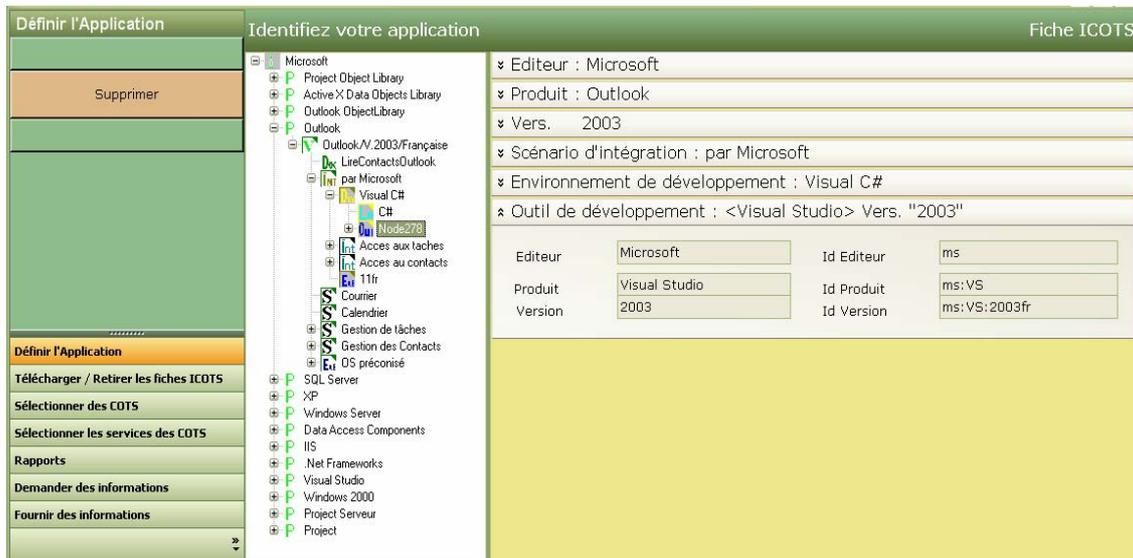


Figure 79 : Ajout de l'outil de développement

On spécifie aussi le langage de programmation C# (Figure 80).

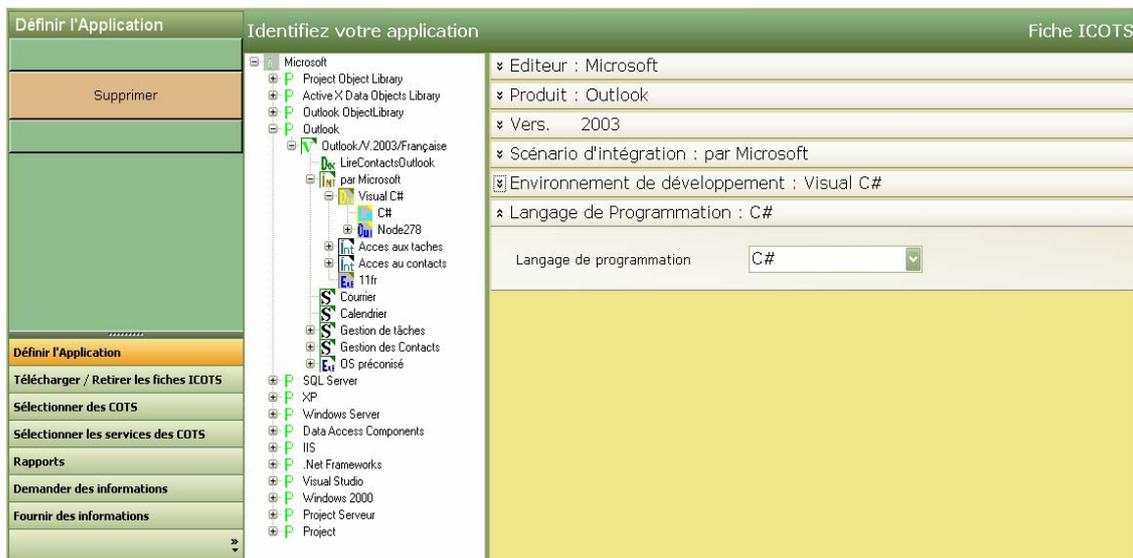


Figure 80 : Ajout du langage de programmation

Après la création de la fiche ICOTS de « MS Outlook » le fournisseur a la possibilité de la diffuser pour qu'elle devienne accessible par les utilisateurs de ce COTS. Cette action est automatique sur notre démonstrateur, même si dans une utilisation industrielle cette action nécessiterait une validation de la part du fournisseur.

Une fois la fiche créée, l'utilisateur de COTS va pouvoir la télécharger pour l'exploiter dans la réalisation d'une application à base de COTS comme nous allons le voir ci-dessous.

I - 3 L'utilisateur de COTS

I - 3.1 Présentation des actions principales

L'utilisateur de COTS suit une démarche que nous avons présentée dans le Chapitre 5 : II - 2 page 148. Elle est reprise à travers les menus situés dans la partie en bas et à gauche de l'interface du démonstrateur (Figure 81).

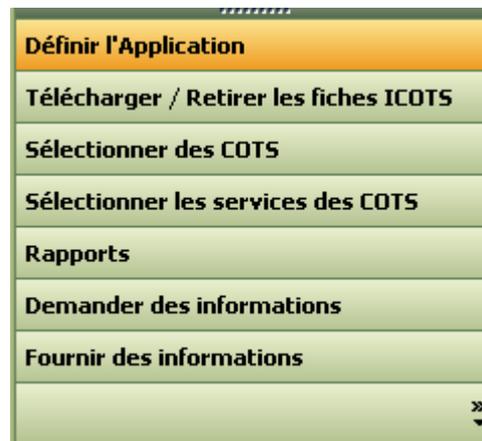


Figure 81 : Menu Général du démonstrateur

La première étape pour l'utilisation de ce démonstrateur correspond au premier élément de ce menu : « Définir l'Application ». C'est en créant la fiche ICOTS de l'application à base de COTS que l'utilisateur de COTS initialise la démarche. Par la suite il téléchargera les fiches ICOTS correspondant aux produits qu'il désire intégrer dans son application. Ce sont les services de ces COTS qui seront associés aux services de son application permettant d'établir un lien entre l'application à base de COTS et les différents COTS utilisés. C'est essentiellement par ces deux actions (la création de la fiche ICOTS de l'application à base de COTS et de l'association des services) que le développeur pourra accéder aux informations utiles qui l'aideront dans la réalisation du prototype et dans l'exploitation des rapports présentés dans le démonstrateur.

I - 3.2 Décrire l'application à base de COTS ou le prototype

Dans l'exemple que nous avons choisi, nous allons décrire une partie de l'application IPPOP (Intégration Produit Processus Organisation pour l'amélioration de la Performance en ingénierie). Dans cette application nous avons besoin des services d'un gestionnaire de messageries, de la manipulation de fiches d'informations liées aux utilisateurs, d'une liste d'utilisateurs à des fins d'identification et de la manipulation des tâches comme le font aujourd'hui des gestionnaires de projets. Ce sont quelques exemples de services qui sont nécessaire à la réalisation de l'application IPPOP.

Nous allons donc décrire ce besoin à l'aide du démonstrateur.

Comme précédemment pour le fournisseur qui établit la fiche de « Ms Outlook » nous créons une fiche IPPOP avec son numéro de version et les services que nous voulons vous présenter (Figure 82).

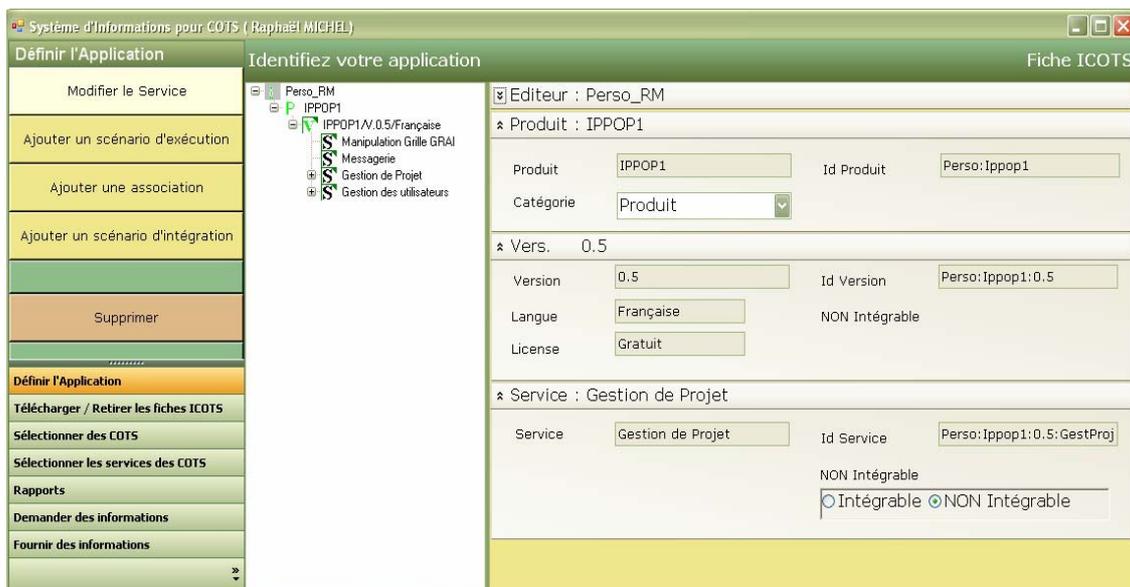


Figure 82 : Fiche IPPOP avec la liste des services

Nous avons dans cette fiche quatre services : la « manipulation de la grille GRAI », la « Messagerie », la « gestion de projet » et la « gestion des utilisateurs ».

Nous complétons la fiche avec l'environnement d'exécution cible de l'application que nous désirons développer. Dans l'exemple ci-dessous IPPOP doit être compatible avec « Windows 2000 SP2 » (Figure 83).



Figure 83 : Ajout du scénario d'exécution d'IPPOP

Nous pouvons aussi, même si ce n'est pas forcément utile, ajouter à cette description un scénario d'intégration. Ce dernier sert à identifier des outils et des langages préférés pour le développement de l'application à base de COTS. Cette information ne sert pas directement pour le développement de l'application à base de COTS mais peut avoir un intérêt dans une phase de sélection. Le démonstrateur pourrait comparer les informations issues des COTS avec les préférences de développement de l'application IPPOP.

Dans l'exemple ci-dessous le démonstrateur propose deux scénarios avec des outils de développement différents : l'un avec « Visual Studio » (Figure 84) et l'autre avec « WinDev ».



Figure 84 : Outil de développement pour l'intégration des COTS

Une fois ces informations introduites dans le démonstrateur, nous allons pouvoir choisir les COTS qui vont permettre de traiter certains services de l'application IPPOP. Nous avons choisi les services « Gestion des utilisateurs » et « Gestion de Projet » respectivement attribué aux COTS « Ms Outlook » et « Ms Project ». Nous avons donc besoin de récupérer les fiches ICOTS de ces deux produits.

I - 3.3 Le téléchargement des fiches ICOTS

La fonction des téléchargements des fiches ICOTS s'apparente à de la veille technologique. Le développeur va se renseigner sur différents produits qui sont susceptibles de convenir à de futures applications. Pour cela il télécharge les fiches ICOTS pour obtenir des informations sur les services, les environnements d'exécution ou la manière d'intégrer les COTS. A partir de ces fiches, il va sélectionner les COTS qui seront utilisés dans l'application IPPOP.

Dans un premier temps il sélectionne dans le menu général : « Télécharger/Retirer les fiches ICOTS ». Puis il sélectionne les fiches à télécharger. Celles-ci sont éditables avant et après le téléchargement pour contrôler les informations qu'elles contiennent comme dans l'exemple de la sélection de « MS Project » et le téléchargement de « MS Outlook » (Figure 85).

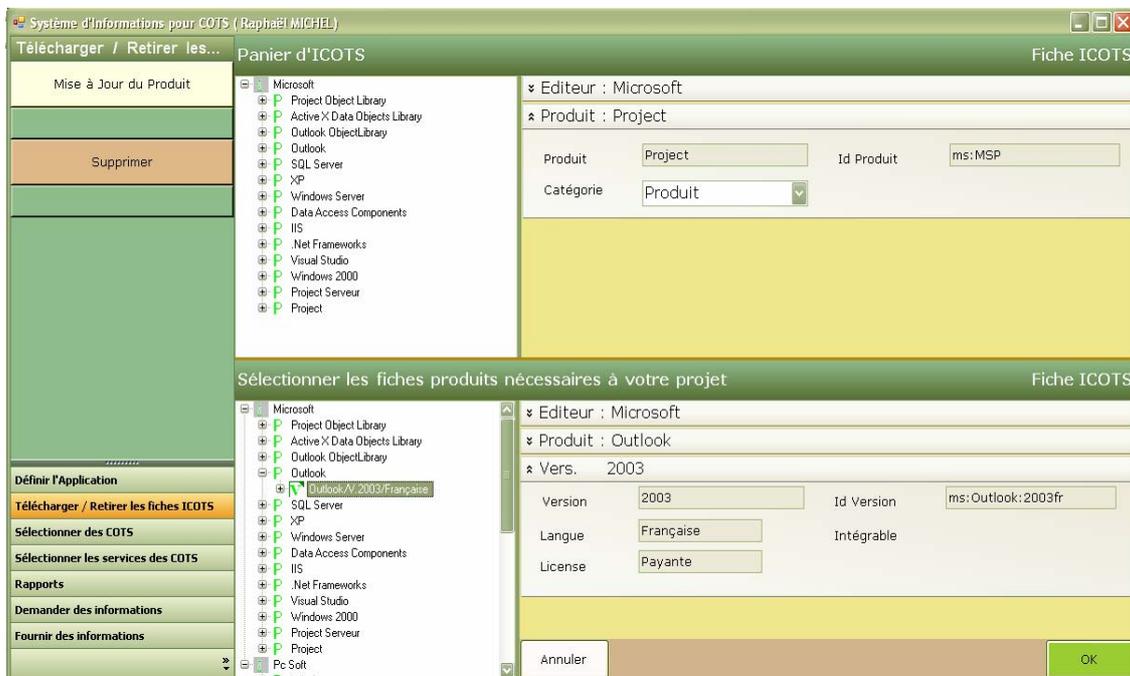


Figure 85 : Téléchargement d'une fiche ICOTS

Parmi toutes ces fiches on demande à l'utilisateur du démonstrateur de sélectionner les produits (et donc les fiches) qui sont nécessaires pour IPPOP : dans notre exemple « MS Project » et « MS Outlook ».

Puis nous allons sélectionner les services de chaque COTS qui seront utiles à l'application IPPOP : La « gestion des contacts » de « MS Outlook » et la « Gestion de Projet » de « MS Project ».

Cette dernière étape servira de filtre pour l'association entre les services de COTS utiles à IPPOP et les services à réaliser dans IPPOP. C'est aussi pour une future évolution du

démonstrateur, la possibilité d'utiliser cette étape pour créer des scénarios permettant de comparer plusieurs configurations de COTS pour la réalisation de l'application IPPOP.

I - 3.4 Associer les services de l'application à ceux des COTS à intégrer

Maintenant que l'on a décrit l'application IPPOP et que l'on connaît les COTS qui doivent être utilisés dans cette application, nous allons pouvoir identifier les services des COTS qui remplaceront les services de l'application IPPOP. C'est ce que nous avons appelé l'association des services.

Pour cela l'utilisateur du démonstrateur sélectionne un service IPPOP et indique (par sélection) le service d'un COTS qui va le réaliser (Figure 86).

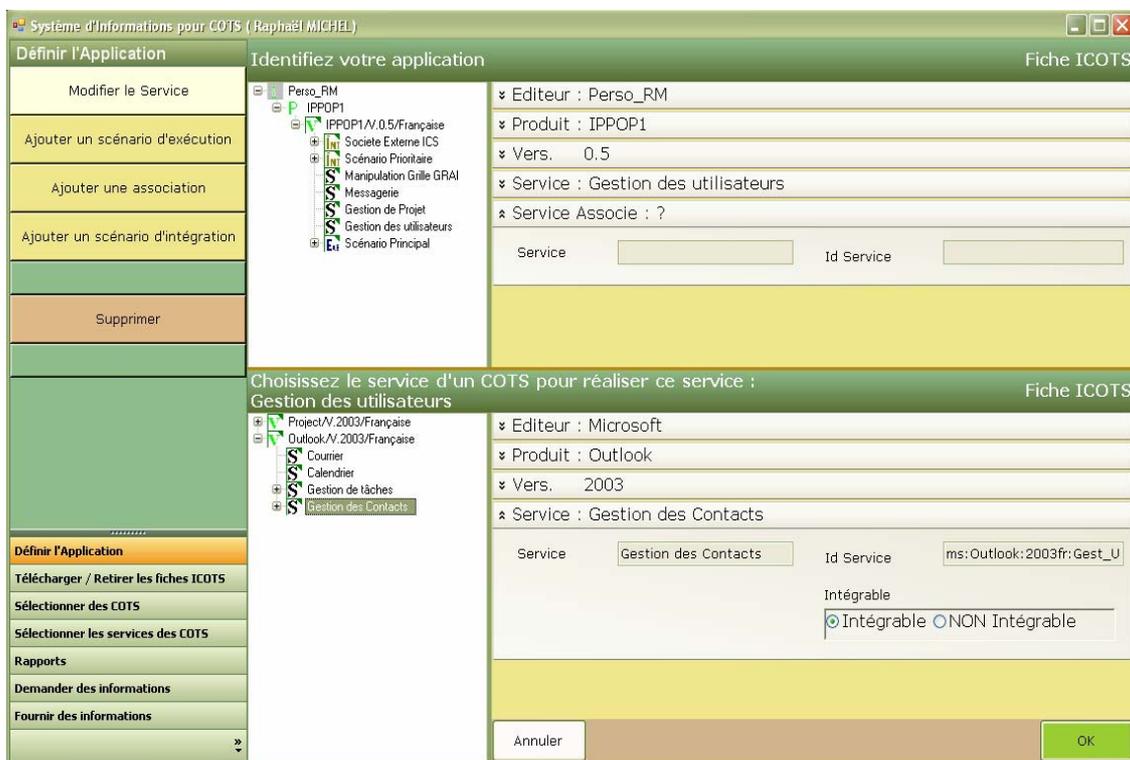


Figure 86 : Association de services

Cette étape est importante car, à elle seule, elle permet au démonstrateur d'établir un traitement sur les données et de proposer les trois rapports mentionnés dans ce mémoire : sur l'environnement d'exécution, l'environnement d'intégration et le rapport de dysfonctionnement.

I - 3.5 Accéder aux informations proposées par le démonstrateur

Lorsque les fiches ICOTS sont téléchargées il est possible en sélectionnant un élément de l'arborescence d'obtenir l'ensemble des informations de la fiche : services, environnements d'exécution, environnement d'intégration, documentation, ... De plus le fait d'associer les services comme nous venons de le voir, le démonstrateur peut proposer à son utilisateur des rapports. Ces derniers correspondent à une analyse des informations renseignées tout au long des manipulations que l'on vient de décrire.

La première opération pour accéder aux rapports c'est de sélectionner le scénario d'intégration de chaque COTS utilisés pour l'application IPPOP. Dans l'exemple que nous utilisons ici, « Ms Outlook » ne propose qu'un scénario alors que « MS Project » en propose deux. L'un (appelé scénario standard) permet une intégration simple du produit alors que le deuxième (appelé scénario alternatif) fait appel à des services internet pour son implémentation en utilisant des produits annexes comme « Ms Project serveur ». Le choix du scénario est donc important pour déterminer l'environnement d'exécution et d'intégration.

Commençons par choisir le scénario standard pour « Ms Project » (Figure 87).

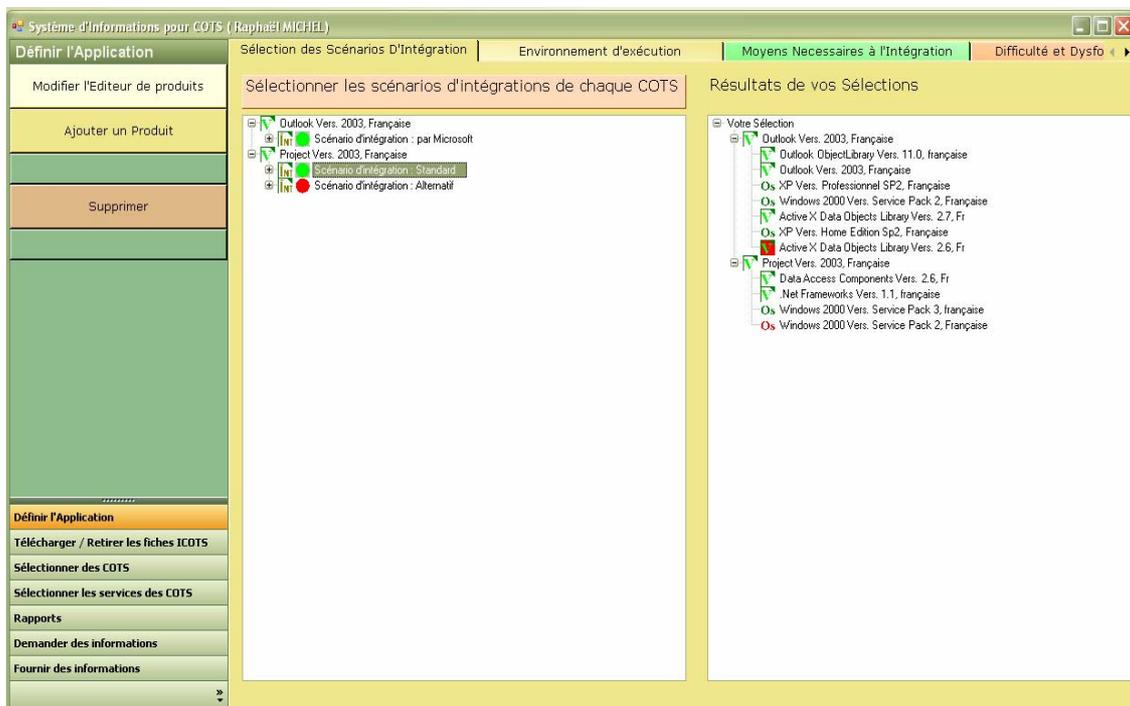


Figure 87 : Choix des scénarios d'intégration pour les rapports

Cette première étape permet de visualiser la liste des produits détectés dans les fiches en fonction des choix des scénarios. En rouge nous pouvons apercevoir les produits indiqués dans chaque COTS comme étant non conforme avec son utilisation.

A partir de là nous pouvons voir la synthèse des informations en fonction de l'environnement d'exécution et de celui d'intégration.

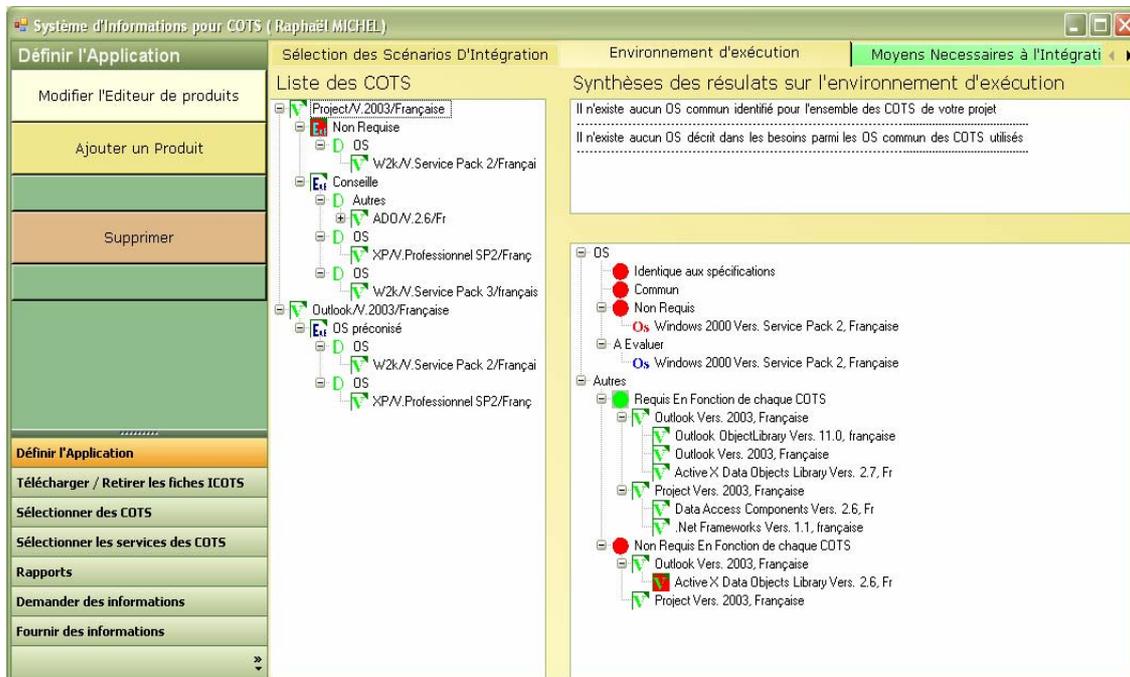


Figure 88 : Résultat de la synthèse pour l'environnement d'exécution

Le démonstrateur propose pour chaque COTS les informations relatives aux différents environnements d'exécution présentes dans les fiches. De plus la synthèse en haut à droite de l'interface nous avertit qu'il n'existe aucun OS commun avec les spécifications et pour l'ensemble des COTS à intégrer. Le démonstrateur détaille en fonction de chaque COTS la liste des produits (non OS) nécessaires pour le fonctionnement du COTS et ceux interdits avec l'utilisation des COTS. C'est le cas notamment pour « Active X Data Object Library vers 2.6 » qui est incompatible avec l'utilisation de « Outlook Vers 2003 ».

Puis l'utilisateur du démonstrateur peut accéder à l'onglet sur les « moyens nécessaires à l'intégration » (Figure 89).



Figure 89 : Résultat du démonstrateur sur les moyens d'intégrations

Le démonstrateur nous propose cinq familles d'informations :

- la liste des logiciels indispensables pour l'intégration en fonction de la sélection des scénarios d'intégration,
- la liste des services à intégrer qui sont identifiés à l'aide de l'association entre les services d'IPPOP et les services des COTS,
- la liste des services restant à développer

- les outils de développement nécessaires pour l'intégration de chaque service avec les langages de programmation et la documentation associée et
- une dernière zone pouvant présenter une documentation comme dans l'exemple ci-dessus de l'exemple de code nécessaire pour intégrer le service de « gestion des contacts » de « Ms Outlook ».

Le dernier rapport propose une synthèse des informations de dysfonctionnement contenues dans les fiches (Figure 90).

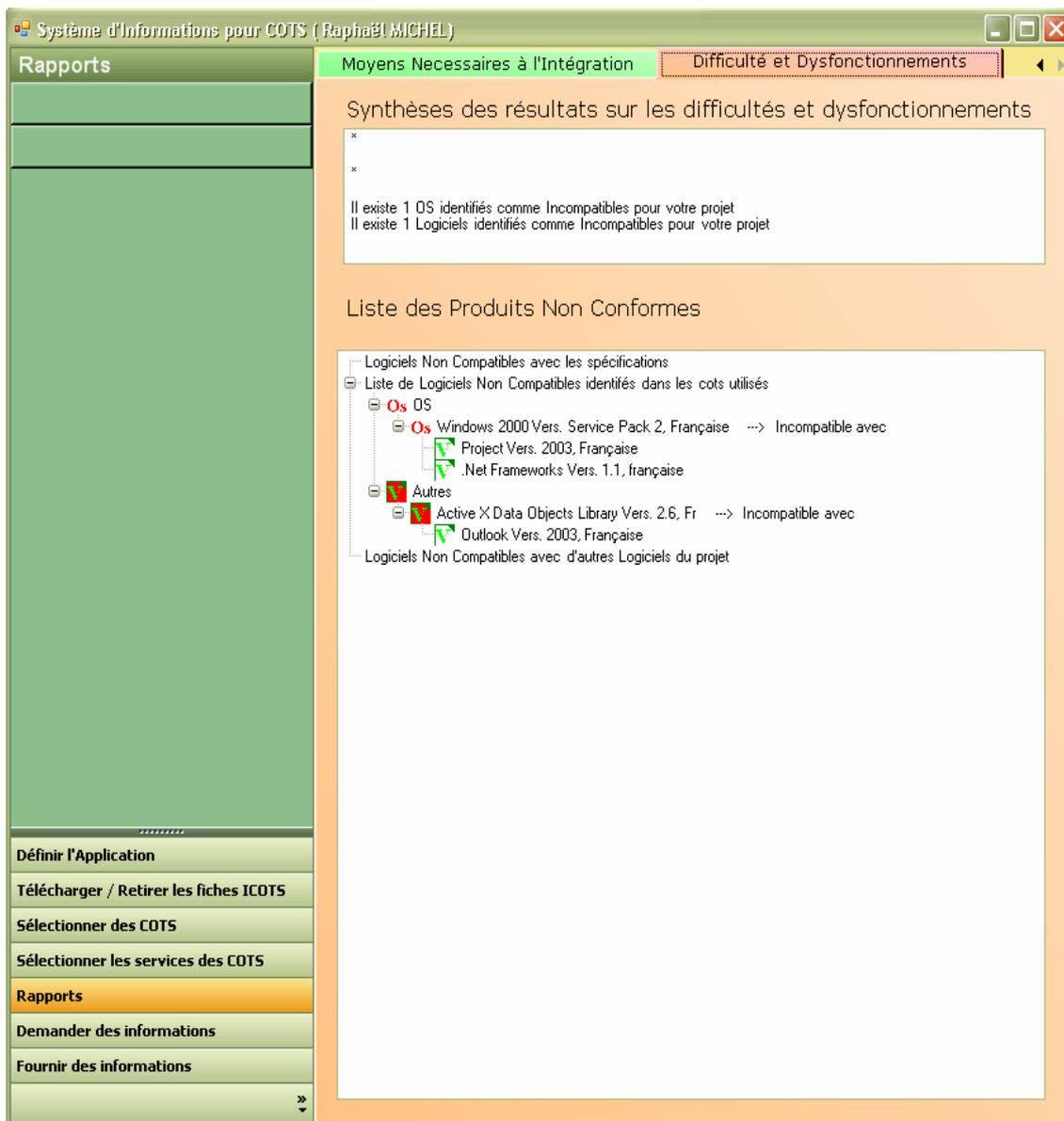


Figure 90 : Exemple d'un rapport de dysfonctionnement

Le démonstrateur propose une synthèse des dysfonctionnements en fonction des informations contenues dans les fiches. Dans le cas présent, le démonstrateur avertit

l'utilisateur que « Windows 200 SP2 » est incompatible avec l'utilisation de « Project Vers 2003 » et de « .Net Frameworks Vers1.1 ». Aussi au niveau des autres produits (non OS) le démonstrateur prévient que « Active X Data Object Library Vers 2.3 » est incompatible avec l'utilisation de Outlook vers 2003 ».

Ces informations sur les données détenues dans les fiches sont un avertissement pour le développeur de l'application IPPOP. La présentation qui en est faite correspond à une synthèse de ces informations.

Par contre si dans la première étape de l'utilisation des rapports, nous avons fait le choix d'utiliser pour l'intégration de « MS Project » le scénario appelé « Alternatif » le résultat de ce rapport nous indiquerait des incompatibilités empêchant cette intégration (Figure 91).

The screenshot shows a software interface titled "Système d'Informations pour COTS (Raphaël MICHEL)". The main window is divided into two panes. The left pane, titled "Définir l'Application", contains several buttons: "Modifier l'Editeur de produits", "Ajouter un Produit", "Supprimer", and a "Définir l'Application" button at the bottom. The right pane, titled "Moyens Necessaires à l'Intégration" and "Difficulté et Dysfonctionnements", displays a report. The report has a title "Synthèses des résultats sur les difficultés et dysfonctionnements" and contains the following text: "Il existe 1 Logiciels Non Compatibles avec un autre Logiciels du projet", "Il existe 1 OS identifiés comme Incompatibles pour votre projet", and "Il existe 1 Logiciels identifiés comme Incompatibles pour votre projet". Below this text is a section titled "Liste des Produits Non Conformes" which contains a tree view of incompatibilities. The tree view is organized as follows: "Logiciels Non Compatibles avec les spécifications" (expanded) -> "Liste de Logiciels Non Compatibles identifiés dans les cots utilisés" (expanded) -> "OS OS" (expanded) -> "Os Windows 2000 Vers. Service Pack 2, Française" (with a red 'X' icon) -> "Incompatible avec Project Vers. 2003, Française" (with a green checkmark icon); "Autres" (expanded) -> "Active X Data Objects Library Vers. 2.6, Fr" (with a red 'X' icon) -> "Incompatible avec Outlook Vers. 2003, Française" (with a green checkmark icon). "Logiciels Non Compatibles avec d'autres Logiciels du projet" (expanded) -> "Active X Data Objects Library Vers. 2.6, Fr" (with a red 'X' icon) -> "Incompatible avec Outlook Vers. 2003, Française" (with a green checkmark icon); "Nécessaire avec Project Vers. 2003, Française" (with a green checkmark icon).

Figure 91 : Deuxième exemple de rapports de dysfonctionnement

Dans ce cas, la dernière partie de l'arborescence (Logiciels Non Compatibles avec d'autres logiciels du projet) présente un dysfonctionnement dû à l'utilisation du produit appelé « Active X Data Objects Library V 2.6 ». Celui-ci est incompatible avec l'utilisation de « Outlook Vers 2003 » et malheureusement il est indispensable à l'utilisation de « Project Vers 2003 ». Dans cet exemple il n'est pas possible d'intégrer dans la même application « Ms Outlook » et « Ms Project » en utilisant les scénarios d'intégration qui ont été sélectionnés.

I - 4 Conclusion

L'exemple du développement d'IPPOP nous a permis de dérouler l'ensemble des écrans du démonstrateur utilisés par le fournisseur et l'utilisateur de COTS. Le démonstrateur permet de saisir les fiches ICOTS, de les manipuler puis de réaliser des traitements conduisant à la présentation de rapports sur l'utilisation des COTS dans une application. Celui-ci par le biais des informations qu'il manipule autour des environnements d'exécution et d'intégration est aussi une aide au développement d'application à base de COTS. Il renseigne le développeur sur les outils de développement permettant l'intégration des COTS. Le démonstrateur propose aussi des informations sur la manière de réaliser l'intégration de chaque service des COTS.

L'utilisation du démonstrateur dans le cadre de cet exemple permet d'apprécier la facilité avec laquelle le développeur d'application d'IPPOP récupère les informations utiles pour son développement. Il a aussi la possibilité de réaliser des traitements sur les informations comme avec les exemples des trois rapports inclus dans le démonstrateur pour analyser les conséquences d'une intégration de COTS dans son application. Dans notre exemple le choix du scénario d'intégration est important puisque en fonction de ce choix l'issue du développement n'est pas la même. Dans un cas l'intégration est réalisable alors dans le second cas elle est vouée à l'échec. Le démonstrateur peut facilement proposer des analyses qui aideront sans aucun doute le développeur de l'application à base de COTS.