



Rapport de TER
Master 1 Informatique Logiciel

Prototypage Virtuel et Modélisation

TAHAN Mehdi

**Laboratoire d'Informatique
et des Systèmes Complexe**

Département d'informatique
U.F.R. Sciences et Techniques
Université de Bretagne Occidentale
20, avenue Le Gorgeu
BP 809
29285 BREST cedex
FRANCE

8 Juin 2009

Résumé : La complexité croissante des systèmes à concevoir oblige à créer de nouveaux outils et à adopter de nouvelles méthodes de conception. Aujourd'hui, on dispose, par exemple, de l'ingénierie dirigée par les modèles pour la conception ainsi que de la réalité virtuelle pour l'exploitation numérique. Mais il existe des manques en la matière comme l'exploitation en synergie des différents outils. Une nouvelle approche a donc été proposée : prise en compte de l'environnement dès le début de la conception et comportement induit par la définition.

Mot-clés : Réalité virtuelle, Prototypage virtuel, Modélisation, Serious gaming

Remerciements

Amara Touil pour sa patience et son aide en SysML

Jean Vareille pour ses conseils et le partage de sa passion

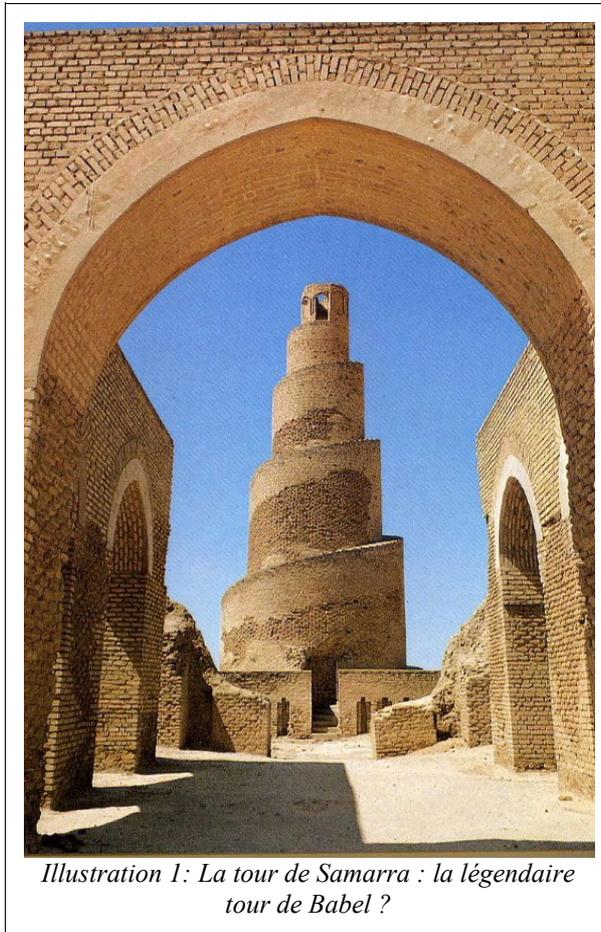
Dominique Morvan pour son aide à la rédaction de ce document

« Un bon croquis vaut mieux qu'un long discours », Napoléon Bonaparte

Table des matières

Introduction.....	5
.....	6
État de l'art.....	6
Point de vue ingénierie.....	6
Point de vue simulation & réalité virtuelle.....	9
Les langages & modèles.....	11
Point de vue ingénierie.....	11
Point de vue simulation & réalité virtuelle.....	14
Faisabilité – réalisation – étude de faisabilité.....	15
Point de vue ingénierie.....	15
Point de vue simulation & réalité virtuelle.....	17
Conclusion.....	21
Bibliographie.....	22

Introduction



La tour de Samarra se trouve en Irak, à 130km de Bagdad, et est le minaret d'une mosquée. Elle a inspiré la vision européenne de la tour de Babel.

La tour de Babel est l'exemple même de la gestion de la complexité. Un grand nombre de participants à une entreprise ne pouvant se comprendre, ne pourront jamais venir à bout d'un système trop complexe. L'interopérabilité des différents acteurs est primordiale.

Il est important de disposer d'un outil permettant le dialogue de manière unifiée et univoque. L'ingénierie actuelle est encore à la recherche de cette méthode « miracle ».

La plupart des systèmes techniques actuels est composée d'une partie matérielle, d'une partie électronique et d'une partie immatérielle (du code source). Cette imbrication de domaines augmente considérablement la complexité de leur conception.

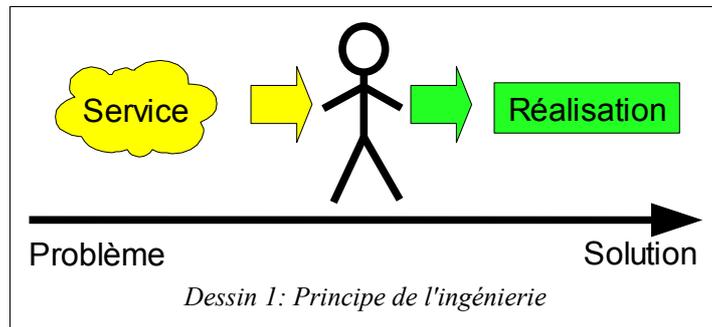
Pour répondre à cette problématique, nous verrons l'état de l'art en ingénierie ainsi qu'en réalité virtuelle, puis les outils d'exploitation associés. Enfin une nouvelle approche de la conception sera testée.

1.État de l'art

1.Point de vue ingénierie

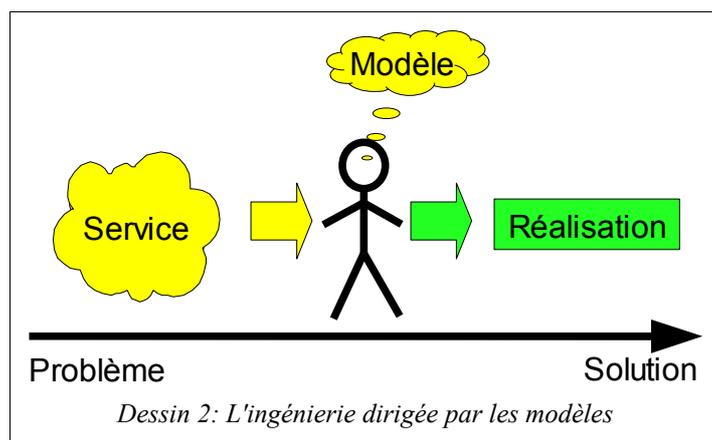
« Ensemble des aspects technologiques, économiques, financiers et humains relatifs à l'étude et à la réalisation d'un projet, qu'il soit industriel, scientifique ou de société », fr.wiktionnaire.org

L'ingénierie, aussi appelé génie, permet de conceptualiser de manière précise et rigoureuse des systèmes alliant la mécanique, l'électronique ou d'autres domaines. L'ingénieur conçoit des systèmes déterministes répondant au plus près à la demande.

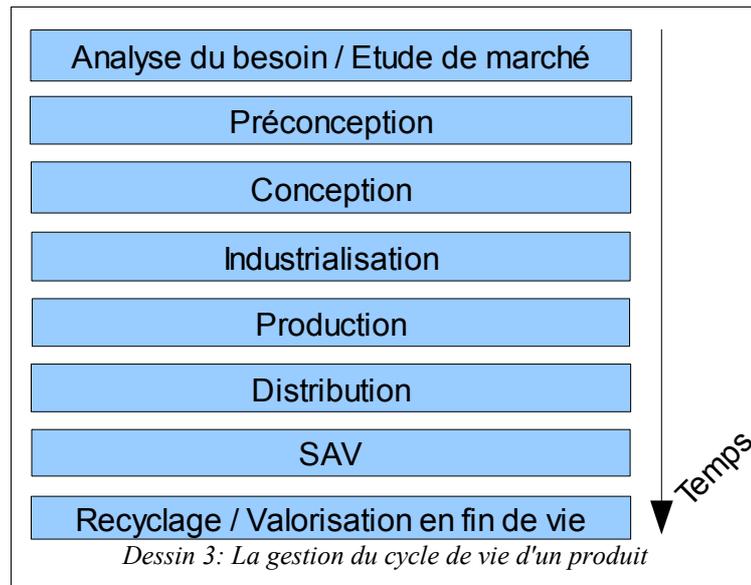


Avec l'augmentation de la complexité des systèmes, un nouveau type d'ingénierie a vu le jour : l'ingénierie dirigée par les modèles (ou IDM, MDE en anglais). L'utilisation de modèles a permis de simplifier la vision de l'ingénieur et ainsi d'augmenter la complexité des systèmes.

« Un « modèle » est la représentation mentale d'un être du monde réel et de son fonctionnement : quand on dispose d'un modèle, on peut simuler mentalement le comportement de cet être. », www.volle.com



La notion de PLM (Product Lifecycle Management ou gestion du cycle de vie du produit en français. Cf Dessin 3) est à rapprocher de l'ingénierie. En effet l'ingénierie intervenant dès les premiers instants du cycle de vie du produit, il est déterminant pour le futur du produit.



Ainsi le schéma classique de conception est :

1. Rédaction du cahier des charges en langage naturel
2. Modélisation
3. Modélisation 3D

Aujourd'hui plusieurs problèmes sont rencontrés lors de l'utilisation de ce schéma :

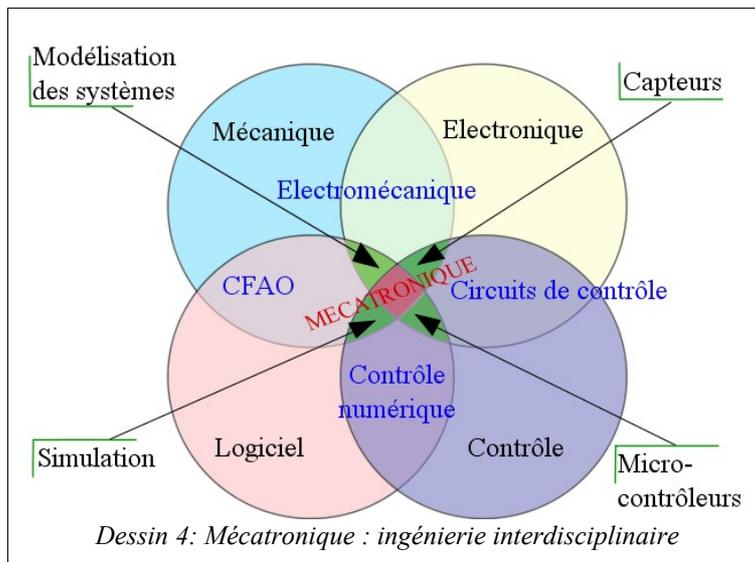
- Le cahier des charges relève de l'art littéraire et inclut toutes ces ambiguïtés. Adil Majjad travaille actuellement sur ce problème (cf. *Modélisation semi-formelle et cahier des charges* dans la section bibliographique)
- la représentation géométrique n'est prise en compte que tardivement dans le cycle de conception (cf. *Important Aspect of Early Design in Mechatronic*)

Différentes pistes d'exploration ont été imaginées comme l'utilisation d'un langage univoque pour l'écriture du cahier des charges ou de l'utilisation d'un modèleur 3D associé à une description objet dès les premières phases de conception.

Enfin, une définition de la mécatronique :

« La mécatronique est la combinaison synergique et systémique de la mécanique, de l'électronique et de l'informatique temps réel. L'intérêt de ce domaine d'ingénierie interdisciplinaire est de concevoir des systèmes automatiques puissants et de permettre le contrôle de systèmes complexes. », fr.wikipedia.org

Cette définition ainsi que le schéma (Dessin 4) démontrent la complexité de la discipline et de l'importance de l'utilisation de modèle.



1. Point de vue simulation & réalité virtuelle

La réalité virtuelle (ou RV) est une notion que l'on retrouvait déjà il y a plus de 2000 ans dans les écrits de philosophes, tel Platon dans son « *allégorie de la caverne* ». Mais c'est dans la seconde moitié du XX^{ème} siècle qu'elle prend réellement la signification qu'on lui connaît aujourd'hui grâce à des auteurs de science-fiction, comme Phillip. K. Dick, qui l'utilise comme trame de leur récit.

Ainsi le premier casque de réalité virtuelle a été créé en 1968-70 par Ivan Sutherland du MIT. Un peu auparavant (1967), Fred Brooks de l'Université de Caroline du Nord avait développé un bras à retour d'effort pour environnements virtuels. Des travaux sur des cockpits virtuels (le programme Visually-Coupled Airborne Systems Simulator ou VCASS dirigé par Thomas Furness de l'US Air Force) virent le jour en 1977. De nombreux travaux concernant cette nouvelle discipline furent réalisés durant cette période.

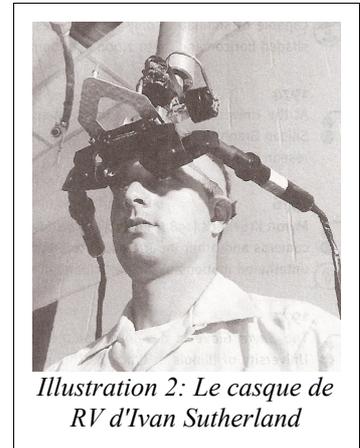


Illustration 2: Le casque de RV d'Ivan Sutherland

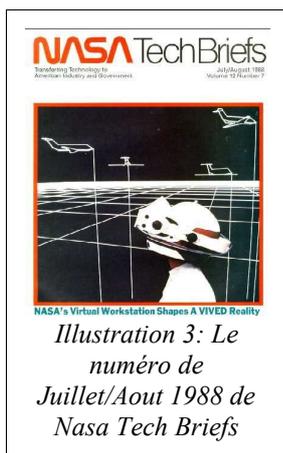


Illustration 3: Le numéro de Juillet/Aout 1988 de Nasa Tech Briefs

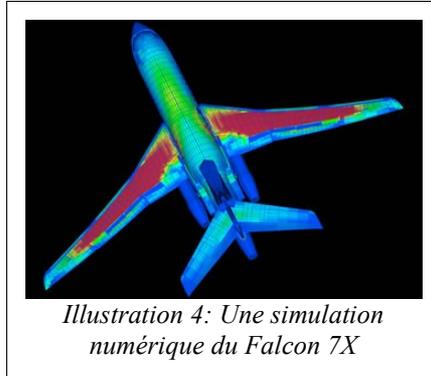
Mais la RV ne prit son envol qu'en 1984, lorsque la Nasa lança le programme VIVED ou Virtual Visual Environment Display. Ce programme avait pour but de développer un nouvel outil d'affichage d'informations et de données pour les futures missions spatiales des astronautes. Ainsi fut développé une version « low-cost » d'un casque de réalité virtuelle employé par l'US Air Force utilisant des composants du commerce. Jusque dans les années 90, le programme va s'enrichir : amélioration du casque, ajout d'un environnement acoustique tridimensionnel, élargissement du domaine d'application et interaction via la main. Le programme VIVED mua en VIEW (Virtual Environment Workstation) en 1988, l'objectif initial ayant été actualisé et offrant une station complète de travail pour environnements virtuels.

Aujourd'hui la RV est on ne peut plus présente, même pour le grand public : en démontre l'augmentation de l'importance du monde des jeux vidéos et l'apparition de nouveaux moyens d'interaction avec les univers virtuels comme la wiimote de Nintendo, pour une meilleure immersion. De plus, plusieurs branches se sont développées au cours du temps permettant à la RV un plus grand champ d'application (réalité virtuelle immersive, réalité augmentée ...).

Afin d'unifier les différentes définitions de la RV, un document a été rédigé : le « Traité de la réalité virtuelle ». Il est communément admis que :

« La finalité de la réalité virtuelle est de permettre à une personne (ou à plusieurs) une activité sensori-motrice et cognitive dans un monde artificiel, créé numériquement, qui peut être *imaginaire, symbolique ou une simulation de certains aspects du monde réel*, » [Fuchs, 1996], [CRTRV, 2004].

Dans le monde de la conception de systèmes, l'utilisation de la RV est un gain important permettant la formation, dans des simulateurs, des utilisateurs ou des techniciens en charge de la maintenance, la manipulation en situation réelle du système ou même la téléopération. La création de prototype virtuels ou DMU (Digital MockUp) permet ainsi de développer des systèmes entiers sans créer un seul prototype. Dassault et son Falcon 7X, entièrement conçu de manière numérique, en est le parfait exemple. Cet « exploit », qui tend à devenir la norme, a été rendu possible par l'utilisation du logiciel Catia permettant d'exploiter une chaîne de conception numérique.



L'utilisation de prototypes numériques dans des environnements de réalité virtuelle apporterait un plus non négligeable, aussi bien en phase de conception qu'en phase d'exploitation.

Conclusion :

Aujourd'hui, nous disposons d'une grande maîtrise dans les domaines de l'ingénierie et de la réalité virtuelle. Mais ces deux mondes peinent à se rencontrer.

2. Les langages & modèles

1. Point de vue ingénierie

Des outils informatiques ont été développés afin d'augmenter le potentiel de l'ingénierie.

Ainsi des langages spécifiques à la description des systèmes ont été inventés : UML et SysML entre autres. L'importance de langage de description commun est capitale pour mener à bien tous projets. L'échec légendaire de la tour de Babel en est le parfait exemple : sans moyen de communiquer, les hommes n'ont jamais réussi à terminer leur entreprise. De même, le caractère univoque d'un tel langage est primordial, car, même si les hommes avaient pu parler la même langue mais n'avaient pu partager les mêmes notions, il est certain que l'œuvre aurait vacillé avant de s'effondrer.

Ainsi l'UML (Unified Modeling Language) a été créé afin de faciliter le développement de programme informatique par l'ingénierie dirigée par les modèles. C'est un langage graphique de modélisation des données et des traitements qui s'appuie sur des langages plus anciens comme Booch, OMT ou OOSE. L'originalité d'UML réside dans l'utilisation de diagrammes (13 au total). Ces diagrammes permettent la description du programme d'un point de vue objet et sont dépendants hiérarchiquement.

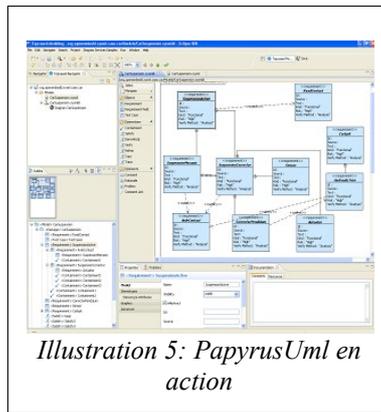
D'UML est né SysML même si cela reste flou : certains pensent que SysML est un simple profil d'UML tandis que d'autre pensent qu'il peut être considéré comme étant un langage propre. UML étant destiné à la description de programme informatique, SysML est quant à lui destiné au système. Il perd certains diagrammes d'UML et en gagne de nouveaux portant à 9 le total.

SysML (Systems Modeling Language) est donc orienté objet et reprend les avantages d'UML. Mais le problème de SysML est qu'il ne doit pas être pris pour autre chose que ce qu'il est : un langage de haut niveau. Ainsi, comme l'ont rapporté Belloir N. , Bruel J.M. , Hoang N. et Congduc P dans leur retour d'expérience concernant la modélisation d'un réseaux de capteurs, SysML montre ses limites jusqu'à une certaine granularité. Il ne permet pas de développer complètement un système, dès que l'on rentre dans un domaine spécifique (électronique, informatique, ...) ; il est impératif de revenir à un langage adapté. L'étude montre toutefois l'intérêt indéniable de SysML et les progrès apportés au domaine de la modélisation système, mais souligne les lacunes du langage.

Modélica est un langage orienté objet mais se différencie de ses concurrents par la définition sous forme d'équation des objets présents dans le système.

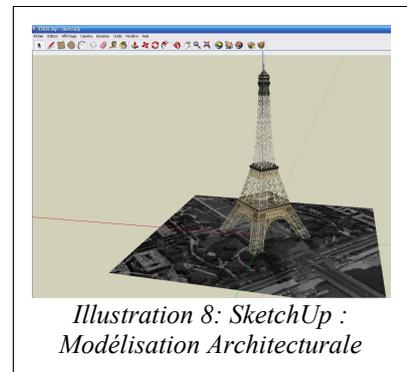
Il existe aussi des langages plus spécifiques comme VHDL (ou VHSIC) dédiés à la description matérielle des systèmes électroniques numériques ou AADL (Architecture Analysis and Design Language, anciennement Avionics Architecture Description Language) dédié à l'embarqué dans le domaine de l'avionique ou de l'automobile (mais pas exclusivement).

Des logiciels permettant une mise en œuvre simplifiée de ces langages existent : Topcased et PayrusUml tous deux basés sur l'environnement Eclipse. Ce sont des éditeurs gérant SysML et UML sous licence open source.



En parallèle, on peut trouver des logiciels de CAO 3D :

- Catia (Conception Assistée Tridimensionnelle Interactive Appliquée) de Dassault Systèmes : Le leader dans le domaine. Il s'organise autour d'ateliers spécifiques à chaque domaine traité (modélisation 3D, schéma de câblage, cinématique ...). Il offre la possibilité a plusieurs équipes réparties autour du monde de collaborer à un même projet grâce à la base de donnée Enovia. Il est utilisé dans le domaine de l'aéronautique mais aussi de la construction navale, de l'automobile ...
- SolidWorks de Dassault Systèmes : SolidWorks a été racheté par Dassault Systèmes en 1997. L'entreprise a d'ailleurs décidé de reprendre la philosophie du logiciel pour son logiciel de CAO phare : Catia. Les 2 logiciels sont donc très proches. Malgré tout, SolidWorks semble plus être destiné au domaine scolaire par son accessibilité et est plus permissif que son grand frère beaucoup plus rigoureux.
- Topsolid de Missler Software : il est le concurrent direct de Catia et Solidworks et fonctionne sur les même bases.
- Blender : il n'est pas à proprement parler un logiciel de CAO. C'est une suite libre et gratuite d'animation, de modélisation et de rendu 3D. Il est a rapprocher de la création graphique. Il est capable de traiter des modèles VRML et intègre un moteur physique (Bullet).
- Google SketchUp : ce logiciel est orienté architecture et se distingue de ses concurrents par son principe d'inférence. On pourra lui reprocher sa relative jeunesse mais Google a su imposer ses outils auprès d'une communauté toujours plus grande. SketchUp ne doit pas être vu comme un outil permettant la modélisation d'un système, mais son orientation le destine à la modélisation 3D du monde et l'exploitation sous Google Earth.
- Open Cascade de EuriWare : Anciennement développé par Matra Datavision puis par Dassault Systèmes, Open Cascade appartient au monde de l'open source depuis 1999.



A noter qu'aucun des logiciels de CAO classique ne propose une représentation de l'environnement.

Au mieux, Catia tente de symboliser l'horizon grâce à un dégradé de couleur en fond. Google SketchUp n'intègre pas non plus la notion d'environnement bien qu'il soit clairement orienté monde réel.

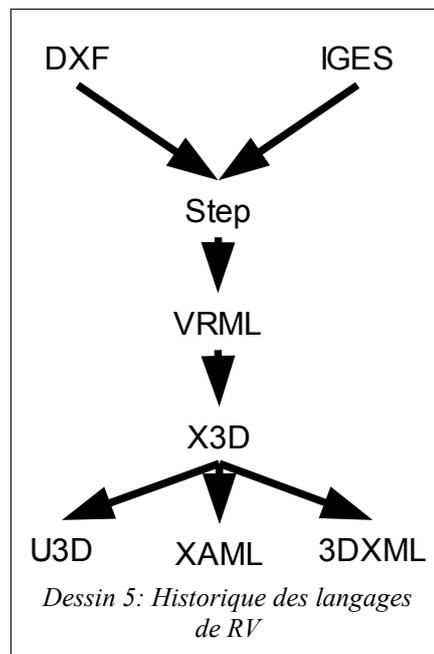
Enfin la déclaration de l'immatériel (code source par exemple) n'est pris en compte que de manière relativement succincte, via des langages de script en général (CATScript, VB, VBA, VBScript ... pour Catia), des paramètres ou des formules (VHDL entre autres).

1. Point de vue simulation & réalité virtuelle

Dans le domaine de la simulation et de la RV, on trouve de nombreux langages et outils.

Catia propose Simulia permettant la simulation du système conceptualisé. Il apporte une solution unifiée d'analyse par éléments finis.

Pour la représentation d'éléments géométriques, il existe un certain nombre de langage et formats.



Le DXF et l'IGES servent à représenter des modèles 3D, principalement dans le domaine de la CAO. Le VRML (Virtual Reality Modeling Language) date de 1994. Le but premier de ce langage est de permettre la représentation d'univers interactifs 3D virtuels. Sa spécification est à la charge du consortium Web3D. Aujourd'hui le X3D (Extensible 3D) lui succède depuis 2005. A l'origine, plusieurs entreprises avaient rallié le consortium afin de définir ce nouveau langage ; mais aujourd'hui, les plus grandes s'en sont désolidarisé : Dassault, Intel et Microsoft. Depuis, ils ont chacun créé leur propre langage 3D basé sur XML, respectivement 3D XML, U3D et XAML. Cependant, ces langages ne sont pas aussi ouverts et documentés que le X3D.

Il existe de nombreux univers virtuel, 3D aussi appelés métavers (terme inventé par Neal Stephenson, auteur de science-fiction, en 1992) . Les MMORPG en font partie ainsi que des applications moins ludiques comme le célèbre 2nd life.

Dans le domaine industriel, Simulia de l'entreprise Dassault est axé sur la simulation. Dassault propose aussi Virtools, permettant d'exporter des modèles vers des mondes virtuels.

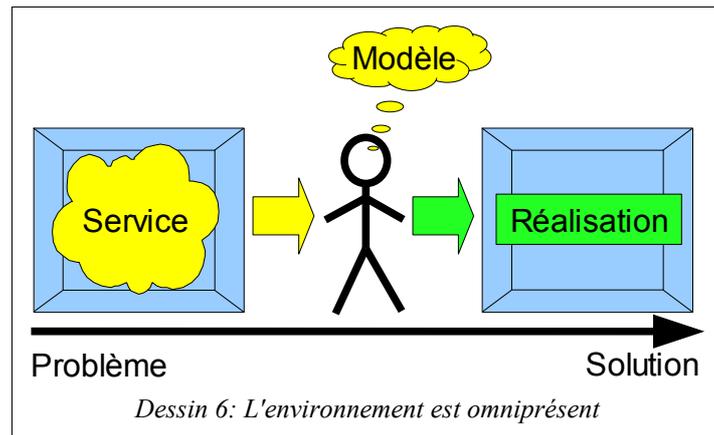
Enfin le CERV (Centre Européen de Réalité Virtuelle) propose des outils comme Mascaret ou Arévi (Atelier de REalité Virtuelle) permettant l'adjonction de comportement autonome aux agents présent dans les métavers.

2.Faisabilité – réalisation – étude de faisabilité

1.Point de vue ingénierie

Il est clair que l'environnement n'est pas une donnée prise en compte dès le début de la conception dans les différents outils présents sur le marché actuellement. Pourtant l'environnement est déjà présent avant même le besoin. Et il sera présent bien après la fin du cycle de vie du système.

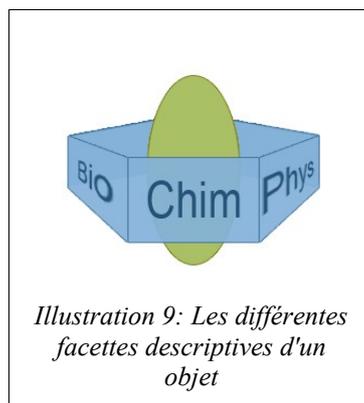
Il est donc primordial que l'environnement soit pris en compte dès l'origine du projet.



Un environnement peut se décrire en plusieurs points :

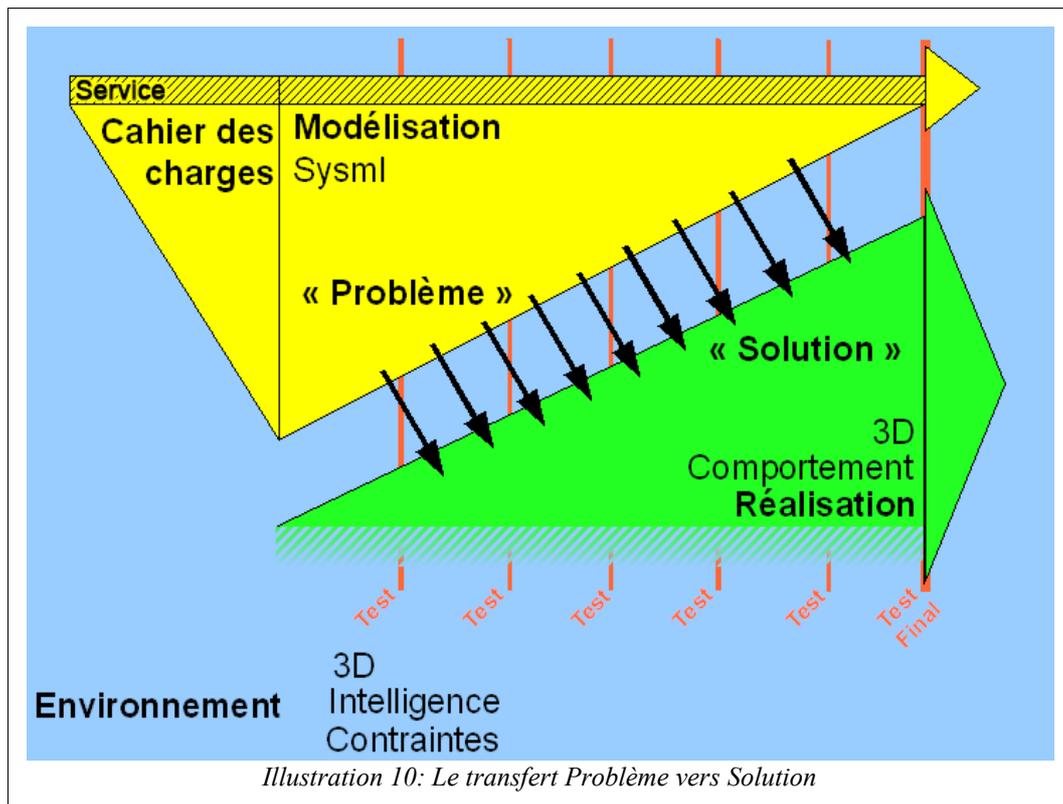
- une représentation 3D
- des contraintes dans différents domaines (physique : loi de la gravité par ex, chimique : l'oxydation des métaux dans l'eau ...)
- une intelligence : des acteurs comme l'être humain, les animaux ...

De même, un système est composé d'objets qui peuvent eux même être définis suivant différents domaines (voir illustration 9).



Les décrire tous de manière exhaustive semble contre-productif. Une autre voie devrait donc être explorée.

Après une longue réflexion, un schéma de conception a été produit :



L'illustration 10 montre une nouvelle vision de la conception d'un système :

- L'environnement est omniprésent, existe avant et existera après la conception
- Une demande de service est créée : ce service sera le fil conducteur du problème
- A partir de ce service, un cahier des charges est rédigé
- La modélisation s'effectue à partir du cahier des charges et inclut une description précise des contraintes de domaines
- A partir du modèle, un transfert est effectué du problème vers la solution. La solution « absorbe » le problème
- A intervalle régulier, des tests sont effectués : il est important de pouvoir utiliser sur un même plan aussi bien le modèle que la réalisation.
- La conception s'achève lorsque la solution a absorbé la totalité du problème (à l'exception du service, qui reste l'élément de validation le plus haut).

À ce schéma, une précision doit être apportée : il existe 3 notions d'environnement dans ce schéma.

- l'environnement : il est global et peut être associé à l'instance la plus grande de l'environnement réel (par exemple, l'univers)
- l'environnement utile : c'est l'environnement pour lequel le système est conçu (une personne possédant des compétences en conduite est une partie de l'environnement utile d'une voiture, une route l'est aussi)
- l'environnement pertinent : il correspond à une notion assez floue de l'environnement utile (qu'il englobe). En théorie, tout système est prévu pour fonctionner en environnement utile mais se retrouve plongé dans un environnement pertinent dans la réalité (le chat qui griffe la voiture ...). Sa limite est donc très floue et reste à l'appréciation du concepteur. Toutefois, il serait intéressant de laisser à l'ordinateur le choix de cette limite, l'humain ayant souvent une vision réductrice du problème.

Enfin la possibilité de tester des modèles et des réalisations dans un même univers reste complexe. L'utilisation de contrainte de domaine, dès la modélisation du système, permettrait une certaine concordance.

1.Point de vue simulation & réalité virtuelle

Actuellement la modélisation 3D se base sur une représentation géométrique associée à une multitude de méta-informations représentant par exemple des propriétés physiques. Ainsi, dans Catia, la modélisation 3D d'un cardan se fait via une représentation géométrique associée à la méta-information « cardan ». Il avait été envisagé d'ajouter à SysML voire éventuellement dans SysML, ce type de modélisation (représentation 3D et méta-information). Durant le développement, la question de la nécessité de la méta-information a été soulevée.

La méta-information est une donnée qui contraint la vision du concepteur : « puisque cette pièce possède la méta-information « cardan », alors, elle aura le comportement d'un cardan ». Cette vision correspond totalement à la philosophie de l'ingénieur qui se base sur des comportements déterministes.

Malgré tout, cette contrainte peut être source de problème. Ainsi, dans l'histoire de l'ingénierie, on retrouve en grand nombre de problèmes dûs à des comportements n'ayant pas été déterminés à la conception car invisibles pour l'ingénieur.

Ce qui suit en est le parfait exemple : une nouvelle génération de TER utilisant des freins à disque avait été conçue. Les freins, une fois en action et sous l'effet des forces de frottements, expulsaient de la matière sur les voies, de la limaille d'acier conductrice. Des commutateurs situés au niveau des passages à niveaux détectant l'arrivée des TERs se retrouvèrent en dysfonctionnement ce qui entraîna l'inactivité des barrières alors qu'un train passait. La limaille de fer se comportait comme un shunt qui empêchait la coupure de contact nécessaire à la détection de l'arrivée d'un train par le système de commande de la barrière. Ce dysfonctionnement causa la mort d'au moins un conducteur qui traversait les voies car la barrière ne s'était pas levée (accident de 2004 à la Roche Maurice, Finistère)

Il est clair que ce comportement n'avait pas été imaginé par les concepteurs. De plus, ce cas étant très rarement présent du fait de l'improbabilité de la réunion des conditions nécessaires (la succession d'évènements climatiques très précis étant un facteur majeur du déclenchement du comportement), sa visibilité en était encore amoindrie. Mais cette vision déterministe utilisée dans l'ingénierie montre ici ses limites.

Il a donc été imaginé que les méta-informations devaient être bannies au maximum. Dans le cas où la représentation géométrique du cardan est cohérente avec la réalité physique, alors sa seule représentation doit suffire à déterminer son comportement. Un second paramètre, la matière constituant le cardan, doit tout de même être présent.

L'intérêt de cette façon de procéder est double :

- L'ingénieur peut se passer de la saisie d'une partie des méta-informations, ce qui représente un gain de temps.
- Le comportement du système n'est plus contraint par la vision de l'ingénieur. C'est le logiciel

exploitant et validant les données de représentation qui sera ici la contrainte (le moteur physique par exemple). Ceci permet de garantir une évolution plus facile en fonction de l'évolution des modèles des domaines entrant en jeu (physique, chimique, biologique ...), seul le moteur logiciel du domaine étant à modifier. De plus, la possibilité de tester le système dans un domaine pour lequel il n'a pas été prévu devient possible (par exemple optique). Enfin la « vision contrainte » du système ne provient plus du concepteur mais du logiciel. On peut imaginer que le peu de contraintes encore présentes disparaissent peu à peu avec l'évolution (et pourquoi pas le partage via l'open-source) des moteurs de domaine.

Il a donc été décidé de retirer au maximum du modèle 3D final les méta-informations. Pour démontrer la pertinence de l'approche, des tests ont été réalisés dans deux mondes virtuels : 2nd Life et Half Life 2.

Si Half Life 2 a été choisi, c'est pour 3 raisons:

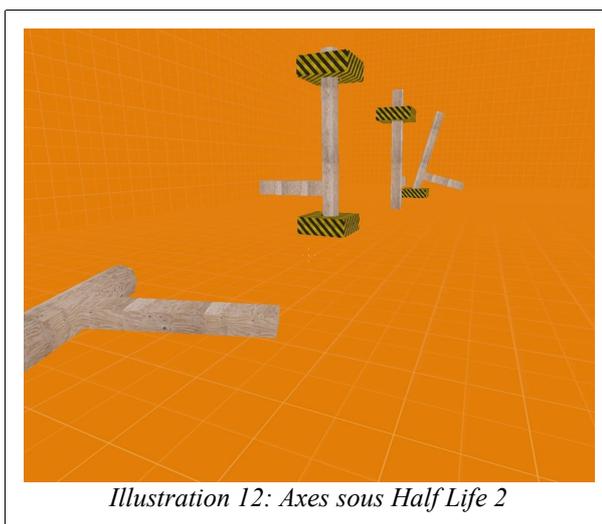
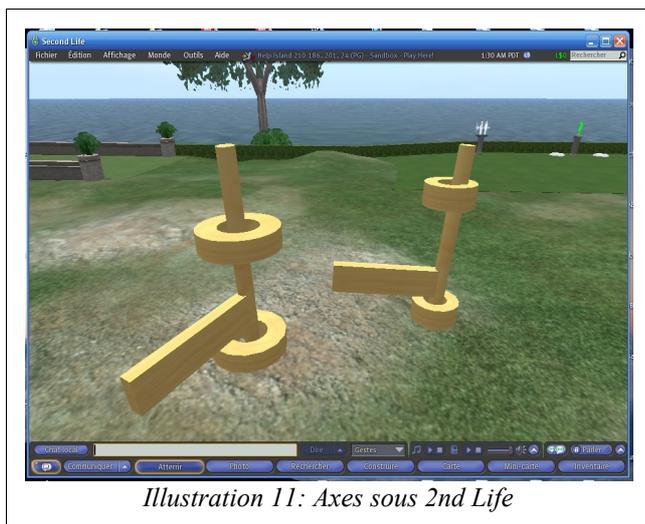
–Il intègre l'un des moteurs physiques les plus performants dans le monde vidéoludique : le moteur Havok

–Il offre un éditeur simple d'accès et gratuit: le Hammer. De plus, le moteur de Half life 2 étant connu pour sa capacité à être modifié, une grande communauté existe depuis plus de 10 ans (date de sortie du premier épisode)

–Ne gérant que la physique, il est possible, par l'adjonction de script au sein des objets, de simuler des comportements biologiques, chimiques, optiques ... De plus, dans le cas d'une étude plus poussée, la possibilité de modifier le Sdk est intéressante afin d'intégrer à tout objet différentes propriétés sans avoir à les « scripter ». Cette avantage est propre au monde du jeu vidéo, les modeleurs 3D classiques ne proposant pas ce genre de possibilité.

De même le choix de l'utilisation de 2nd Life avait pour justification la validation sur différente plates-formes bien que le moteur employé reste le moteur Havok (Version 4).

Pour information, il existe des moteurs physiques open-source comme Ode ou Bullet (utilisé dans Blender).



Sous Half Life 2, les résultats ont démontré la pertinence de l'idée. Les « axes » avaient pour seule méta-information leur matière (plus quelques paramètres propres au moteur Havok), qui d'ailleurs n'est jamais entrée en ligne de compte durant les tests. Les « axes » ont eu un comportement d'axes,

bien qu'ils ne soient pas déclarés en tant que tels. Leur comportement étant de plus cohérent avec leur environnement (maintient via 0, 1 ou 2 attaches), comme dans la réalité.

Avec 2nd Life, les résultats ont été identiques mais il est à noter que la précision du moteur était plus faible et que, sur l'illustration 10, l'axe de droite n'avait pas le comportement attendu (il présentait de nombreux bugs) bien qu'il soit modélisé de manière convenable et identique à celle effectuée sous Half Life 2.

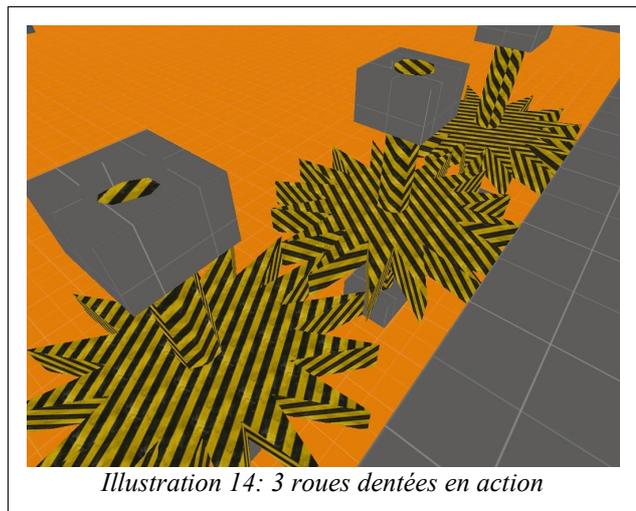
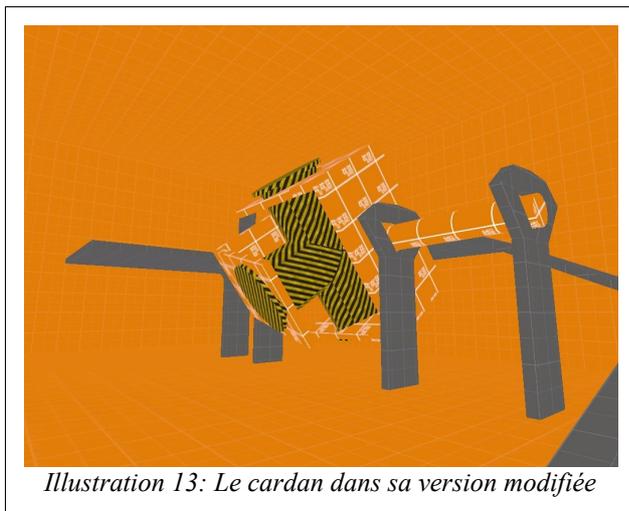
Ce test démontre donc la viabilité de l'idée mais aussi l'importance de la qualité du moteur physique.

Un second test fut lancé afin de pousser plus loin les possibilités du moteur physique et d'en trouver les limites. A partir de ce test, seul Half Life 2 fut utilisé comme plate-forme de développement.

La modélisation d'un cardan ainsi que quelques rouages fut testé.

Durant la phase de test du cardan, l'une des spécificité recherchées (le comportement est induit par la modélisation et non par le concepteur) fut mise en avant. Une première version du cardan, semblant cohérente, ne fonctionna pas. La pesanteur fit tomber les pièces du cardan au sol, comme ce qui ce serait produit dans la réalité, le rendant in-opérationnel. Quelques modifications (ajout de tête de maintien au croisillon central) permirent de rendre le cardan fonctionnel.

Les roues dentées ne posèrent pas de difficultés particulières et la transmission de la rotation des pièces se faisait parfaitement.



Il est à noter que seule l'utilisation d'un moteur (sous la forme d'une méta-information) fut utilisée.

Ce test a permis de montrer qu'il est possible de réaliser des systèmes plus complexes qu'un simple axe. Toutefois le moteur Havok semble montrer quelques limites. Comme il avait été montré dans 2nd Life, l'importance de la qualité du moteur est primordiale. Le cardan a du être modélisé avec une taille gigantesque afin de permettre au moteur physique de calculer les interactions. Le moteur Havok est 32 fois moins précis que le moteur graphique d'Half Life 2.

Une hypothèse peut être émise sur l'importance du format de la représentation 3D. Dans Half Life 2, la représentation 3D est entièrement polygonale. Pour une utilisation graphique, cela convient tout particulièrement, mais pour des calculs précis d'interaction physique, la limite est vite atteinte.

En effet, un disque (base d'un axe) est vu comme une succession de facettes et non pas comme une surface. Les mouvements induits par la rotation de l'axe sont donc saccadés. Le choix d'un langage de type VRML (permettant la déclaration d'un cylindre en tant que cylindre et non en tant que succession de polygone) pour ce type d'exploitation, semble donc mieux adapté.

Une dernière série de test eut pour objectif de modéliser au complet le système Saturne/MiaBot, mais, par manque de temps, la tâche n'a pu être finalisée. Différentes représentations (Illustration 15 & 16) ont été effectuées mais de nombreux bugs (dûs à la qualité du moteur Havok) n'ont pas permis d'obtenir le résultat souhaité.

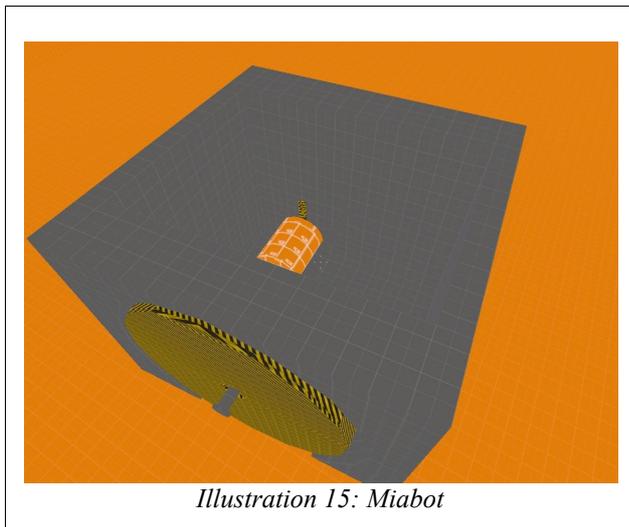


Illustration 15: Miabot

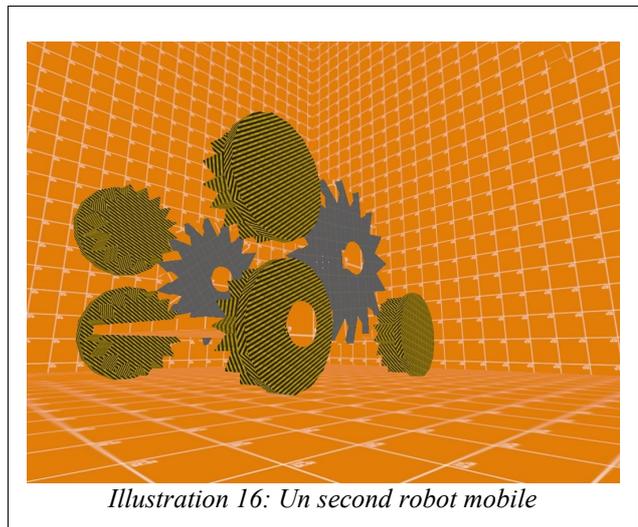


Illustration 16: Un second robot mobile

A travers ces tests, il a été démontré que la seule représentation géométrique associée au matériau utilisé, permettait de simuler le comportement du système.

Toutefois, il est important d'apporter quelques précisions:

- il est clair que Hammer est inadapté à la conception de systèmes complexes.
- l'utilisation de modèles (les objets 3D du jeu comme une arme, un véhicule ou un lampadaire) permettrait peut être d'avoir des interactions physiques plus précises.
- Half Life permettant l'import de modèle blender (via un système de conversion fait par un logiciel tiers), il serait envisageable d'offrir au jeu des bases de modèles très complètes accessibles sur Internet.

En conclusion, l'utilisation conjointe de moteur physique, chimique, biologique, etc, permettrait la validation dans un environnement du système conçu. D'un point de vue strictement matériel, les technologies actuelles sont prêtes : les moteurs physiques sont fortement parallélisables et exploitables via des cartes graphiques Nvidia ou Ati (voir l'utilisation de carte Tesla, entièrement dédiée au calcul parallèle) permettant des traitements bien plus lourds que celui étudié ici.

Conclusion

Plusieurs points sont à soulever :

- Le cahier des charges : doit il rester un art littéraire ou devenir un langage « cahier de charge » ? L'utilisation d'un langage semi-formel est une voie d'exploration.
- La nécessité de la prise en compte de l'environnement est un point capital pour le bon déroulement d'une conception de système.
- Le développement de nouveaux moteurs de domaine (à l'instar des moteurs physiques) doit être effectué.
- La prise en compte de contrainte de domaine durant la phase de modélisation est nécessaire.

Il reste donc beaucoup de travail dans le domaine de la recherche informatique concernant l'ingénierie. Mais l'utilisation de moteurs physiques dans le monde du jeu vidéo peut laisser supposer que la création de différents moteurs de domaines se multipliera dans le futur.

Bibliographie

- Belloir N. , Bruel J.M. , Hoang N. , Congduc P. . *Utilisation de SysML pour la modélisation des réseaux de capteurs*, Université de Pau et des pays de l'Adour.
- Turki S. (2008). *Ingénierie système guidée par les modèles: Application du standard IEEE 15288, de l'architecture MDA et du langage SysML à la conception des systèmes mécatroniques*, Thèse, Laboratoire LISSMMA Supmeca Toulon.
- Sell R. , Coatanéa E. , Christophe F. (2008).*Important Aspect of Early Design in Mechatronic*, 6th International DAAAM Baltic Conference
- Majjad A. (2009). *Modélisation semi-formelle et cahier des charges*, Laboratoire Lisyc, équipe Susy, UFR sciences.
- Yannou B. (2001). *Préconception de produits*, Laboratoire Productique-Logistique de l'Ecole Centrale Paris
- Yannou B. *Modélisation 3D géométrique et physique pour la préconception de systèmes mécatroniques complexes*, LISSMA Supméca Institut Supérieur de Mécanique de Paris
- Fuchs P. , Moreau G. , Berthoz A. , Vercher J.L. (2007) . *Le traité de la réalité virtuelle*, (4 tomes) , Presses de l'Ecole des Mines de Paris