Stage de fin d'étude
2 Avril au 27 Septembre 2002

Conception et développement d'applications Peer to Peer

BERNARDET William

Promotion RICM 2002
Option Architecture et Réseaux

Maître de stage : Raphaël LEBLANC
Responsable ISTG : Olivier RICHARD

Remerciements

Je voudrais particulièrement remercier M. Jean BUFFET, pour m'avoir permis d'effectuer ce stage au sein du service Recherche et Technologie de *Dassault Systèmes*.

Je tiens aussi à remercier mon maître de stage Raphaël LEBLANC pour sa disponibilité et ses précieux conseils.

Je remercie également toutes les autres personnes de la société que j'ai côtoyées durant mon stage et qui ont fait que celui-ci se passe dans les meilleures conditions possibles.

Avertissement

Sommaire

Table des figures

Introduction

La technologie Peer to Peer, actuellement en pleine expansion, fait couler beaucoup d'encre. En effet elle doit sa renommé à des applications, telles que « Gnutella » ou « Napster », qui utilisent la cette technologie toujours dans le même but : le partage de fichiers.

Le service Recherche et Technologie, dans lequel j'ai effectué mon stage, est un maillon stratégique de *Dassault Systèmes* dans le cadre des recherches sur les technologies d'avenir. Le développement de nouvelles solutions basées sur la technologie Peer to Peer promet d'apporter un réel avantage dans le domaine du travail collaboratif. Le but de mon stage est justement de concevoir des applications collaboratives mettant en oeuvre cette technologie.

Ce rapport présente le travail réalisé durant le stage. Dans un premier temps, la société *Dassault Systèmes* vous sera présentée. Ensuite un chapitre introductif décrira brièvement l'architecture de CATIA et de son environnement de développement. Une troisième partie consacrée au Peer to Peer, détaillera la plate-forme utilisée. Enfin la dernière section relatera les applications développées lors de ce stage.

Présentation de l'entreprise

Créée en 1981, pour poursuivre le développement d'outils de CAO lancé par les *Avions Marcel Dassault*, la société *Dassault Systèmes* connaît depuis une croissance régulière, et est aujourd'hui reconnue comme le leader mondial de son marché.

I. SECTEUR D'ACTIVITE : LA CFAO

Avant l'apparition des premiers systèmes de CAO, les produits industriels étaient conçus sur papier dans les bureaux d'études, et les données et dessins nécessaires à la définition des différentes pièces d'un produit circulaient dans l'entreprise sous la forme d'une quantité importante de plans et de rapports divers. Avec l'arrivée des logiciels de **CFAO** (**C**onception et **F**abrication **A**ssistées par **O**rdinateur) depuis une vingtaine d'années, le stockage de ces données devient beaucoup plus aisé, permettant la conception de produits industriels de façon essentiellement numérique, sans recours à des maquettes physiques.

Cette évolution, sinon cette transformation des métiers de la conception et de la fabrication d'un produit, trouve sa justification dans l'outil lui-même. En effet, il s'agit d'un outil à la fois de conception et de calcul, mais aussi et surtout de communication dans l'espace 3D. Ces évolutions ont permis une forte réduction des coûts et une diminution importante des temps de cycle de développement. Désormais, les logiciels de CFAO sont au cœur de nombreux projets aéronautiques et automobiles, comme la Twingo et le Boeing 777.

La croissance dans ce secteur confirme bien le développement de l'utilisation de ces logiciels. Les méthodes de conception dans l'industrie se transforment rapidement et ce à tous les niveaux, depuis le cahier des charges du produit jusqu'à sa fabrication. Cette évolution concerne aussi bien les concepteurs que les ingénieurs.

II. LE GROUPE DASSAULT SYSTEMES

I.2. Présentation générale

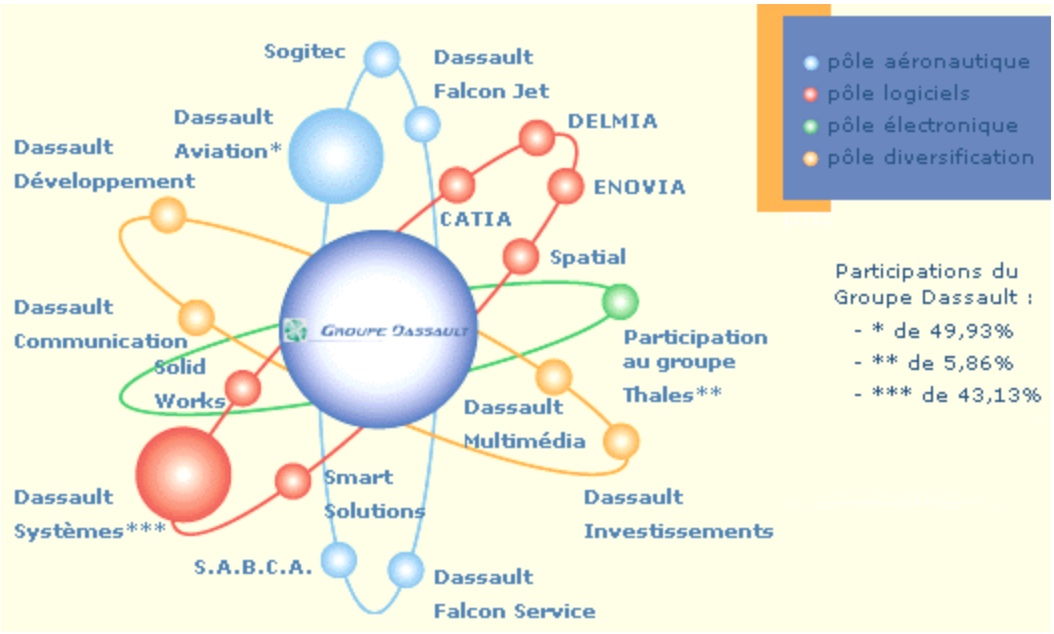
Les logiciels développés par *Dassault Systèmes* sont articulés autour de deux marchés: le marché orienté processus industriel ou Gestion de Développement de Produit (PDMI), et celui orienté outils de conception (Conception Assistée par Ordinateur, Fabrication Assistée par Ordinateur, Imagerie Assistée par Ordinateur). *Dassault Systèmes* propose les solutions *CATIA*, *DELMIA* et *ENOVIA,* leaders sur leurs marchés respectifs.

- *CATIA* comprend une gamme de 150 produits qui permettent la simulation de l'ensemble des phases de conception, d'analyse, d'assemblage, de fabrication et de maintenance d'un produit.

- *DELMIA* fournit une gamme complète de logiciels permettant de concevoir simuler et optimiser les processus de fabrication et les ressources associées : les outillages aux chaînes d'assemblage tout en rendant plus efficaces les taches manuelles. *DELMIA* offre des technologies de pointe pour concevoir virtuellement des usines, avant de les construire.
- Dans le domaine de la Gestion de Développement Produit, *ENOVIA* fournit un environnement d'ingénierie partagé avec plusieurs systèmes CAO. Il permet de gérer graphiquement et de partager les informations sur les produits, les processus, et les ressources de fabrication, à tous les niveaux de l'entreprise et avec ses partenaires, durant le cycle entier de vie du produit.
- *Dassault Systèmes* propose le logiciel *SolidWorks* qui a pour mission d'apporter la puissance de l'outil de modélisation en trois dimensions sur le bureau de chaque ingénieur.

Les données *CATIA*, *DELMIA*, et *ENOVIA* sont accessibles à tous, au travers du Web. Grâce à *CATWeb* et *ENOVIA Portal*, tous les intervenants d'un projet peuvent travailler simultanément sur la maquette digitale devenue visible, les différents services de l'entreprise, les distributeurs, les sous-traitant, les acheteurs.

La base installée compte plus de 12 000 clients *CATIA-CADAM*[1] et 9 500 clients *SolidWorks*. La majorité des produits *Dassault Systèmes* est installée en Europe (50%), le reste en Amérique (30%) et en Asie (20%).



Le groupe Dassault et ses filiales

[1] Computer Augmented Drafting And Manufacturing

A ce jour, *Dassault Systèmes* possède trois centres de compétences principales qui fournissent support technique et services : *Dassault Systèmes of America* et ses agences pour l'Amérique du Nord et l'Amérique latine, *Dassault Systèmes K.K.* pour le Japon et *Dassault Systèmes France* et ses agences pour l'Europe.

Société	*Localisation*	*Activités*
Dassault Systèmes	Suresnes	Définition de la stratégie CFAO Applications Mécaniques CATIA – CADAM Marketing et support
Dassault Systèmes of America	Los Angeles	Applications Mécaniques CATIA – CADAM CADAM Solutions Construction d'usines Marketing et support
Dassault Systèmes K.K.	Tokyo	Marketing et support
Dassault Data Service	Suresnes	Prestation de services
ENOVIA	Caroline du Nord	Solutions de gestion de données techniques de nouvelle génération (PDM II)
DENEB	Michigan	Entreprise de fabrication digitale
SolidWorks	Massachusetts	Solutions économiques de conception 3D sous Windows
Dassault Systèmes AG	Hanovre et Munich	Applications avancées pour le style Centre de compétence pour les clients allemands
Dassault Systèmes Provence	Aix en Provence	Applications avancées surfaciques Maintenance et évolution des produits MATRA
MATRA Datavision		Solutions d'ingénierie et de services Support d'Euclid

Le succès de *Dassault Systèmes* sur le marché doit beaucoup à son partenariat des premiers jours avec la firme *IBM* qui a introduit sur le marché, vendu et supporté *CATIA* à travers le monde depuis 1981. Depuis, les forces de vente d'*IBM* ont fait avec succès la promotion du logiciel *CATIA* comme leur fer de lance dans le domaine de la CAO/CFAO et aidé à introduire le produit chez certaines des plus grandes compagnies industrielles du monde et leurs fournisseurs.

IBM possède des intérêts minoritaires chez *Dassault Systèmes*, acquis lors de la cession de *CADAM*, et possède 50% de la filiale japonaise *Dassault Systèmes K.K.*.

CATIA et *CADAM* répondent aussi bien aux besoins des petites et moyennes entreprises qu'à ceux des grands comptes : 50% des clients possèdent moins de 5 sièges alors que les grandes sociétés

travaillent en ingénierie simultanée[2] avec des milliers de postes répartis sur plusieurs sites dans différents pays.

Dassault Systèmes compte aujourd'hui parmi ses clients, dans de nombreux domaines:

Secteur	*Principaux clients*
Aéronautique et Aérospatial	Dassault Aviation, Boeing, Bombardier (Canadair), SNECMA, SEP, AlliedSignal Engines, Lockheed Martin, British Aerospace (avions militaires), …
Automobile	Renault, Ferrari, Mercedes-Benz, Peugeot, Porsche, Saab, Volkswagen, Volvo, BMW, Honda, FIAT, Rover, Cosworth (Road Car Engines), Sauber, Prost (écuries F1), …
Construction navale	General Dynamics Electric Boat, US Navy (sous-marins), …
Biens de consommation	Daewoo, Hyundai, Canon, Sony, Peavey Electronics, Honda, Black & Decker, l'Oréal, …
Conception d'usines et de centrales	Shell, Raythorn Engineers & Constructors (projet de réacteur thermonucléaire), …

CATIA a ainsi permis l'élaboration de projets médiatiques comme le bateau de *Guy Delage*, le prototype *Courage* ou bien la formule 1 de l'écurie *Prost Grand Prix* (cf. Figure 1).

Figure 1 - Rendu photo réaliste de biens de consommation, avec l'autorisation de l'écurie Prost

Même si *CATIA* est né de l'extrême complexité des projets aéronautiques, c'est à l'industrie automobile que ce logiciel de CFAO doit son succès.

Ainsi *CATIA* est au cœur de nombreux projets aéronautiques et automobiles et affiche également des références impressionnantes dans des domaines très variés allant de la construction navale à l'architecture et aux équipements médicaux jusqu'aux biens de consommation.

[2] L'ingénierie simultanée ou l'ingénierie concourante représente un ensemble d'actions à réaliser intelligemment entre tous les acteurs qui participent à un objectif commun. La notion d'objectif relègue au second plan l'organisation classique par fonctions de l'entreprise, pour focaliser sur la satisfaction du client, et donc sur les projets qui le concerne. Les acteurs interviennent de plus en plus tôt et peuvent influer sur la conception d'un produit ou d'un service.

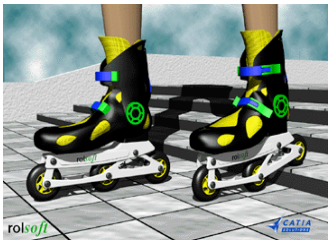


Figure 2 - Rendu photo réaliste de biens de consommation (Rolsoft)

Figure 3 - Rendu photo réaliste de biens de consommation(Electrolux)

II.2. Historique

- **Les années 70** La société ***A**vions **M**arcel **D**assault (**AMD**)* est précurseur dans le domaine de la conception et de la fabrication assistées par ordinateur. Premier client *CADAM* en Europe, *AMD* donne naissance en 1975 au logiciel *CATI* qui permet de concevoir et d'usiner la première voiture en soufflerie en 4 semaines au lieu de 6 mois.
- **1981** Le logiciel *CATI* est rebaptisé *CATIA* et la société *Dassault Systèmes* est créée pour répondre et anticiper le succès obtenu dans l'aéronautique. Elle fera évoluer *CATIA* tandis que sa commercialisation sera assurée par *IBM* dans le cadre d'un partenariat.
- **1982 à 1988** *Dassault Systèmes* sort les versions 1 à 3 de *CATIA*. Les évolutions successives du produit et la force commerciale d'*IBM* en font le leader dans les applications aéronautiques et automobiles.
- **1992** *Dassault Systèmes* décide d'élargir sa gamme de produits en reprenant les activités de développement de la société *CADAM Burbank* en Californie et crée ainsi une filiale appelée **DSA** (***D**assault **S**ystèmes of **A**merica)*. Celle-ci continue de développer le logiciel *CADAM* et, est désormais responsable du développement et du support de nouveaux produits ainsi que du maintien de l'interopérabilité entre *CATIA* et *CADAM*.
- **1994 à 1996** La version 4 de *CATIA-CADAM* Solutions est disponible sur HP 9000, *Silicon Graphics* et *Sun*. Parallèlement, la société *Dassault Systèmes* est introduite en bourse à Paris et au *NASDAQ* américain.
- **1997** *Dassault Systèmes* acquiert *Deneb Robotics* pour anticiper la demande de ses clients sur la fabrication digitale et *SolidWorks Corporation* pour répondre au marché des outils de conception. Elle élargit son offre et lance une nouvelle gamme de produits appelée *CATWEB*™ pour adresser les besoins du *Network Computing*.
- **1998** *Dassault Systèmes* crée *ENOVIA Corporation* dans le cadre du renforcement de son alliance avec *IBM* pour répondre au marché *PDM II* (Système de Gestion des données Techniques et de développement de produits de Nouvelle Génération). La version 5 de *CATIA* est disponible en fin d'année, fonctionnant aussi sous *Windows NT*. *Dassault Systèmes* acquiert *Matra Datavision* pour acquérir les technologies d'*Euclid*, à savoir *Euclid STYLER*, *Euclid MACHINIST*, *STRIM* et *STRIMFLOW*, ainsi que son expertise dans ces domaines. Ces produits, ainsi que l'accès sous licence à la technologie *CAS.CADE*[3], permettront à *Dassault Systèmes* d'élargir son offre actuelle, particulièrement dans le domaine de la modélisation de formes et de surfaces à caractère esthétique, l'usinage par commande numérique et la simulation d'injection plastique.
- **1999** *Dassault Systèmes* crée *Dassault Systèmes Provence* pour le développement d'applications avancées dans le domaine de la reconstruction de surface, d'outils de conception. D'autre part, *Dassault Systèmes AG* est établit en Allemagne en tant que son site

[3] *CAS.CADE* est un atelier de développement logiciel technique qui permet de créer des applications scientifiques et techniques notamment dans le domaine de la CFAO. Il offre un très large éventail de composants logiciels recouvrant les domaines de la géométrie, du graphique, de la base de donnée et du calcul des propriétés physiques des modèles créés. *CAS.CADE* est utilisé par *Matra Datavision* pour ses propres développements d'applications de CFAO et par plus de 70 sociétés dans le monde pour leur propre développement interne ou pour des applications commerciales. Les domaines d'application pour lesquels *CAS.CADE* est utilisé sont très variés : CAO/CFAO, structures navales, schématique, géo-sciences, reverse engineering, médical, etc.

de référence pour le développement de « styling solutions ». Enfin *Dassault Systèmes* devient actionnaire majoritaire de *Smart Solutions Ltd*. pour adresser complètement les segments du marché de *TeamPDM* et de *PDM II*.

- **2000** *Dassault Systèmes* annonce la création de *DELMIA*, un nouveau produit pour le « e-Manufacturing ». *DELMIA* regroupe les produits *Deneb*, *EAI-DELTA* et *SAFEWORK* sous une unique marque et offre des solutions de définition et de simulation des processus de fabrication numérique.

II.2. Dassault Systèmes en quelques chiffres

Chiffres clés de Dassault Systèmes (source rapport annuel 1999)

Dassault Systèmes est aujourd'hui le plus grand laboratoire de développement de logiciel en Europe avec plus de 2650 employés répartis sur 40 sites.

Depuis juin 1996, *Dassault Systèmes* est cotée au Marché à Règlement Mensuel[4] et au *NASDAQ* américain. Aujourd'hui la compagnie a distribué 55,4 millions d'actions.

En dix ans, la société a connu une croissance spectaculaire. Elle a vu ses effectifs passer de 20 personnes en 1981 à plus d'un millier aujourd'hui, pour la plupart des ingénieurs. Son chiffre d'affaires a progressé d'un million de francs français à plus d'un milliard et demi à ce jour.

En 1999, *Dassault Systèmes* a réalisé un chiffre d'affaires de 3,3 milliards de francs, en augmentation de 22,1% par rapport à 1998. On retrouve, comme en 1997 et 1998, un rythme de croissance égal à deux fois celui du marché et ce, malgré un ralentissement lié au phénomène du passage à l'an 2000. D'excellents résultats sur les deux segments de marché visés ont été obtenus, avec la vente de 28 698 licences *CATIA* sur le marché orienté Processus Industriels ("Process-Centric"), et de 17 168 licences *SolidWorks* sur le marché orienté Conception ("Design-Centric").

Grâce à ce total de 45 866 nouvelles licences, la part de marché dépasse le seuil des 20%. Le résultat d'exploitation et le résultat net – hors coûts d'acquisition – se sont élevés respectivement à 1,158 milliards de francs et 733 millions de francs, soit une croissance de 12,8% et 16,6% par rapport à l'exercice précédent. La marge d'exploitation s'est légèrement érodée en raison du niveau élevé des investissements réalisés, notamment pour renforcer les réseaux de vente *SolidWorks* et *Deneb*, et pour accélérer la mise à disposition de produits, plus particulièrement en matière de gestion des données et de fabrication digitale.

L'année 1999 marque un tournant dans l'histoire de Dassault Systèmes: la Société ayant pris la place de numéro 1 mondial de son secteur (estimation de Daratech). Dassault Systèmes a accompli des performances financières significatives grâce à une meilleure couverture des besoins des clients et à un approfondissement de sa relation avec IBM, tout en investissant pour assurer la croissance future de ses résultats.

Actuellement, Dassault Systèmes compte près de 1200 employés sur le site de Suresnes, qui est aussi son siège social.

4 Marché sur lequel le paiement des échanges ne se fait qu'à la fin du mois boursier, au jour de liquidation. Le Règlement Mensuel est la "star" de la bourse de Paris : c'est là que sont cotées les sociétés françaises et étrangères les plus importantes. C'est également lui qui enregistre le plus grand nombre de transactions pour les actions. Avec le marché au comptant, il forme la cote officielle.

III. PRESENTATION DU SERVICE SRT

Le Service de Recherche et nouvelles Technologies est une entité indépendante au sein de la direction de Stratégie de Dassault Systèmes. Son rôle est double: d'un coté assurer une veille technologique multi-domaines, de prototyper des solutions innovantes pour évaluer l'applicabilité des nouvelles technologies aux besoins produit de la société, afin de former du personnel et des stagiaires sur les technologies les plus avancées. Ce service doit assurer à tout moment un haut niveau de compétences dans tous les domaines techniques de l'entreprise ; ceci va de la gestion des bases de données jusqu'aux représentations géométriques, avec un accent particulier sur les technologies Internet. Ses membres essaient donc, dans les domaines clef, d'être toujours deux à cinq ans en avance sur le marché. Les travaux réalisés par ce service ont un but stratégique : permettre à Dassault Systèmes de rester le leader dans le domaine de la CFAO.

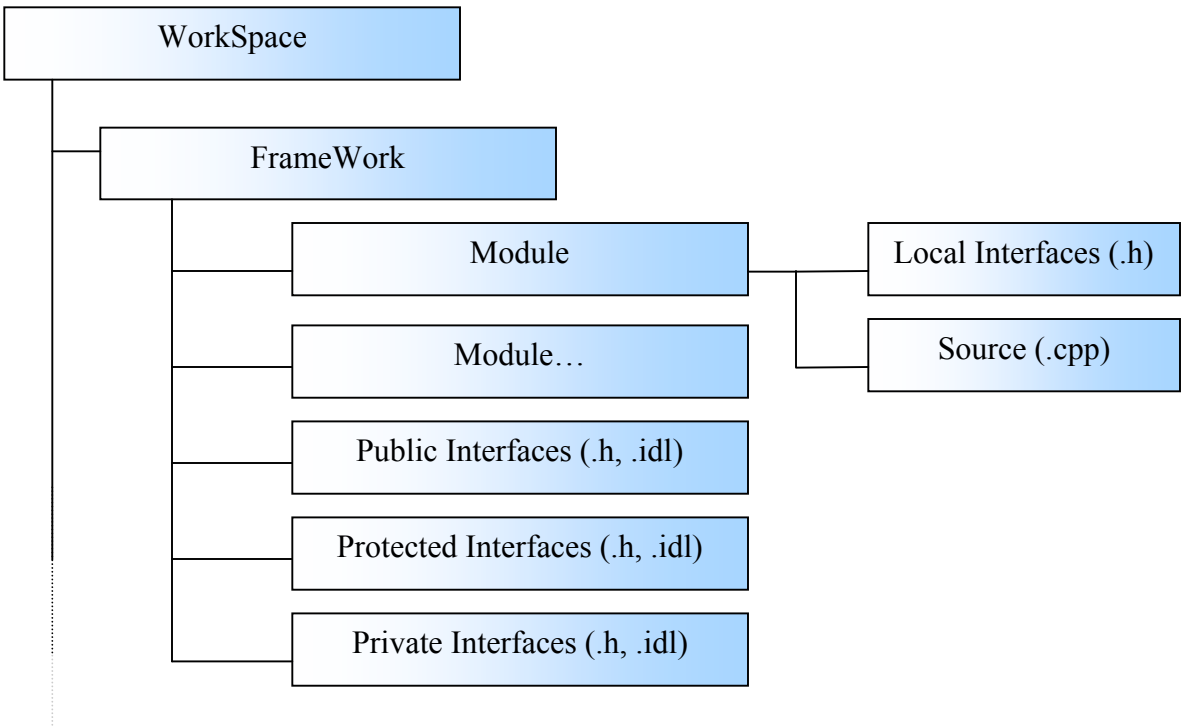
Architecture de CATIA

Mon stage consistant à réaliser des services Peer-to-Peer pour CATIA, il me paraissait primordial d'en expliquer brièvement la modélisation. Il faut savoir que le logiciel phare de Dassault Système est basé sur l'architecture nommée Component Application Architecture (CAA), actuellement version 5. Cette API est destinée à tous les développeurs désireux d'ajouter de nouvelles fonctionnalités aux logiciels tels que CATIA, ENOVIA ou encore DELMIA.

I. PRESENTATION TECHNIQUE DE CATIA

I.1. Organisation générale de CATIA

Les millions de lignes de code de CATIA sont organisées sur plusieurs niveaux permettant ainsi un meilleur découpage du code source.



Hiérarchie générale du code de CATIA V5

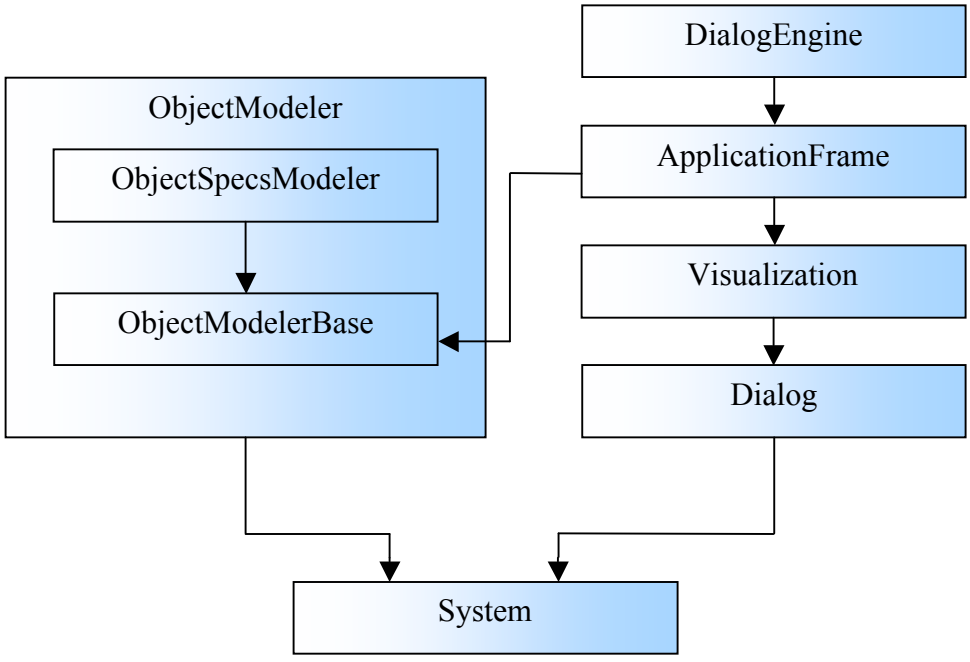
L'entité la plus importante de cette organisation est le Workspace, il représente l'application (ici CATIA). Ensuite viennent les Frameworks, ce sont des spécialisations, c'est à dire des ensembles de fonctionnalités regroupés autour d'une même activité. Il en existe de différentes sortes :

- System : Contient les classes de base de CATIA.
- Dialog : Classes permettant de gérer l'interface utilisateur.
- DialogEngine : Gestion des interactions avec l'utilisateur.
- Visualisation : Gestion du rendu.
- etc.…

Ce découpage du code joue un double rôle: d'un coté, il améliore l'architecture de l'application en diminuant les cohésions, et d'un autre, il facilite la gestion des différents projets. En effet chaque framework correspond de manière simplifiée à un service de *Dassault Système*. Chaque framework est ensuite subdivisé en entité de granularité plus fine : le module. Un module correspond soit à un exécutable, soit à une Bibliothèque Dynamiquement Linkable.

I.2. Modélisation générale de CATIA

Nous savons maintenant comment sont organisés les fichiers dans CATIA, nous allons donc voir l'architecture générale de CATIA, c'est à dire quels liens existe-t-ils entre les frameworks de base ?



Positionnement des principales framework de l'architecture de CATIA V5

Le schéma ci-dessus représente le positionnement des principaux framework dans CATIA. On remarque rapidement que le framework System se détache du reste, car il est à la base de

l'environnement CATIA. En effet il contient toutes les briques de base comme par exemple la gestion des listes, des threads, du temps ou encore des chaînes de caractères….

Ensuite deux parties se distinguent : l'Object Modeler et la partie User Interface.

La partie interface utilisateur ou plus simplement la fenêtre CATIA, permet de faire le lien entre les objets représentés en mémoire par des SpecsObjects est les interactions effectuées par l'utilisateur sur leur(s) représentation(s) graphique(s).

Le deuxième partie est l'Object Modeler qui contient les classes définissant les documents, les containers, les liens inter-documents… Mais qui fournit aussi des mécanismes puissants pour gérer un modèle objet avancé.

I.3. Une modélisation objet avancée

CATIA v5 a été conçu pour fonctionner sur différents systèmes d'exploitation tels que Windows NT et UNIX. Aux vues de cette hétérogénéité, *Dassault Systèmes* a donc dû développer sa propre technologie objet inspirée de COM.

Cette technologie appelée ObjectModeler est compatible avec COM (sous Windows NT, COM sert de base à ObjectModeler), bien que ObjectModeler possède de nombreuses fonctionnalités de plus que COM, qui facilitent notamment l'intégration des nombreux produits prévus autour des composants d'infrastructure.

Contrairement à COM, il est possible, à tout moment, d'agréger des extensions à une implémentation d'une instance. Ces extensions permettent alors:

- D'étendre dynamiquement le comportement en proposant l'accès à de nouvelles interfaces.
- Elles permettent également d'enrichir dynamiquement les données de cette instance (ce qui nécessite, entre autres, des mécanismes de synchronisation entre les instances et les données).
- Comme ces extensions peuvent être définies dans d'autres bibliothèques partagées que celle de l'implémentation de l'instance modifiée, il est possible d'étendre quasi à volonté n'importe quel composant.

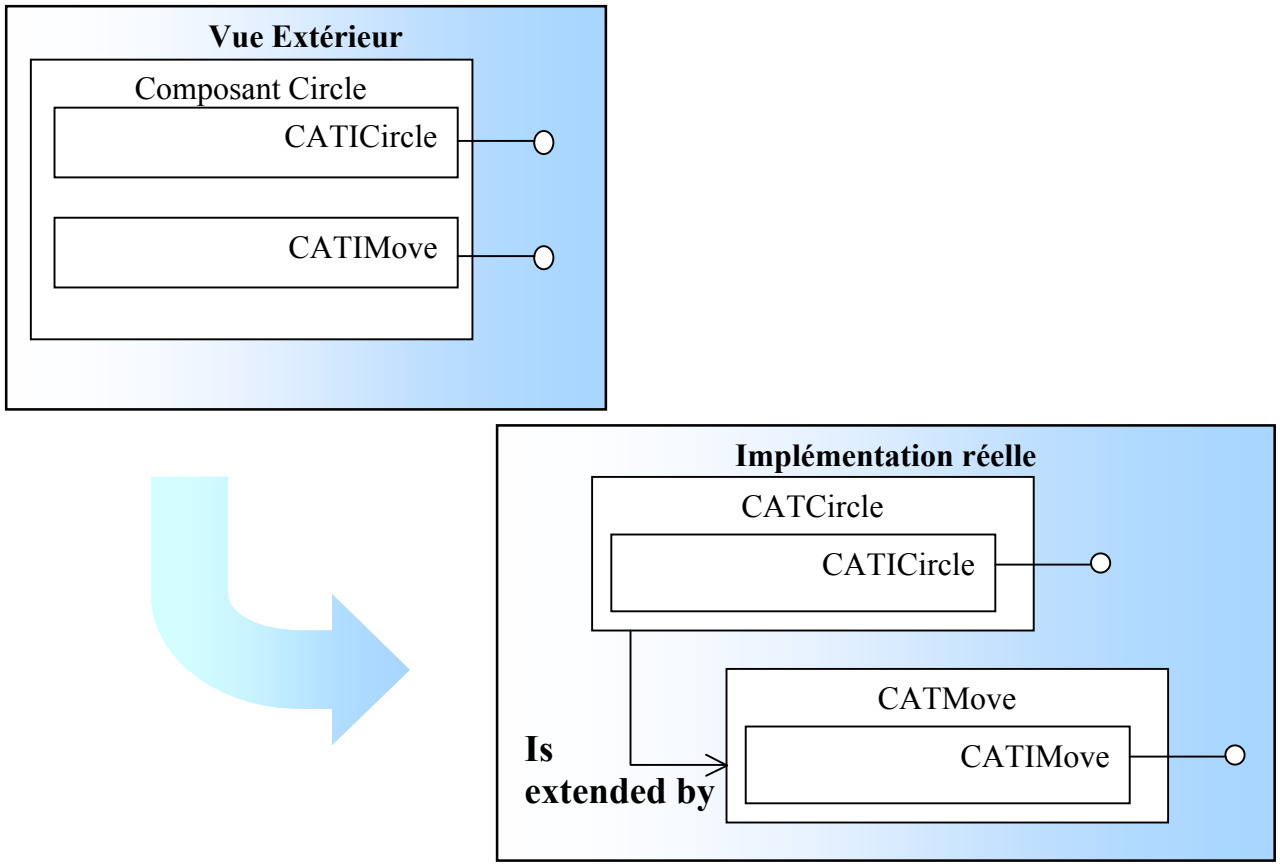
Afin de bénéficier de ces capacités, tout en restant compatible avec COM, la classe de base de la technologie COM IUnknown fut étendue pour satisfaire ce besoin.

Voici un exemple illustrant l'utilisation de cet technique. Lorsque l'on a besoin de dessiner un cercle, on utilise le composant « Circle Factory » qui crée une instance du composant (aussi appelé feature) « Circle » et retourne un pointeur vers une « CATICircleInterface ». Si on a besoin d'effectuer des opérations sur ce cercle, par exemple le déplacer, on invoque la méthode QueryInterface() pour obtenir un pointeur sur l'interface CATIMove de l'instance du compsant « Circle ». Quelle que soit l'opération à effectuer sur un feature, elle s'effectue au travers des interfaces.

L'implémentation d'un tel feature est très différente de la vue extérieure que l'on peut en avoir. En tenant compte des avantages apportés par la CAA V5 un composant de ce type se découpe de la manière suivante :

- Une classe d'implémentation, c'est la classe principale du composant.
- Des classes d'extensions de cette dernière, chacune de ces classes implémentant une ou plusieurs interfaces.

Le schéma ci-dessous illustre cette différence :



Vue Extérieur et modélisation ObjectModeler

La souplesse du modèle Objet CAA v5 et sa modularité sont dues à ces extensions. De plus cela permet de construire ou de modifier les composants sans recompiler les applications des clients.

Une extension d'une classe CAA v5, est une classe séparée de la classe principale du composant, et qui l'étend en implémentant d'autres interfaces qu'elle ne pouvait pas exposer, elle permet d'ajouter des propriétés ou des comportements.

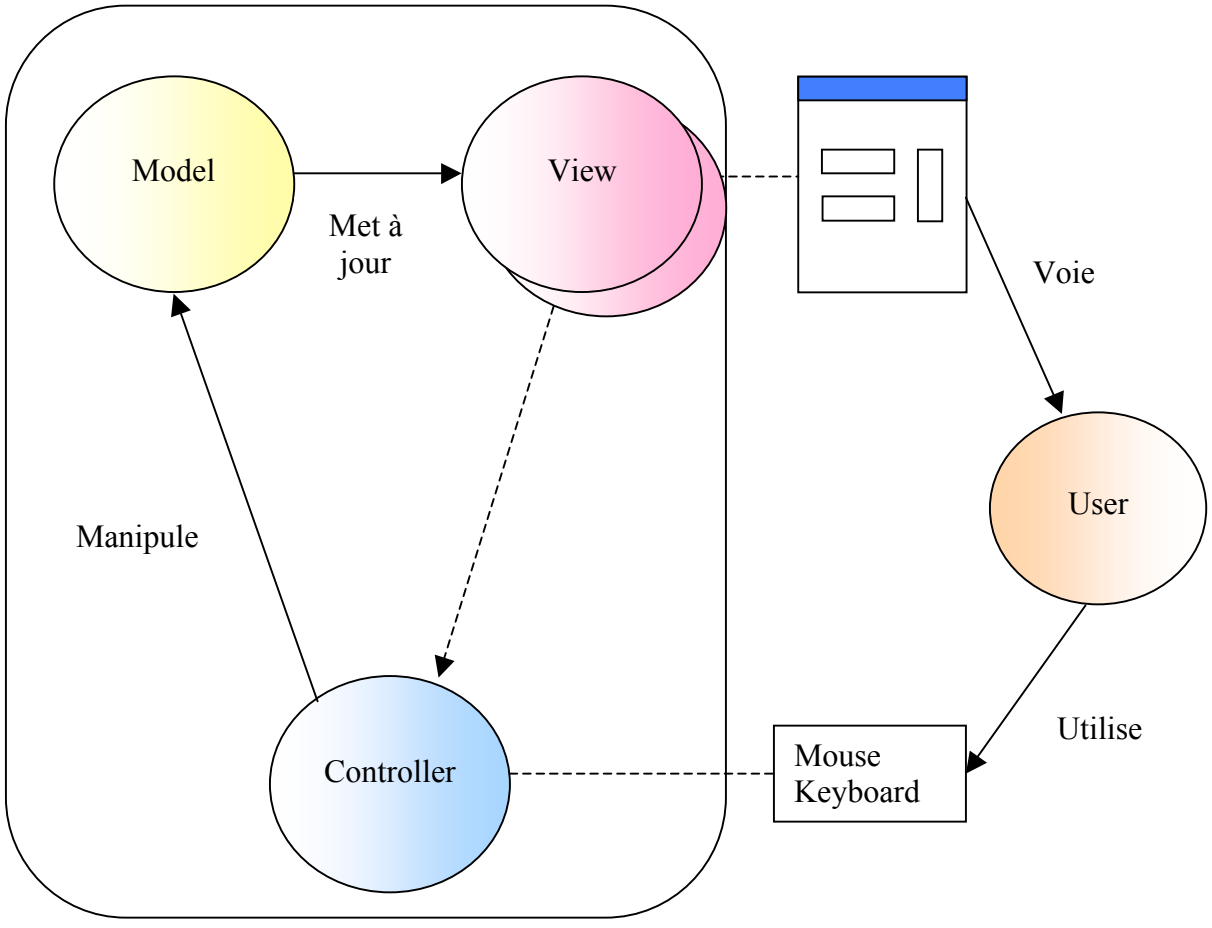
La relation entre une extension et la base du composant s'effectue à travers un dictionnaire.

II. MODÉLISATION MODEL-VIEW-CONTROLLER

D'autres évolutions ont été apportées à la version 5 de CATIA. En effet la partie graphique a aussi été modifiée, dorénavant une modélisation à agents de type Model-View-Controller est utilisée.

I.1 Quelques rappels sur MVC

Dans Smalltalk-80™ l'interface utilisateur était basée autour d'un framework plus connu sous le nom de Model-View-Controller ou MVC. Cette architecture repose sur des agents ayant un rôle bien défini.



Modélisation Model-View-Controller

Le model MVC est très simple, mais aussi incroyablement efficace. Le principe fondamental sur lequel repose MVC est la division de l'application en fonction de trois modules :

- **Model :** Le cœur de l'application. Il maintient l'état courant et les données de l'application. Quand des changement significatifs on lieu il répercute ces modifications sur ces **vues**.
- **Controller :** La partie interface utilisateur manipulée par l'utilisateur.
- **View :** La partie interface utilisateur qui renvoie les informations sur le **model**. Toutes **views** désirant des informations provenant du **model** doit impérativement être enregistré auprès de celui-ci.

De plus il est tout à fait possible d'avoir plusieurs vues rattachées à un model, ce qui peut être pratique quand on veut rendre compte d'informations de manière différente, comme par exemple l'affichage d'une température en degré Celsius et Fahrenheit.

I.2 MVC dans CATIA

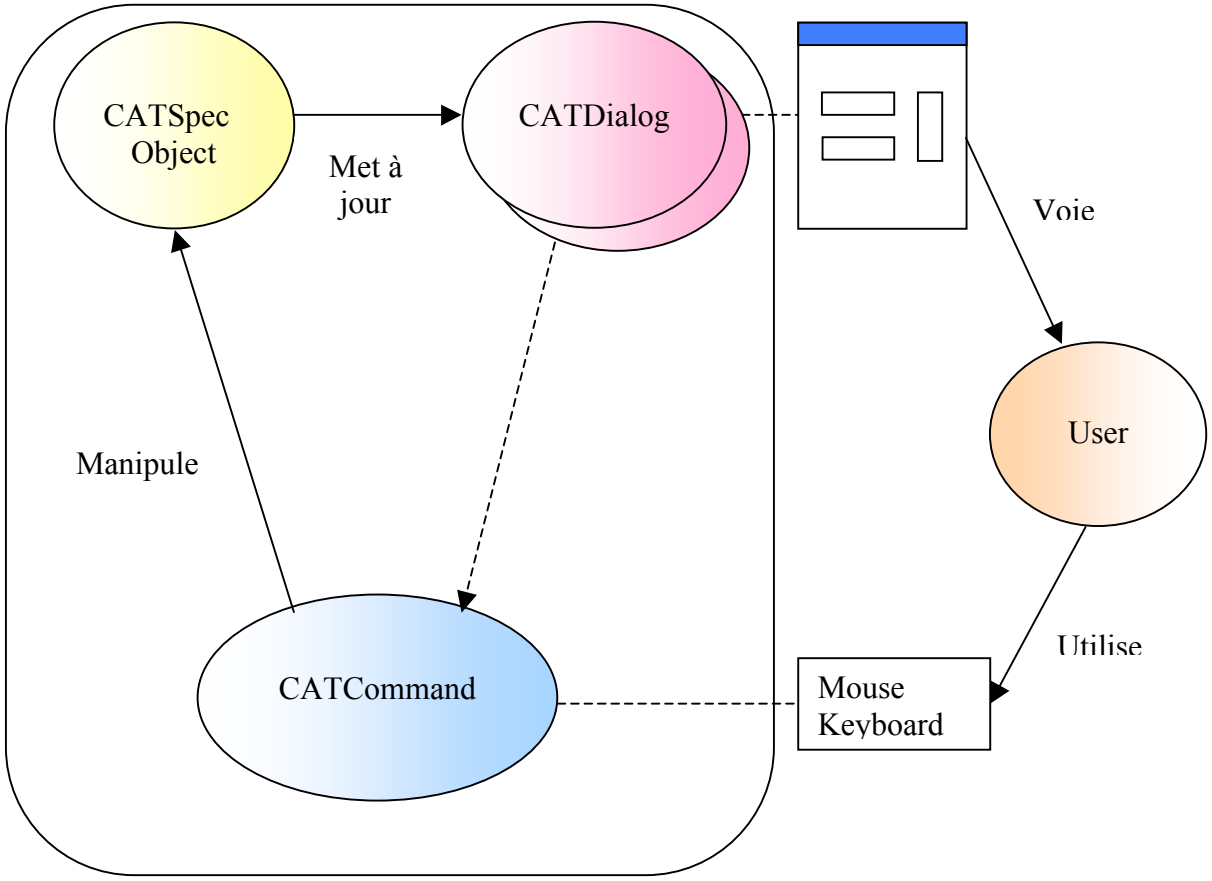
La grande interactivité proposée par l'architecture de CATIA requiert un mécanisme d'interaction gérer efficacement. Ce mécanisme doit permettre de :

- Proposer à l'utilisateur final des outils appropriés pour lancer les tâches qu'il désire effectuer, et ceci par le biais d'accès multiples.
- Réaliser une séquence de dialogue appropriée pour faire et défaire toutes actions simples ou complexes.
- Gérer l'activation de commande par simple clique d'un utilisateur.
- Gérer l'accession aux documents et leur mise à jour.

Les deux dernières propositions illustre une partie du paradigme MVC. L'architecture de CATIA met à disposition un objet qui apporte ces mécanismes : la CAA Command. C'est la clé de voûte qui rend le modèle d'interface utilisateur de CATIA interactif.

Dans CATIA le design pattern MVC est implémenté de la manière suivante :

- CATDialog : représente la partie interface utilisateur. Cette partie correspond en fait à la vue du modèle MVC.
- CATCommand : il représente la partie contrôle. Toutes les interactions utilisateurs effectuées sur l'interface graphique sont centralisées ici.
- CATSpecObject : Cette partie correspond au modèle. Le *CATSpecObject* représente généralement l'abstraction d'un objet graphique.



Modélisation Model-View-Controller dans CATIA

I.3 MVC et les services Peer to Peer

Dans le cas des services Peer-to-Peer développés, le modèle MVC est appliqué comme suit :

- La partie **Model** est constituée du service peer-to-peer. Il s'agit de la partie de l'application qui traite les données. Elle fournie des services à la partie Controller comme l'envoi de données vers un peer, la demande de données d'un peer ou encore la notification de modifications effectuées sur les données... Cette partie s'appuie essentiellement sur le noyau peer-to-peer qui lui fournit les briques de base pour la communication inter-peer.
- La partie **Controller** sert de lien entre l'utilisateur et l'application. Elle réagit aux actions de ce dernier pour effectuer les interactions nécessaires avec la partie Model en vue d'utiliser ses données.
- La partie **View** se contente de proposer des services pour réaliser des mises à jour de l'interface utilisateur.

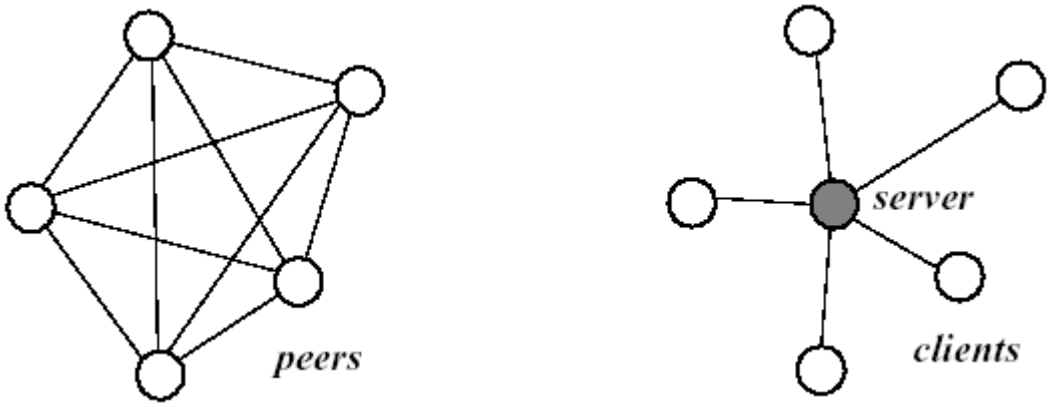
La plate-forme Peer to Peer

I. LA TECHNOLOGIE PEER TO PEER

I.1. Définition

Plusieurs définitions du Peer to Peer existent selon le groupe de travail dans lequel vous êtes, mais une définition générale semble ressortir : le terme Peer to Peer (abrégé par P2P) désigne une classe de systèmes et d'applications basées sur le principe des ressources distribuées pour effectuer une fonction critique de manière décentralisée.

On peut préciser cette définition(un peu trop générale) en disant qu'une application est Peer to Peer quand elle propose des services aussi bien en tant que client, que serveur. En effet ceci devient compréhensible quand on cherche à traduire peer : en français ce terme donne pair(dans le sens de jumeau). L'expression équivalente de Peer to Peer devient alors Pair à Pair(mais la formulation d'égal à égal est aussi très utilisée).



P2P versus Client/Serveur

Comme nous le montre la figure ci-dessus une architecture client/serveur est essentiellement axée autour d'un serveur, ce qui implique une grande dépendance des clients envers celui-ci. Mais cette technologie offre aussi une grande facilité d'administration. A contrario le Peer to Peer propose une plus grande mobilité du fait que chaque pair est autonome, et du même coup réduit ça sensibilité aux pannes. Mais survient un nouveau problème : comment retrouver une grappe de machines mobiles avec des adresses IP dynamiques ? (Ce problème ne se posait pas avec un environnement client-serveur classique)

Pour résoudre ce problème il existe deux philosophies :

- L'annuaire : il permet de fournir une adresse qui sera notre point d'entrée dans la grappe (mais donne aussi la possibilité d'organiser la grappe). Mais cette solution s'oppose au paradigme du Peer-to-Peer du fait de l'utilisation d'un serveur.
- La découverte : utilisation d'un protocole qui va rechercher d'autres peers. En générale ce type de technologie utilise des protocoles basés sur du broadcast/multicast.

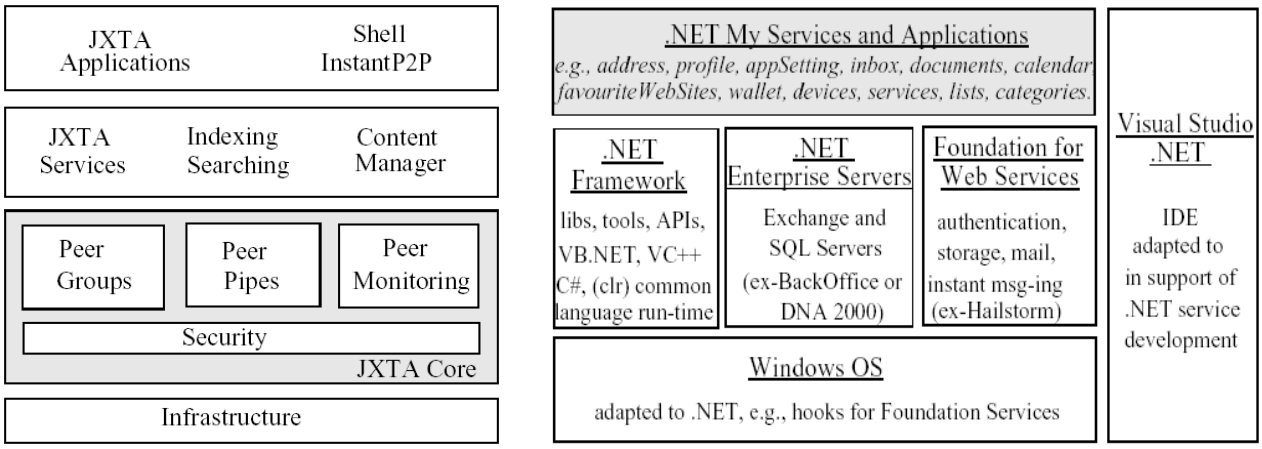
I.2. Exemples d'applications

Des applications utilisant les principes du peer to peer existent déjà. En générale il s'agit d'applications de partage de fichiers telles que *Gnutella*, *Free Haven* ou encore *Mojo Nation*, mais il en existe d'autres telles que *FreeNet* ou *Groove* qui permettent en plus du partage de fichiers, d'échanger des messages ou de jouer en réseau.

I.3. Les frameworks Peer to Peer

Récemment beaucoup de plates-formes de développement d'architecture Peer to Peer ont vu le jour. Les plus connues sont JXTA de Sun Microsystem et .NET de Microsoft. Les principaux intérêts de ces frameworks sont :

- L'interopérabilité : en facilitant la localisation des peers interconnectés, en permettant de participer à des activités communautaires ou encore d'offrir des services(essentiellement partage de ressources du style fichier).
- Portabilité : ces plates-formes sont designées pour être indépendantes du langage de programmation (C/C++, Java ou encore C#). Mais aussi du système d'exploitation (Windows, MacOS, Un*x).
- Ubiquité : leur implémentation est possible du PC de bureau au PocketPC.



JXTA versus .NET

Malgré des différences entre les protocoles de communications utilisés par .NET et JXTA (respectivement SOAP et PRP/PDP/PIP/PBP/ERP/RVP), on remarque de très nettes ressemblances entre les deux architectures présentées ci-dessus : un corps(couche de communication), des services de base(recherche, gestion des peers), et des services spécialisés.

II. PRESENTATION DE LA PLATE-FORME PEER TO PEER

II.1. De l'individualisme à la collaboration

Comme indiquait dans la présentation de Dassault Systèmes, CATIA permet facilement de créer, simuler, valider et la maintenir un produit. Mais il manquait une fonctionnalité importante à ce logiciel : la possibilité de pouvoir l'utiliser dans un environnement distribué. Et ainsi passer de l'ère du travail individuel à celle de la collaboration.

C'est dans cette optique qu'a été choisie de développer une plate-forme distribuée de type Peer-to-Peer.

II.2. Pourquoi une nouvelle plate-forme ?

En effet des plates-formes Peer-to-Peer sont déjà disponible sur le marché, pourquoi ne pas en avoir choisi une déjà existante. Plusieurs raisons essentielles à cela :

- De mauvaises performances(JXTA)
- Toujours en développement(.NET) et propriétaire
- Intégration dans CATIA(langage C++)
- Portabilité entre les différents Unix et Windows
- Passage à l'échelle

Ces raisons ont fait qu'il a été plus simple d'en développer une nouvelle, ayant une architecture similaire à celle déjà existante, mais directement intégrer à l'environnement de développement CAAv5.

II.3. Architecture

Cette plate-forme se veut autonome, une architecture totalement décentralisée a donc été choisie. En effet un protocole de découverte est utilisé pour trouver les autres peers. Ce qui induit le problème du choix de l'identification : comment être sur que tous les peers aient un identifiant unique? Pour résoudre se problème, la plate-forme génère automatiquement un identificateur grâce au *login* de l'utilisateur, au *nom de la machine* sur laquelle CATIA s'exécute et à la *date courante*(très utile dans le cas ou plusieurs peers sont présents sur la même machine). Ce système procure aussi un autre avantage, celui d'abstraire, pour la couche applicative, les informations liées au réseau(adressage, protocole utilisé,…..).

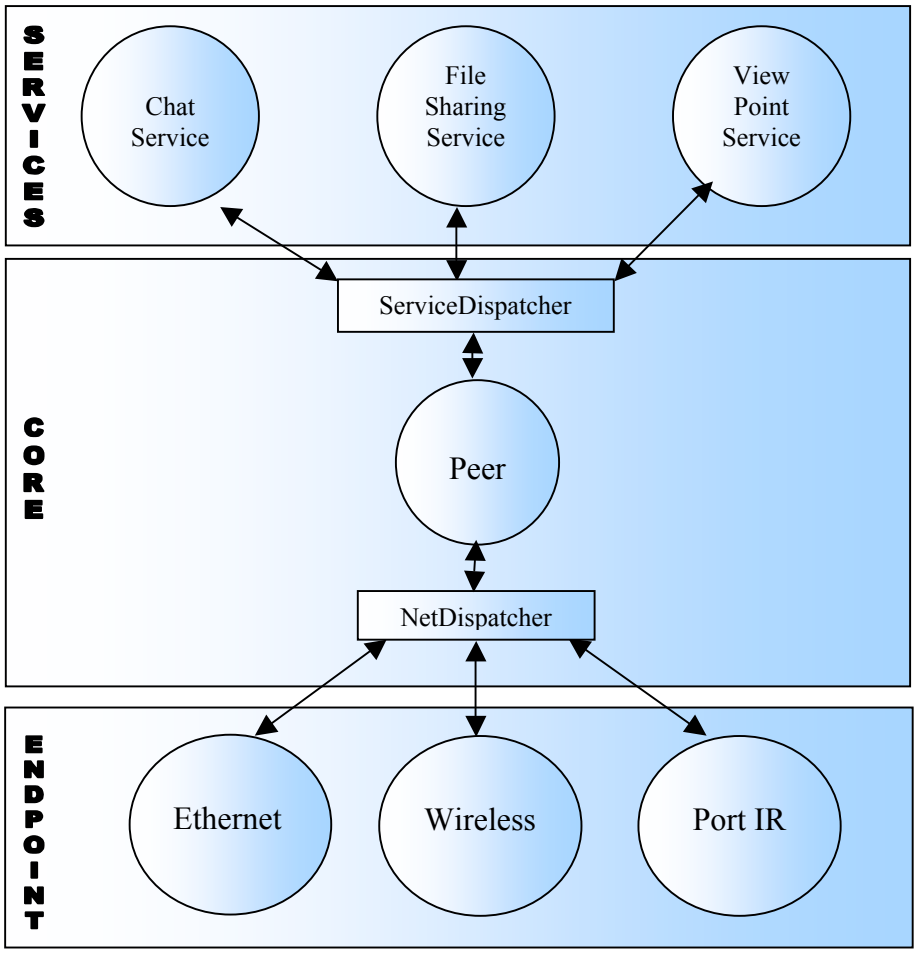
Une autre notion importante a aussi été apportée à la plate-forme *P2P* : il s'agit de la notion de groupe. En effet lorsqu'un grand nombre de pair se trouve sur le réseau il est plus aisé de les rechercher dans une arborescence que dans une liste peu équivoque. Cette technique a des répercussions sur le système de communication, en effet lorsqu'un peer est dans un groupe donné il reçoit les messages provenant des groupes parent, mais pas des sous-groupes. Ce qui a pour conséquence de décharger les services en effectuant un traitement d'élimination des paquets dans les couches plus basses de la pile *P2P*, ce qui entraîne une certaine économie de ressources(CPU et mémoire).

La plate-forme Peer to Peer est scindée en trois parties distinctes :

- **Le Endpoint :** cette couche est le lien entre le réseau est le cœur de la pile Peer to Peer. C'est à ce niveau que le tri des messages est effectué. Chaque EndPoint est enregistré auprès du NetDispatcher qui est chargé d'aguiller les messages vers la bonne sortie.

- **Le Peer :** cette partie est le cœur du système, c'est ici qu'est effectuée la gestion des voisins et des groupes.

- **Le Service :** il s'agit de la partie applicative qui utilise le noyau peer to peer. Le service s'enregistre(le type des messages/streams attendus) auprès du ServiceDispatcher qui est chargé de remonter les messages vers le service destinataire.



Architecture générale de la plate-forme Peer to Peer

II.4. Implémentation

La plate-forme de développement Peer to Peer est intégrée à CATIA sous la forme de deux modules ajoutés au framework « ApplicationFrame » :

- **CATP2Pcore** **:** implémentation du noyau Peer to Peer (méthodes d'accès au « réseau »).
- **CATP2Pcommand :** implémentation des services, comme par exemple un forum de discussion ou un service de partage de fichiers.

Les protocoles utilisés pour l'implémentation sont :

- **TCP :** pour la communication entre deux peers sous forme de stream ou pour l'expédition d'un message vers un peer donné.

- **UDP(Multicast) :** pour l'envoi de messages vers un group.

Mais il est tout à fait possible de les changer sans modifier les applications, en effet l'utilisation des identifiant rend l'utilisation de ces protocoles transparente pour celles-ci.

Le protocole *TCP* fut choisi pour d'évidentes raisons : sa fiabilité et son efficacité. Il est utilisé pour la communication directe entre deux peers (*1 vers 1*). Pour implémenter les communications de type *1 vers N*, le choix c'est porté sur le *Multicast* du fait de ses performances. Mais il faut garder en tête que son utilisation pose certains problèmes, comme le passage des firewalls ou des routeurs (normaux ou NAT). Néanmoins un palliatif de ce problème existe : le développement d'un service de routage qui permettrait la circulation des messages et flux au-delà de ces barrières. Un autre désavantage est à mettre en évidence, il s'agit de la possible perte de paquets due à la non fiabilité du protocole UDP, il faut donc utiliser les primitives d'envoi de messages vers un groupe avec précaution (exemple : s'en servir uniquement pour l'envoi d'évènements).

Travail effectué

Ma mission a pour but de créer des applications distribuées au-dessus de la plate-forme Peer-to-Peer. Ces développements ont pour finalité le test de faisabilité des applications, et la connaissance de l'intérêt qu'apporteraient de telles fonctionnalités à CATIA en vue d'une intégration produit.

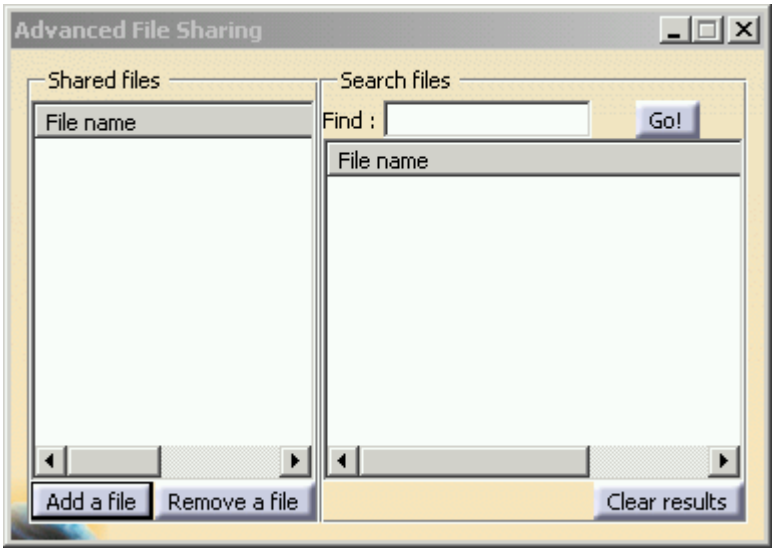
I. PARTAGE DE FICHIERS

Il s'agit de la première application que j'ai ajoutée à CATIA. Elle se décompose en deux parties : une première qui s'occupe de lister les fichiers que l'on désire partager, et une seconde de la recherche de fichier.

I.1. Interface graphique

L'application est activable par le biais d'une icône présente sur une barre d'outil :

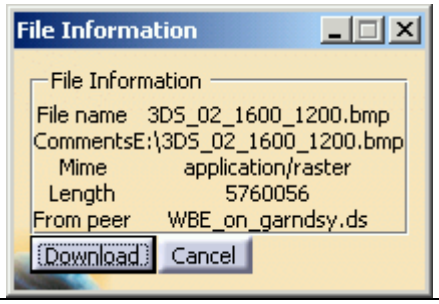
Voici son interface graphique :

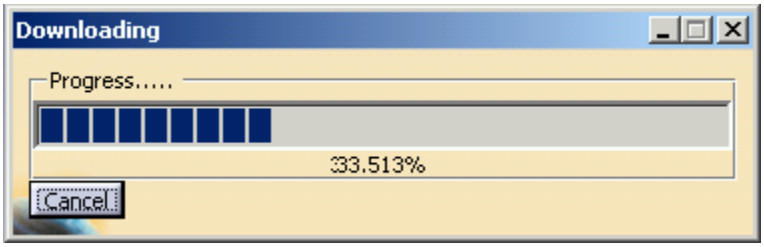


Ci-contre la fenêtre principale de l'application. Celle-ci propose clairement les deux fonctionnalités suivantes :

- le partage
- la recherche

Cette fenêtre s'affiche lorsque l'utilisateur sélectionne un des résultats de la recherche, lui donnant ainsi des informations pratiques sur celui-ci, tout en lui offrant la possibilité de le télécharger.



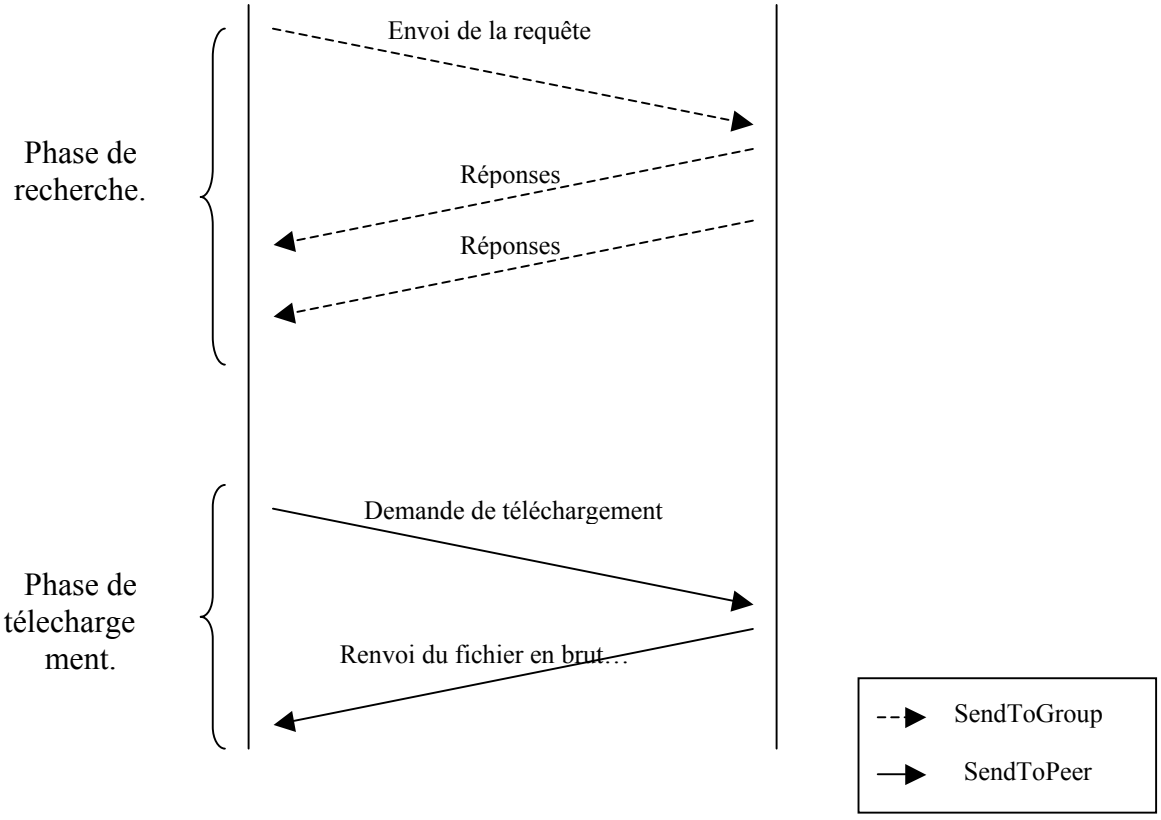


Ci-contre la fenêtre de téléchargement qui rend compte de l'état d'avancement du download, et autorise son annulation.

Cette fonctionnalité se veut aussi compatible avec la version existante sur le portail java, ce qui implique donc l'usage du même protocole de communication, présenté ci-dessous.

I.2. Le protocole de communication

Voici un scénario typique d'utilisation de l'application. L'utilisateur rentre une expression régulière, puis lance la recherche. Des résultats lui sont ensuite renvoyés. Il sélectionne un fichier et décide de le télécharger.



Scénario typique de recherche de fichiers

Pour mettre en œuvre ce protocole deux types de messages et un type de flux ont été déclarés :

- Messages :
  - “QUERY_SERVICE” : envoi de la demande.
  - “QUERY_RESULT_SERVICE” : envoi des réponses.
- Stream :
  - “DOWNLOAD_SERVICE” : lance le téléchargement d’un fichier.

Le protocole de cette application est très simple. Un peer désirant effectuer une recherche envoie à tout son groupe un message de type QUERY_SERVICE contenant la requête à traiter. Ensuite chaque peer recevant ce message le traite : il recherche dans sa liste de fichiers partagés les fichiers dont le nom concorde avec l’expression régulière envoyée. Ensuite si le nombre de résultats n’est pas nul le peer répond en renvoyant au demandeur une liste de fichiers dans un message du type QUERY_RESULT_SERVICE.

Le fait que l’application ne supporte pas les recherches multiples nous évite l’ajout d’une estampille pour discriminer les différentes requêtes. Pour vérifier si le résultat reçu est valide une simple comparaison entre la requête courante et celle renvoyée dans les réponses suffit.

Lors du processus de téléchargement l’application ouvre un stream vers le peer possesseur du fichier, puis lui envoie les informations relatives à celui-ci(son nom...), ensuite l’autre peer lui répond : soit en lui renvoyant le contenu du fichier, soit par une erreur.

L’architecture pourtant simple de cette application m’a permis de faire un tour d’horizon de la plate-fome CNEXT, et ainsi de mieux appréhender les aspects suivants :

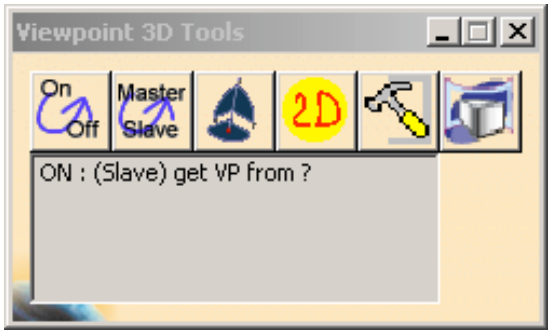
- Programmation graphique : utilisation du framework Dialog, modélisation Model-View-Controller.
- Les Threads dans CATIA.

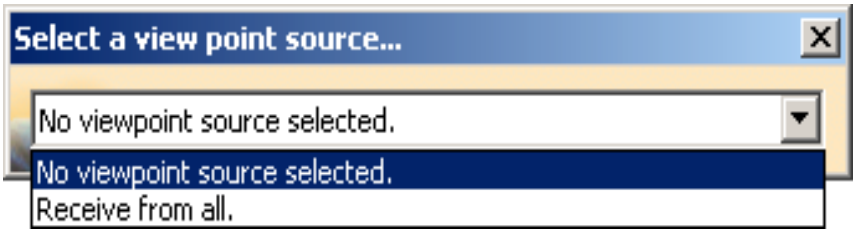
II. PARTAGE DU POINT DE VUE ET ECHANGE D'ANNOTATIONS

Cette application a pour but le développement de manière collaborative de produits grâce à CATIA. Pour ce faire l'application dispose de deux services. Le premier permet d'émettre le point de vue d'un peer. Le deuxième sert à partager des vues annotées, et à les modifier en librement.

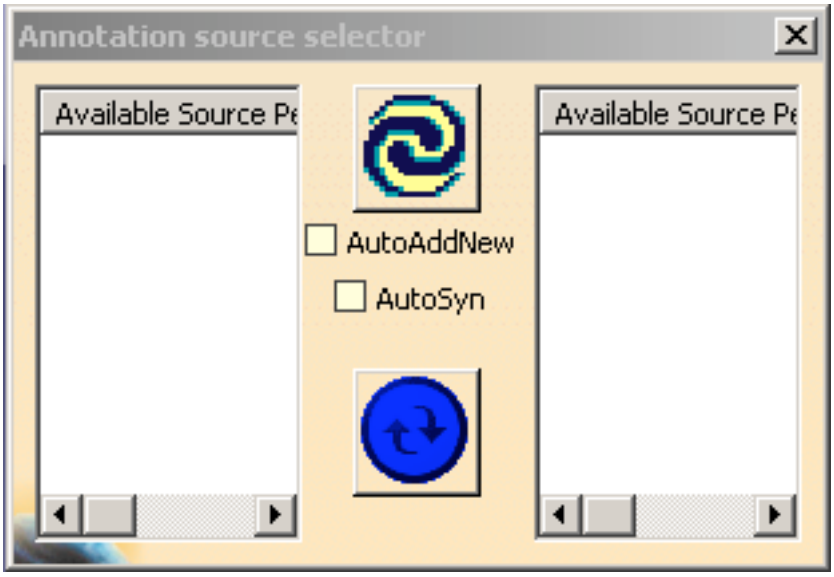
II.1. Interface graphique

Ci-contre la boite de dialogue permettant d'activer le service de partage du point de vue, et celui du partage des annotations.

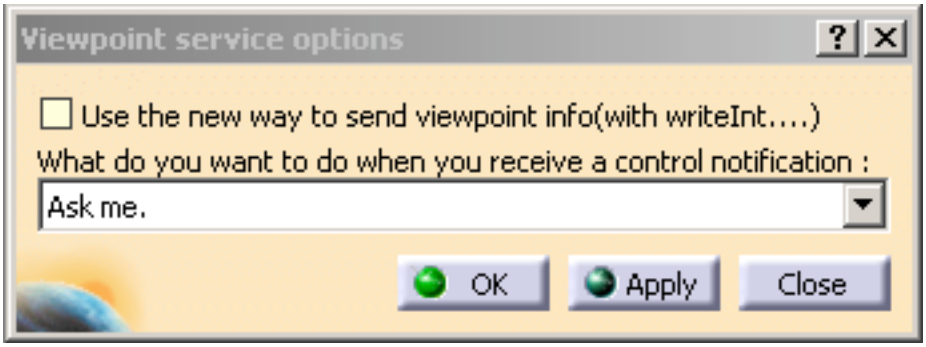




Lorsque l'outil de partage de point de vue est activé, il est possible de choisir la source émettrice. Plusieurs choix sont possible : soit aucune source(donc je reste maître de mon viewpoint), soit tous les peers, soit un peer précis.



Jumelé à l'outil de partage de point de vue, celui de partage des annotations. Grâce à cette boite de dialogue il est possible de choisir les sources émettrices d'annotations et de se synchroniser avec elles. Mais s'y l'on désire qu'un peer entrant dans notre groupe devienne automatiquement une source, il faut cocher l'option « Auto add new ». De la même manière il existe la possibilité de s'auto-synchroniser avec les nouveaux peers, grâce à l'option « Auto synchronise »



Options disponibles avec le service. (Que faire en cas de demande de contrôle de la part d'un PocketPC, choix de la méthode d'envoi)

II.2 Modélisation du service

Ce service est en fait scindé en deux services Peer to Peer :

- Le Viewpoint Service : qui gère le partage du point de vue.
- Le Macro Service : qui s'occupe de l'échange des annotations.

II.3. Le service de viewpoint

Le but de ce service est de faciliter une revue collaborative en permettant à tous les peers de voir le même objet sous le même point de vue. L'implémentation de ce service été réalisée par mon maître de stage : Raphaël LEBLANC. Mon travail consista à doter ce service de fenêtres de configuration et de lui joindre un service de partage de vues annotées.

Dans un premier temps une boîte de dialogue pour la sélection du peer émetteur fut écrite. Son fonctionnement est simple, à chaque fois qu'un nouvel émetteur est détecté, il est rajouté dans la liste. De la même manière lorsqu'on remarque sa sortie on le retire de la liste.

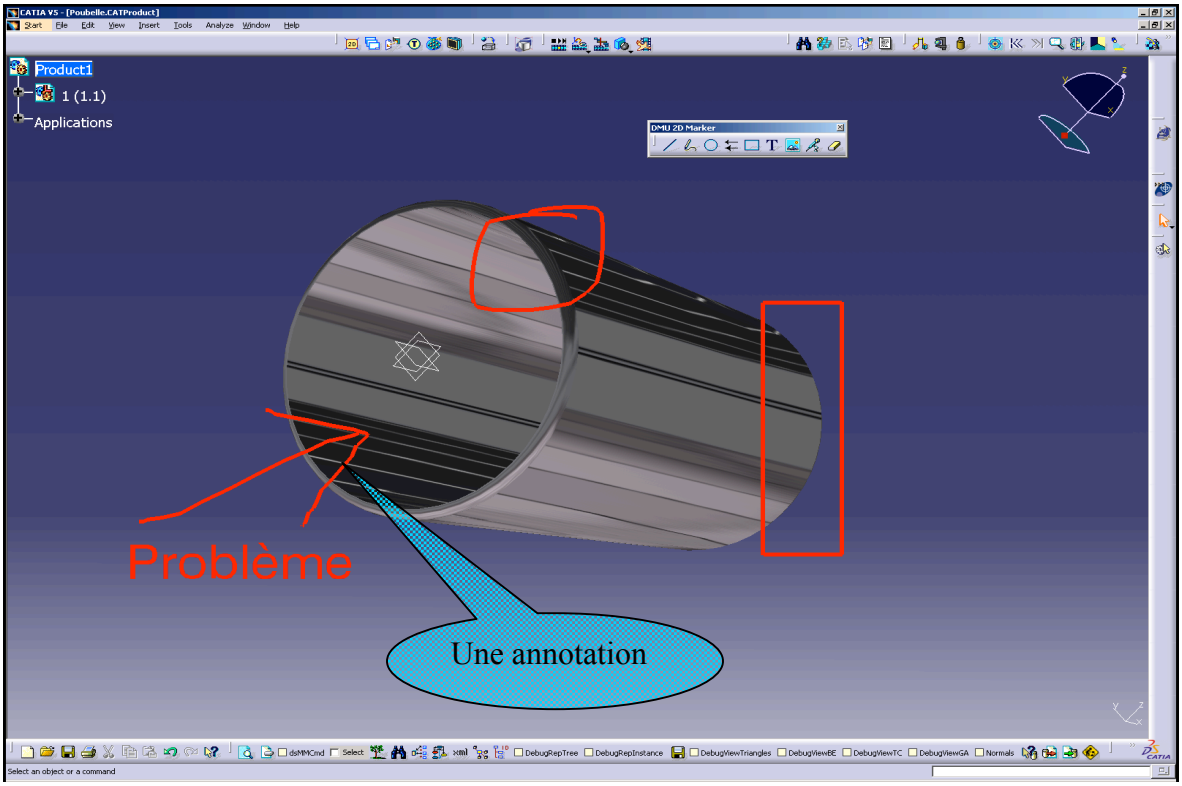
Ensuite j'ai rajouté un comportement différent au service dans le cas d'une demande de contrôle de la part d'un périphérique du type PocketPC. En effet le service développé sur pocket Pocket PC n'a pas tout à fait le même fonctionnement que sur PC : c'est le PPC qui choisit qui il dirige. C'est pour cela qu'une pop-up, qui demande si l'on veut se faire diriger, a était implémenté.

- Messages :
  - "VIEWPOINT_NEW_WAY" : envoi du viewpoint en brut (un peu plus rapide).
  - "VIEWPOINT_3D_SERVICE" : envoi du viewpoint sous la forme d'une chaîne de caractères.
  - "VIEWPOINT_3D_SERVICE_PPC" : envoi du point de vue de la part d'un PocketPC.

Ces messages (selon la configuration choisie) sont émis à chaque changement de point de vue sur l'émetteur. Le travail du récepteur est assez simple : il décode les données et met à jour son viewpoint.

II.4. Le service d'échange d'annotations

Ce service autorise le partage de vues annotées entre Peer. La copie d'écran ci-dessous nous montre ce que sont les annotations dans CATIA.



L'implémentation de cet applicatif est basée sur une philosophie très simple : un peer n'a le droit de garder que des informations qui le concerne, et ne doit pas publier d'informations concernant d'autres peers. Ces règles simples sont utilisées pour maintenir une forte cohérence des informations sur la grappe.

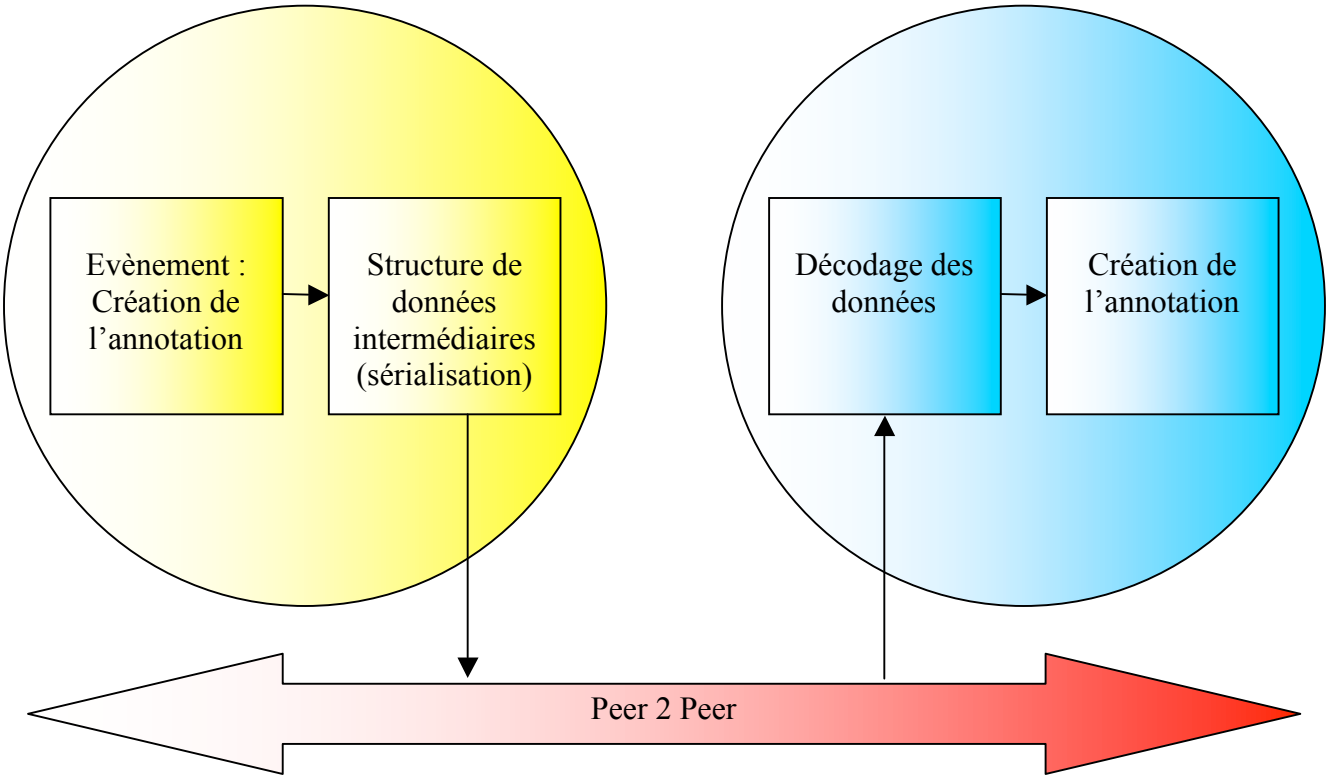
II.4.1. Les annotations

Il faut savoir qu'un marker peut prendre plusieurs forme : cercle, ligne, rectangle…. Et que chaque marker est rattaché à une vue donnée. Donc sur une machine chaque couple (ViewName, MarkerName) est unique, mais ce n'est pas le cas dans un environment Peer to Peer. C'est pour cela que l'on concatène le nom du peer au nom du marker, ce qui permet de rendre chaque couple unique sur la grappe.

(ViewName, MarkerName) ==> (ViewName, PeerName.MarkerName)

Nous sommes maintenant capables d'identifier une annotation de manière unique. Il faut maintenant les faire transiter entres les peer. Au vue de la complexité d'une annotation (multitude d'informations, polymorphisme), j'ai décidé de créer une structure de donnée intermédiaire permettant de stocker les informations nécessaires à la création ou à la modification d'une annotation. Ces informations seront sérialisés pour pouvoir être envoyées.

Le schéma ci-dessous montre de quelle manière les annotations sont envoyées aux autres peers.



Architecture de l'échange d'annotations

Cette structure de donnée contient les informations suivantes :

- Nom de la vue
- Nom du marker
- Nom du possesseur.
- Définition de la vue : Point de vue
- Définition du marker, informations nécessaires pour le créer : type, position…

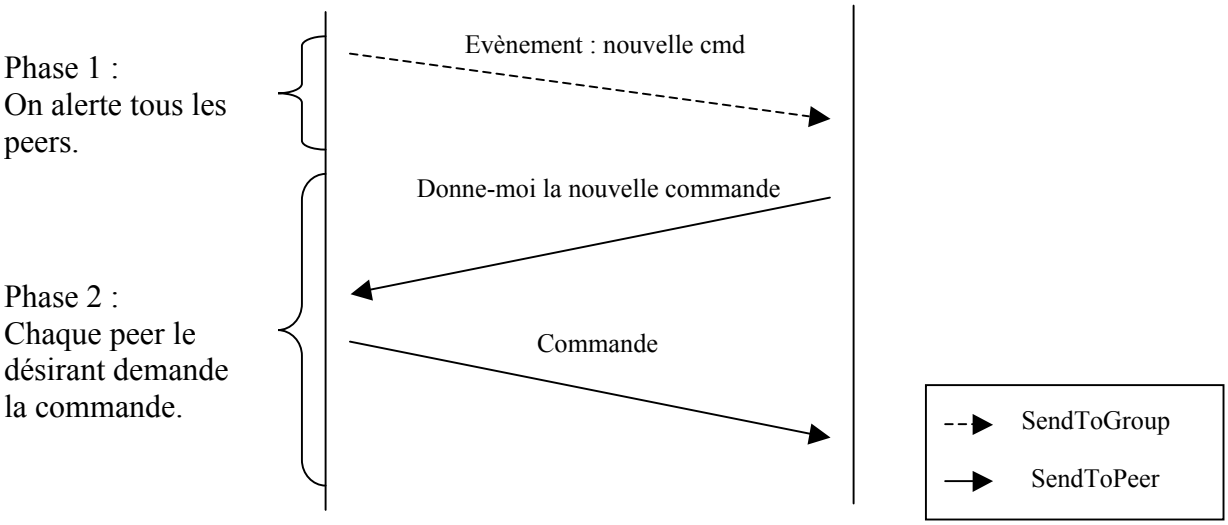
L'évènement qui nous averti de la création d'une nouvelle annotation est générée par le journal. En effet CATIA est doté d'uns système évolué de logging qui permet entre autre la gestion de l'undo ou l'enregistrement de macro. C'est sous la forme d'une macro Visual Basic que les informations liées aux annotations vont-nous êtres délivrés. Ensuite un parsing rudimentaire de ce script nous permettra facilement de les extraire et de remplir la structure de données.

Inversement une fois arrivée sur le peer distant la structure de donnée va être utilisée pour générer un script et ainsi créer ou modifier une annotation.

II.4.2. Le protocole d'échange

Le protocole d'échange doit prendre en compte certaines contraintes de le plateforme Peer to Peer, tel que l'impossibilité d'envoyer des messages de plus de 2ko. La taille de la structure intermédiaire pouvant être variable il était impossible d'envoyer l'annotation directement au groupe.

Le message envoyé vers le groupe aura alors une sémantique légèrement différente : ce sera un événement qui signifiera qu'une nouvelle commande. Ensuite les peers désirant télécharger cette annotation initieront eux même la connexion. Voici le chronogramme de l'échange :



Obtention d'une commande

Une évolution du protocole fut nécessaire pour gérer un historique des commandes. En effet le protocole ci-dessus ne permet de gérer qu'un seul niveau. La gestion de plusieurs niveaux requiert l'identification unique de chaque commande au sein de la grappe Peer to Peer. Pour ce faire nous allons associer un numéro à chaque commande, ainsi le coupe (numéro de commande, nom de peer) sera unique pour chaque commande. Le numéro de commande correspond en fait à une horloge

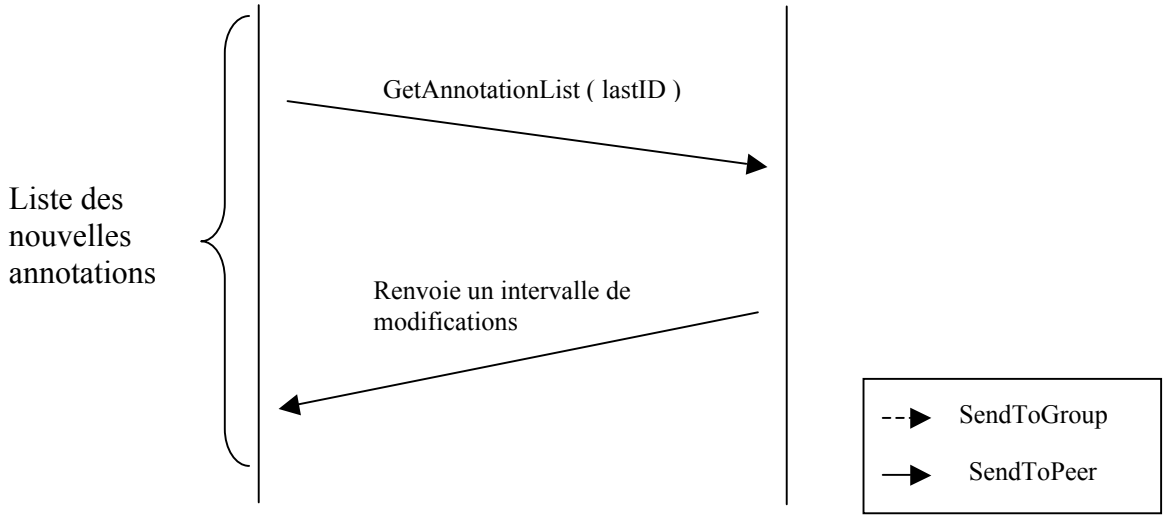
logique, car ce numéro est sans cesse incrémenté. Cette ID est envoyé dans l'événement pour indiquer la dernière commande disponible.

Le peer demandeur doit ensuite communiquer cette ID pour lors télécharger, puis le sauvegarder. Cette ID servira aussi à vérifier que l'on n'est pas en train de télécharger deux fois la même commande ou bien que tous les événements aient été traités.

Voici la liste des messages et streams utilisés :

- Messages :
  - "MACRO_NEW_ANNOTATION" : nouvelle commande disponible.
- Stream :
  - "MACRO_GET_ANNOTATION" : obtention d'une annotation.

L'historique autorise aussi la synchronisation avec les émetteurs. Pour facilité la re-synchronisation une nouvelle méthode a été implémentée : GetAnnotationList, qui prend en paramètre le dernier ID que le peer a émis et nous renvoie ce qu'il y a de nouveau à télécharger.



Obtention de la liste des nouvelles commandes

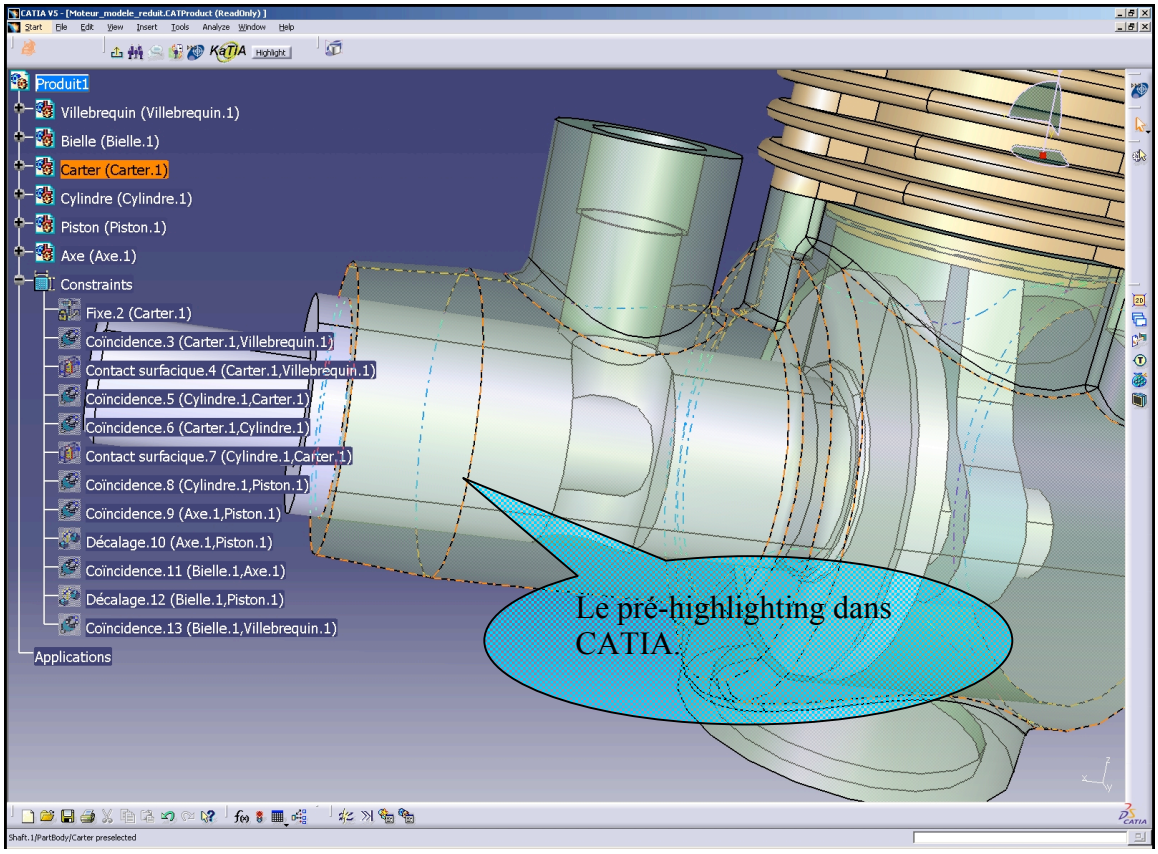
Ces trois méthodes permettent l'envoi de commande c'est à dire à un peer propriétaire d'une annotation de la modifier ou d'en créer de nouvelles. Pour qu'un autre peer ait la possibilité de les modifier ou de les détruire, il faut qu'il puisse prévenir le propriétaire qui sera chargé d'avertir les autres peers des transformations. Cette méthode s'appelle MACRO_MODIFICATION_REQUEST, et prend pour paramètre une commande spéciale. En effet celle-ci n'a pas à être convertie par le

propriétaire, c'est le peer qui effectue la modification qui se charge de faire les traductions nécessaires avec ces tables de correspondances. (et aussi de les maintenir à jour en cas de délétion)

Toutes ces méthodes ont permis de construire un service capable de gérer l'envoi d'annotations, leur modification ou délétion, aussi bien par le peer créateur que par n'importe quel peer de la grappe.

III. LE CO-HIGHLIGHTING

Lors d'une démonstration ou d'une simple réunion il peut être utile de mettre en évidence certaines partie d'un produit. Pour cela CATIA dispose de la fonction de pré-highlighting(PSO), qui surligne d'un trait pointillé la partie de la pièce présélectionnée. Le service de cohighlighting se propose d'utiliser cette fonctionnalité de manière collaborative.



III.1. Gestion de la présélection dans CATIA

Dans l'architecture CATIA il existe trois types de sélections :

- Prehighlighted Set of Objects (PSO)
- Highlighted Set of Objects (HSO)
- Current Set of Objects (CSO)

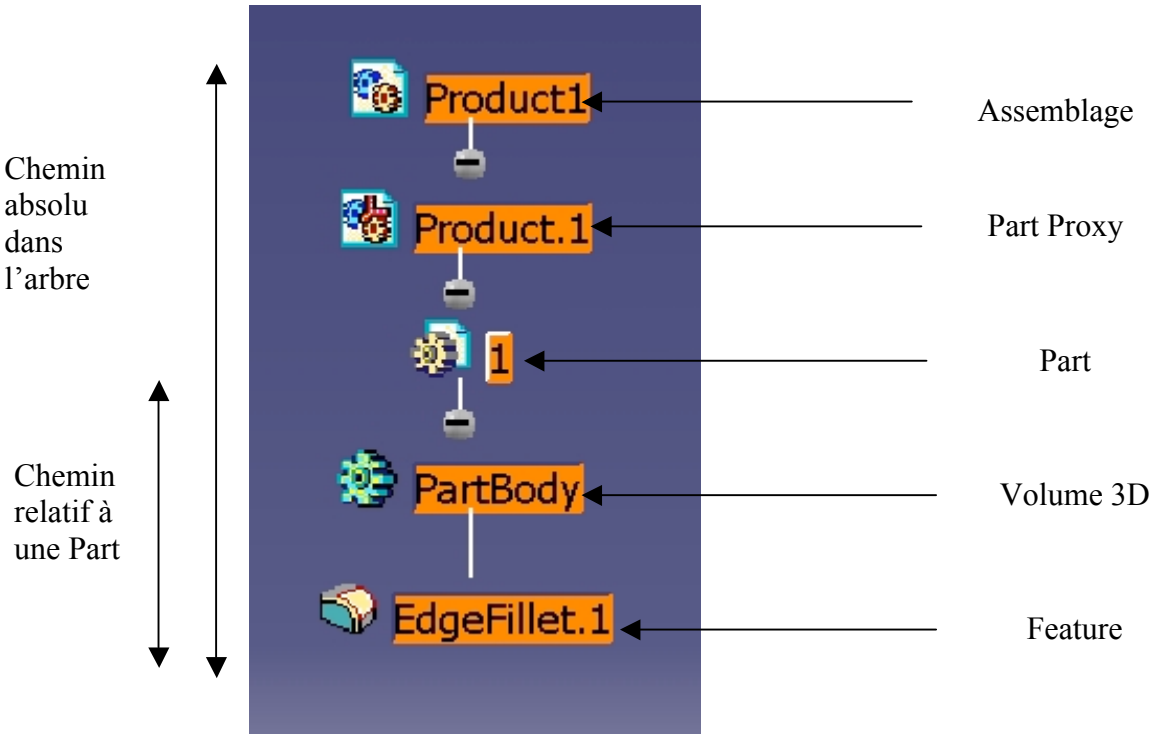
Ces trois types de sélection se différencient de la manière suivante.

Le CSO est la liste des objets courant. Les objets sélectionnées sont placés dans le CSO permettant aux différents commandes de rechercher des objets dans cette liste. Les fonctionnalités tels que le Coupé/Copié - Collé et le Glissé-Déposé l'utilisent. Les objets présents dans cette liste voient leur représentation graphique highlightée.

Le HSO, contient la liste des éléments highlightés. Très similaire au CSO puisque tous les éléments présents dans le HSO le sont aussi dans le CSO.

Le PSO est la liste des objets actuellement manipulés. C'est objets sont soit pré-activés, soit déplacés.

PSO et HSO sont des spécialisations de XSO qui est utilisé par l'interface graphique pour gérer le rendu graphique des divers éléments.

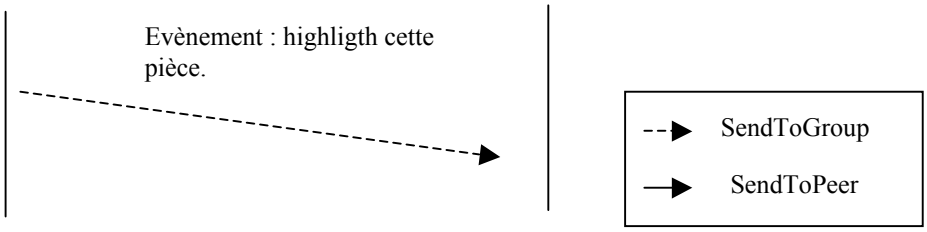


Exemple de chemin dans l'arbre

De plus il faut savoir que chaque feature dans l'arbre est identifié par un Global Unique ID. On peut ainsi aisément substituer les GUIDs aux features dans le chemin absolu. Un autre problème se pose, en effet d'un peer à l'autre la racine du chemin peut varier quelque peu, d'un coté une personne édite une aile d'avion et de l'autre l'avion complet. C'est pour cela que le chemin envoyé est relatif, la nouvelle racine correspond au GUID de la Part. Ainsi lors de l'analyse sur le peer distant, celui-ci doit le compléter pour qu'il devienne absolu.

III.2. Protocole d'échange

Grâce aux différentes analyses sur la gestion des sélections précédemment effectuées le protocole de communication inter-peer ce voit très simplifié. En effet il ne contient qu'un type de message, qui sert à envoyer une liste de GUID « stringuifiés » (le chemin relatif). De plus l'utilisation d'un événement(Multicast) permet d'optimiser la bande passante utilisée.



Emission des données de co-highlight

IV. CHANGEMENT DE CONTEXTE

IV.1. La gestion du contexte dans CATIA

CATIA est une application générale, mais pouvant se spécialiser selon des besoins spécifiques. Les spécialisations sont disponibles sous la forme d'espaces de travail : les Workshops, qui sont eux-mêmes subdivisés en diverses palettes d'outils : les Workbenchs. Le fait de double cliquer sur un élément de l'arbre peut initier un changement de contexte, pour mettre à disposition les outils lui correspondant.

Le service WorkbenchTransition se propose de gérer ces transitions et des les exporter *via* la plate-forme peer to peer, pour ainsi proposer à un utilisateur distant les bons outils au moment opportun.

IV.2 Les workshops et les workbench

CATIA se veut une application de modélisation 3D généraliste, des milliers d'utilisations peuvent en être faites. Mais comme il serait vite ingérable d'avoir à disposition de nombreuses commandes qui ne correspondraient pas à nos besoins, les fonctionnalités de CATIA ont été divisées en espaces de travail nommés Workshop. Chaque espace de travail correspond en fait à un métier donné, comme par exemple l'aéronautique et l'électronique. Ainsi chaque corps de métier peut acquérir les fonctionnalités permettant de répondre à ces besoins.

Mais ce découpage n'étant pas encore assez fin, les workshops sont eux-mêmes découpés en Workbenchs. Il s'agit de suite d'outils répartis en fonction de leur action.

IV.3. Protocole d'échange

L'initiation de la transition de contexte est induite par un changement de chemin actif dans l'arbre(c'est à dire l'objet qui va être édité). Pour effectuer la transition sur un peer distant il suffit d'identifier l'élément actif courant et de le renvoyer vers les autres peers sous forme de GUID.

Le protocole d'échange se voit ainsi très simplifié. En effet un seul message suffit pour effectuer le changement. Pour cela l'émission d'un événement vers un groupe va être utilisée.

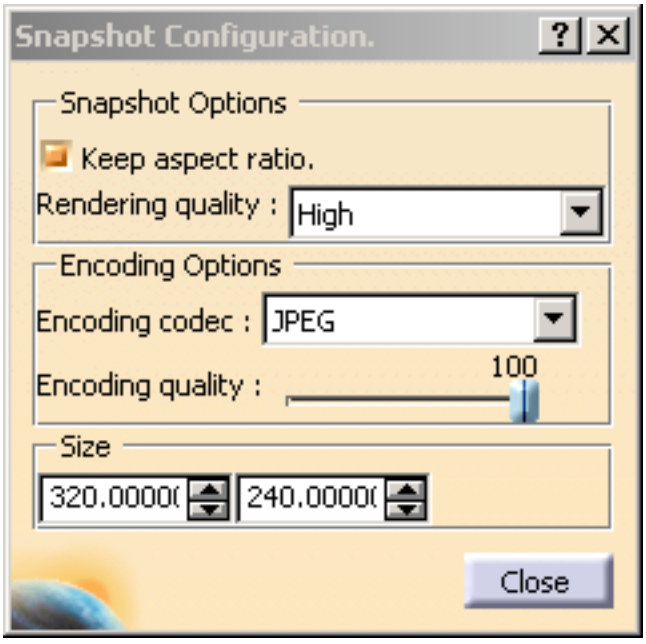
V. SERVICE DE SNAPSHOT

Ce service est dédié à réaliser des captures d'écran.

V.1. Interface graphique

Voici l'interface graphique liée à ce service.

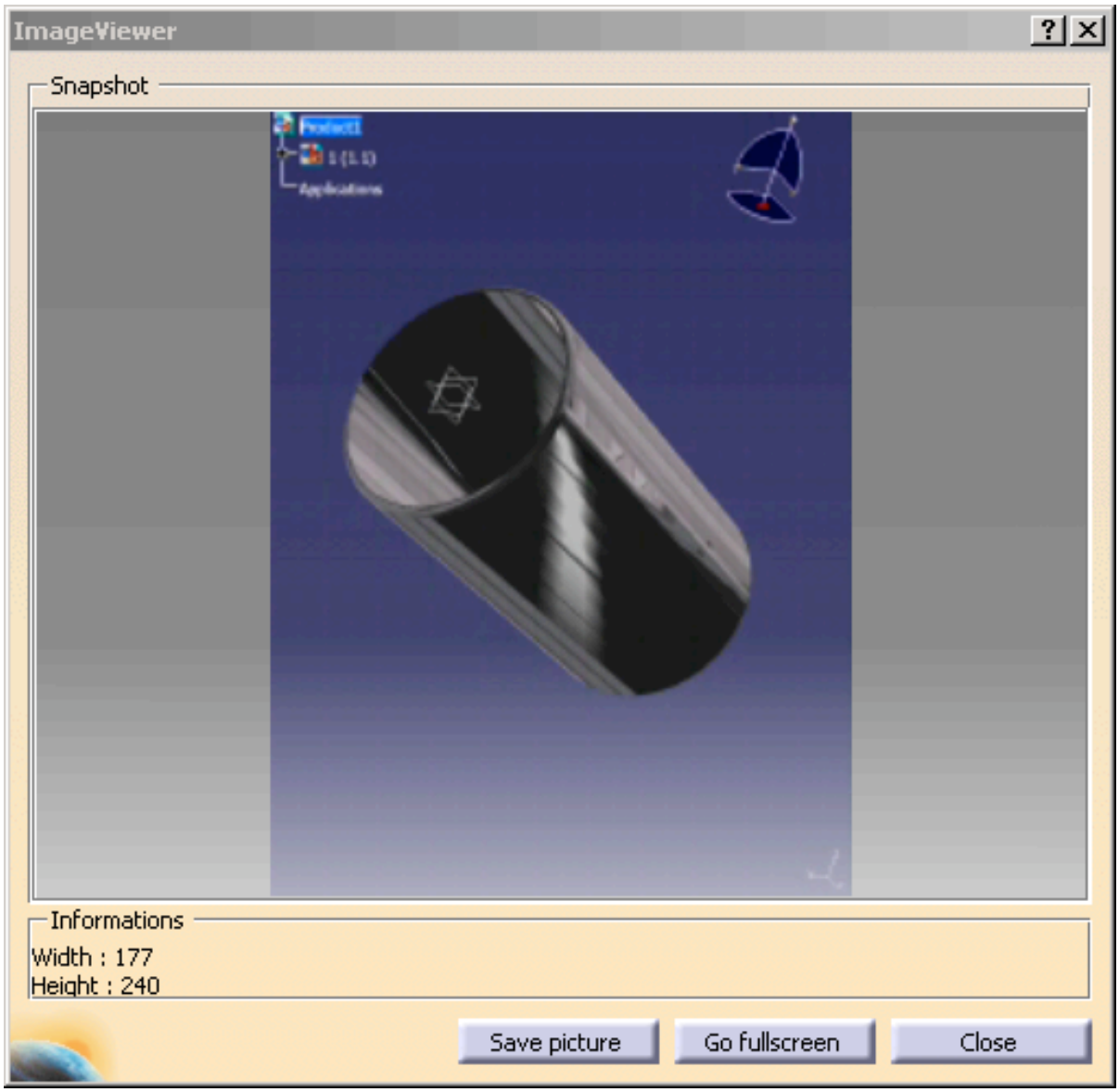
Le service est disponible sous la forme d'une barre d'outils. Deux actions sont alors proposées à l'utilisateur : soit de prendre une photo d'écran, soit configurer le service. Dans le deuxième cas la boite de dialogue ci-dessous apparaît.



Grâce à cette fenêtre il est possible d'agir sur la prise de vue. On peut spécifier la qualité du rendu 3D, son aspect (maintien du rapport hauteur/largeur). Il est aussi possible de choisir le type de compression des données renvoyées : RGB, Noir&Blanc, JPEG… Enfin on peut spécifier la taille de la photo.

Un fois le service configuré, nous sommes enfin prêts pour prendre des snapshots. Pour cela il suffit de cliqué sur le premier bouton de la barre d'outils.

Notre prise de vu apparaît alors à l'intérieur de l'ImageViewer (ci-après). Celui-ci nous permet de la sauvegarder sous différents formats (BMP, JPEG, PNG, RGB, TIFF) ou encore de l'afficher en plein écran.

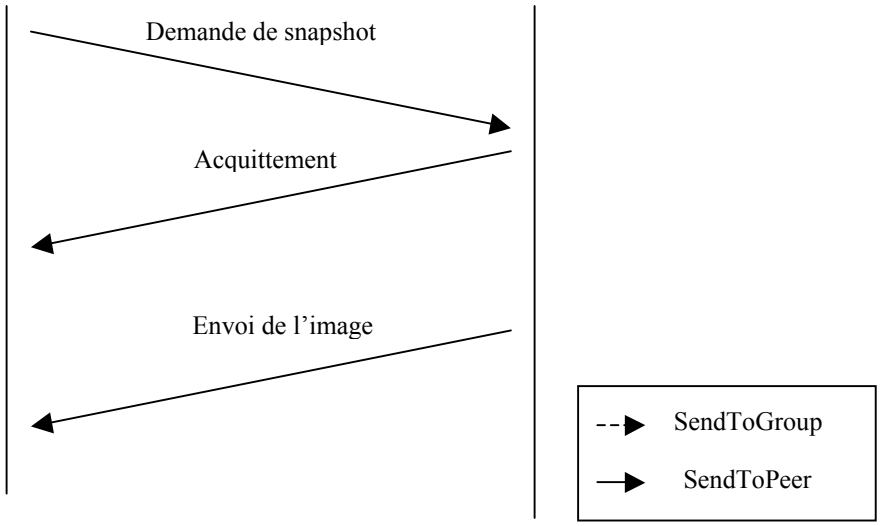


V.2. Le protocole de communication

Techniquement le processus de capture est très simple. Il se déroule en trois étapes :

1. Demande de snapshot (envoi de la configuration désirée)
2. Validation par le peer distant.
3. Si tout est OK, il renvoi l'image sous la forme demandée.

Ces étapes correspondent en fait à l'invocation d'une méthode distante en mode synchrone : sérialisation des paramètres, retour de la fonction, sérialisation des paramètres de retour.



Echanges lors d'une demande de snapshot

Ce service possède des intérêts multiples. Entre deux CATIA, il peut servir lors d'une revue collaborative distante ou encore de savoir si un de nos collaborateurs est en train d'effectuer des modifications sur une pièce donnée. Il peut être aussi utilisé par un Pocket PC, trop peut puissant pour effectuer le rendu de pièce CATIA. En effet ce service est actuellement utilisé par un autre stagiaire (Laurent MICHOT) développant des outils collaboratifs pour « ordinateurs de poche ».

Conclusion

De part la diversité des thèmes abordés, ce stage m'a permis d'acquérir de nombreuses connaissances techniques :

- Utilisation du langage C++, modélisation objet avancée.
- Utilisation de l'environnement de développement CAA v5.

Ce stage a été pour moi une occasion unique de découvrir l'organisation du travail, la manière dont sont gérés les projets au sein d'un groupe d'envergure mondiale. Le fort contenu technologique de ce stage m'a permis de mettre en pratique les connaissances enseignées au sein de la spécialisation Architecture et Réseaux de ma formation d'ingénieur en Réseaux Informatique et Communication Multimédia, mais aussi de me familiariser avec de nouveaux outils (développement sous Windows).

Même si le travail effectué lors de ce stage ne s'inscrit que dans le cadre de la réalisation de prototypes d'applications, et pourra donc être remanié lors de développements ultérieurs, cela restera pour moi une expérience enrichissante, et par bien des aspects, passionnante.

Bibliographie

I. DOCUMENTS

- Dejan S. Milojicic, Vana Kalogeraki, Rajan Lukose,Kiran Nagaraja1, Jim Pruyne, Bruno Richard, Sami Rollins ,Zhichen Xu
  Peer-to-Peer Computing, Rapport technique, HP Laboratories Palo Alto, Mars 2002

II. LIENS INTERNET

- The MVC design pattern : http://www.cs.indiana.edu/~cbaray/projects/mvc.html
- Le site du projet Freenet : http://freenet.sf.net
- Le site du projet JXTA : http://www.jxta.org
- Le site .NET : http://www.microsoft.com/net/

Conception et développement d'applications Peer to Peer

La technologie Peer to Peer, actuellement en pleine expansion, fait couler beaucoup d'encre. En effet elle doit sa renommé à des applications de partage de fichiers, telles que « Gnutella » ou « Napster », qui utilisent la cette technologie toujours dans le même but : le partage de fichiers.

Mais cette technologie disposant de nombreux atouts architecturaux promet beaucoup dans le domaine du travail collaboratif. Ce qui est plutôt intéressant pour une application telle que CATIA, où des dizaines de collaborateurs sont amenés à travailler ensemble.

BERNARDET William