

GraphiteOne: un diamant pour Linux?

Lorsque **Cascade** est devenu OpenSource, sous le nom d'**OpenCascade**, une certaine agitation a gagné les développeurs. Des projets de grand intérêt se sont annoncés. On en était tout émoustillé. On allait voir ce qu'on allait voir. Oui, mais : aujourd'hui, force est de constater que des projets "grand public" menés à terme, il n'y en a guère. Il y en a cependant au moins un qui mérite qu'on s'y attarde: **GraphiteOne**. C'est ce que l'on va faire : s'attarder, le contempler, et même l'utiliser.

1 En quelques mots :

GraphiteOne Technology est une compagnie sise à Berne, Suisse, ayant pour objectif le support et le pilotage du programme de DAO 3D qui porte son nom, et ce, avec une ambition affirmée: produire une solution supérieure à l'existant. Le groupe s'est donc tourné vers des composants dont les capacités professionnelles alliées à une stabilité reconnue de tous légitiment le choix et la possibilité d'atteindre l'objectif fixé.

Ainsi, **GraphiteOne** recourt à:

- **QT**, bibliothèque en C++ de **Trolltech**, pour construire l'interface: fenêtrage, boîtes de dialogue etc,
- **Hoops**, bibliothèque 3D en C de **Tech Soft America**, pour afficher les données fournies par OpenCascade. Hoops est un produit professionnel utilisé par SolidWorks, CadKey, AutoDesk, Dassault System (CATIA), Bentley etc,. Hoops fournit des visualiseurs de fichiers 3D (*3D Part Viewer*) pour les moteurs Parasolid, Acis et Granite. Il offre également gratuitement pour Linux ses bibliothèques de développement (**HOOPS/3DAF**). Informations complémentaires et téléchargement sur : <http://hoops3d.com>
- **OpenCascade**, bibliothèque en C++, pour effectuer tout ce qui concerne la modélisation, du simple point à la topologie des surfaces, en passant par les vecteurs, les matrices, la représentation mathématique des courbes etc.. L'histoire d'OpenCascade remonte aux années 80, lors de l'annonce faite par **Matra Division** de ce qui allait devenir le fameux système DAO/FAO, **Euclid**, opérationnel en 1987. En 1993, sur l'expertise acquise avec Euclid, Matra créa une plate-forme de développement, **CASCADE** (*Computer Aided Software for Computer Aided Design and Engineering*). C'est en 1999 que cette plate-forme devint Open Source; elle est utilisée par BMW, Mitutoyo, EADS, Alcatel, EDF, CEA, etc. Informations complémentaires et téléchargement sur : <http://opencascade.com>
- **Python**, langage de scripts, pour cimenter et structurer l'ensemble. Le choix d'un langage interprété à la volée, ou compilé en *bytecode*, peut paraître étrange pour un projet de l'envergure de **GraphiteOne**. C'est oublier sa stabilité (Cf **Sketch**, écrit en Python, jamais bloqué en plusieurs années d'utilisation), sa facilité de mise en oeuvre (développeurs dixit), sa capacité à être orienté *Script, Fonction ou Objet*, ou les trois en même temps, ses liens possibles avec d'autres codes... Bref: tout le contraire d'un langage de dilettante.

De la sorte, **GraphiteOne** se présente sous la forme d'un exécutable de 2670 octets, en réalité un *script bash*, et de 189 fichiers python dont la taille d'aucun n'excède 3 ko, correspondant chacun à une commande du programme. A cela s'ajoute encore 3 Mo de python répartis en 428 scripts qui cohabitent avec 125 Mo de bibliothèques *.so. Les scripts sont signés du maître d'oeuvre, **Thomas Schmidt** :

thomas.schmidt@graphiteone-cad.com.

2 L'obtenir... :

Par le passé, **GraphiteOne** s'appelait **TomCad**, comme Linux de Linus s'appelle... Linux : le nom de

l'auteur ne se trouve pas que dans le copyright! Coïncidence ou non, **GraphiteOne** dispose d'un mode *Construction* très efficace, qui fait songer à celui de **TS Cad/Draw**, DAO/Illustration sous Windows, primé en son temps, et que l'on devait à **TommySoftware**.

Curieuse coïncidence patronymique, non? De nos jours, ce shareware couronné est devenu un produit commercial nommé **Cad6**.

Le téléchargement de **GO**, en version 1.2 du 11/02/2004 lors de la rédaction de cet article, s'effectue sur <http://http://www.graphiteone-cad.com/en/index.htm>, section *Download, Free version*.

Avant de procéder au téléchargement proprement dit, un texte relatif à la licence d'utilisation apparaît; il s'agit d'une licence de type **EULA**, c'est-à-dire *Free Personal End-User Licence Agreement*, stipulant que **GraphiteOne** peut être utilisé librement pour un travail personnel, ou dans but d'enseignement ou de recherche, mais aucunement pour une activité générant des profits pécuniaires. Exit donc toute destination commerciale, sauf entente préalable entre les propriétaires et les éventuels utilisateurs.

Après lecture, il suffit de cocher la case d'acceptation des conditions exposées et de cliquer sur > *Proceed*. Les prérequis logiciels sont alors spécifiés, à savoir : une distribution Linux récente, le langage Python et les bibliothèques C/C++ dont certaines sont incompatibles avec **OpenCascade**. Ainsi, conviendra-t'il d'installer les bibliothèques de compatibilité suivantes:

- Sur Suse 8 et 8.1, le paquetage *compat-2002.3.23-1.rpm*.
- Sur Red Hat 8, le paquetage *compat-libstdc++-7.3-2.96.110.rpm*.
- Sur Debian Woody, le paquetage *libstdc++2.9-glibc2.1*.

Si, comme pour le présent test qui a été réalisé sur une Mandrake 9.2, vous répondez à tous les prérequis énoncés, téléchargez les paquetages qui conviennent. Soit, ici, les paquetages pour Suse9.0 et Mandrake 9.1:

- *OpenCascade Libraries*. 25 Mo
- *GraphiteOne Libraries*. 13 Mo
- *GraphiteOne 3D Modelling*. 3 Mo

On remarque la possibilité de télécharger un paquetage *GraphiteOne 2D Drafting*; il n'a d'intérêt que pour ceux qui veulent se satisfaire de la **DAO 2D**. Pour les autres, ils le trouveront inclus le paquetage **3D**.

L'illustration "Téléchargement" montre les différentes distributions Linux supportées.

Linux Distribution	Download
SuSE 8.0/8.1/8.2	OpenCascade Libraries
RedHat 8.0	GraphiteOne Libraries
Debian Woody	
Slackware 8.1/9.0	GraphiteOne 2D Drafting
Mandrake 9.0	or GraphiteOne 3D Modelling
SuSE 9.0	OpenCascade Libraries
Mandrake 9.1	GraphiteOne Libraries
	GraphiteOne 2D Drafting
	or GraphiteOne 3D Modelling
RedHat 7.3	OpenCascade Libraries
	GraphiteOne Libraries
	GraphiteOne 2D Drafting
	or GraphiteOne 3D Modelling
RedHat 9.0	OpenCascade Libraries
	GraphiteOne Libraries
	GraphiteOne 2D Drafting
	or GraphiteOne 3D Modelling

Téléchargement

3 ... et l'installer :

L'installation se passe en douceur, soit en utilisant **Konqueror** et **guRPMI**, soit dans une console en mode *SU* par les commandes:

```
rpm -ihv graphiteone-oc-1.2-1.rpm  
rpm -ihv graphiteone-libs-1.2-1.rpm  
rpm -ihv graphiteone-3d-design-1.2-1.rpm
```

Un utilisateur **Debian**, installera les paquetage rpm grâce à *alien*:

```
alien --install graphiteone-oc-1.2-1.rpm etc.,.
```

Si des difficultés liées aux dépendances non satisfaites se présentent, passez en force en usant des options *--force* et *--nodeps*, de sorte que les commandes ressemblent chacune à ceci:

```
rpm -ihv --force --nodeps graphiteone-libs-1.2-1.rpm.
```

Un sous-répertoire */opt/GraphiteOne* occupant 128,7 Mo sur votre disque dur est alors créé. Le lancement peut maintenant s'opérer en console, ou Alt=F2 dans KDE, par la commande :

```
/opt/GraphiteOne/bin/graphiteone &. Comme il n'est pas pratique d'avoir entrer la commande à chaque fois, il serait judicieux de créer un raccourcis sur le bureau KDE, ou d'écrire un petit script nommé GO, par exemple, et que l'on sauvera dans /usr/local/bin.
```

Contenu du script :

```
#!/bin/sh  
cd /opt/GraphiteOne/bin  
exec graphiteone --ogl=yes
```

Evidemment, ce script doit être rendu exécutable!

On remarque le paramètre *--ogl=yes*, d'une importance capitale: Il permet d'activer l'accélération **OpenGL**. Sans cela, la manipulation d'objet 3D en mode rendu *Ombrage de Phong* devient un véritable calvaire, et sectionne net toute envie d'utiliser le programme. En revanche, avec le recours à **OpenGL**, sur la machine de test disposant d'une part de 512 Mo de ram, et d'autre part d'une carte **n'Vidia GeForce4** nantie de 64 Mo de ram dédiée, le tout servi par les pilotes d'origine **n'Vidia**, l'utilisation de **GraphiteOne** est un régal. Pour tous les connaisseurs de **SoliWorks** (logiciel qui nous sert de référence pour cet article), j'affirme que les zooms animés, les changements de vue en transitions douces, et les déplacements libres à la souris, tant en rotation qu'en panoramique, sont de facture et de qualité équivalentes à ceux dont ils ont l'habitude dans leur programme favori.

NOTA :

Il se peut que sur une distribution très récente, **GraphiteOne** appelle des bibliothèques d'une version différente de celles qui sont installées. Un simple lien symbolique résoudra le problème.

4 Le découvrir :

La tentation est grande de démarrer en 3D sans lire la documentation: il s'agit d'une entreprise vouée à l'échec. En effet, **GraphiteOne** se situe très loin de l'ergonomie intuitive de **SolidWorks**, qui semble détenir la palme en la matière. Aussi, pour faire un test, avant de vouloir créer son propre volume, il semble plus réaliste de chercher à importer une modélisation extérieure et d'observer le comportement du programme.

Cinq formats peuvent être lus:

- **IGES** : Format décharge de données 2D et 3D ASCII, indépendant de la plate-forme, normalisé ANSI. Signifie *Initial Graphics Exchange Standard*. Plusieurs versions existent, avec ce que cela sous-entend

comme risques d'incompatibilité. **GraphiteOne** lit et écrit l'**IGES V5.0**.

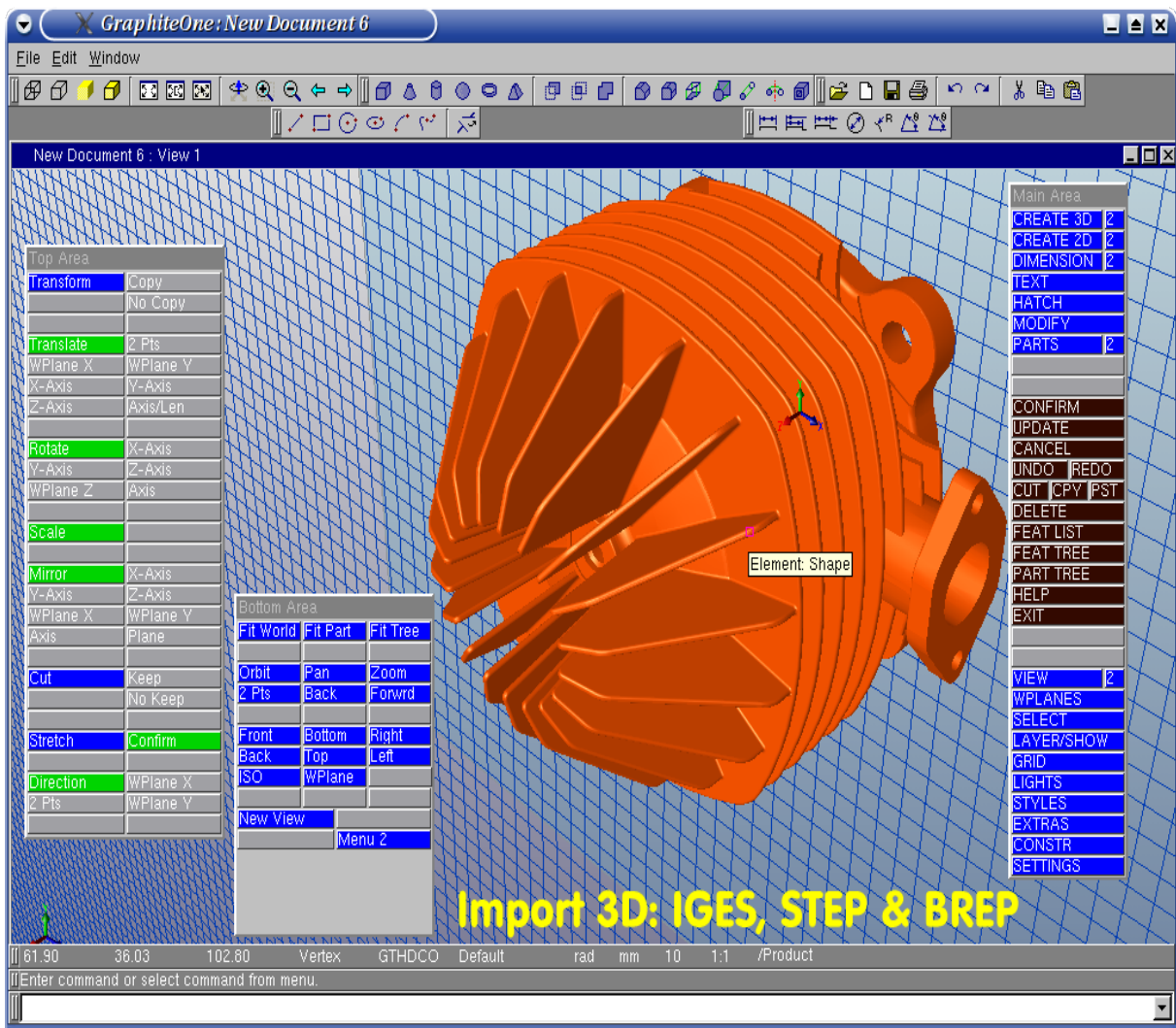
- **STEP** : Format d'échange de données 3D ASCII, normalisé ISO 10303 et pour cette raison, sans doute le meilleur choix. Signifie *Standard for the Exchange of Product Model Data*. Ecrit et lu par la plupart de modeleurs actuels.

- **BREP** : Format 3D ASCII de **Cascade**. Signifie *Boundary REPrésentation*. La communauté du libre semble ne plus devoir maintenir la bibliothèque B-Rep depuis juin 2003, au profit du format GTS : *GNU Trianguled Surfaces*.

- **DXF** : Format d'échange de données 2D et 3D d'**Autodesk**, binaire ou ASCII. Signifie *Drawing Exchange Format*. Il en existe plusieurs versions liées aux différentes moutures d'**AutoCad** : R12, R13, R14, 2000... Incompatibilités possibles. **GraphiteOne** ne lit que le DXF 2D.

- **TCF** : Format 2D et 3D en *xml* de **GraphiteOne**. Signifie *TomCad File*.

A l'usage, on constate que des problèmes de lecture peuvent survenir, selon les formats importés et les paramètres d'importation que l'on impose à **GraphiteOne**. En effet, il est demandé à chaque ouverture d'un fichier autre qu'un **tcf** de se prononcer sur le nombre d'éléments constitutifs de l'objet que l'on charge (élément unique ou multiple), ainsi que sur le type et le niveau de décomposition en facettes ou en solides assemblés. Il convient d'admettre que l'on aura parfois à réitérer les essais face à un modèle récalcitrant, qui, finalement finira par s'afficher comme sur l'illustration : "*Import 3D : IGES, STEP & BREP*".



Tant que nous sommes sur le sujet des formats de fichiers, signalons que **GraphiteOne** sait écrire les formats qu'il lit, auquel s'ajoute:

- **VRML** : Format d'échange de données 3D ASCII très répandu. Il existe en version 1 et 2. **GO** ne connaît que la première. Signifie *Virtual Reality Modeling Language*.

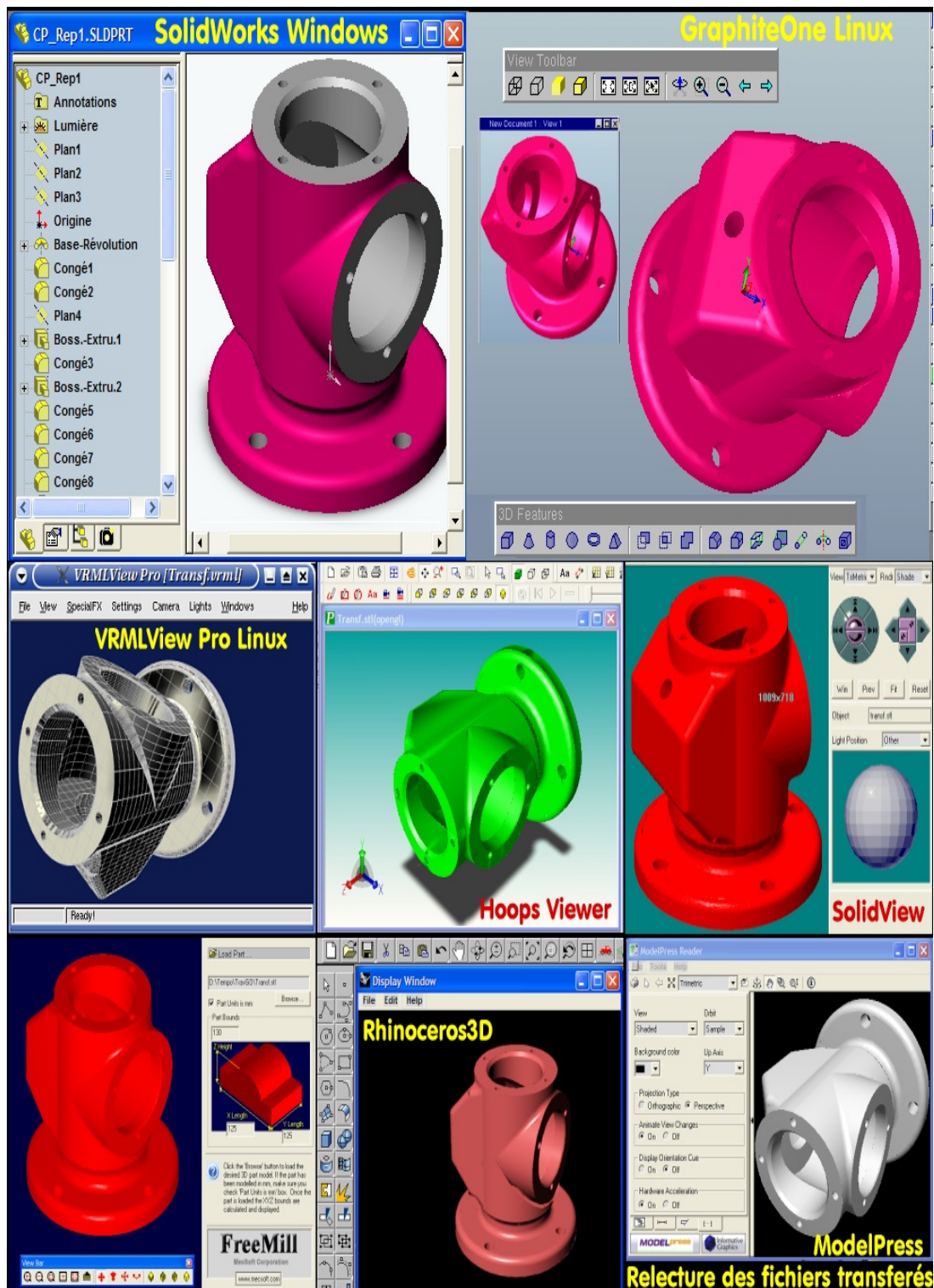
- **STL** : Format d'échange de données 3D ASCII et binaire, utilisé principalement par les machines de

prototypage pour lesquelles il est un standard de fait; est également reconnu dans les système de FAO. Signifie *STereoLithography*. **GO** sauvegarde le STL en ASCII.

Afin d'évaluer la validité des formats ainsi créés, une modélisation **SolidWorks** a été importé en IGES dans **GraphiteOne**, puis sauvegardée par lui en VRML V1, STL, BREP V1, IGES V5 et STEP . De retour sous Windows, il s'est avéré que :

- le **STL** a été lu par Solidview2004, Rhinoceros 3D, Freemill, Hoops 3D Parasolid, Quick 3D Pro, ExoTKAD, Accutrans et Model Press.
- le **VRML** a rencontré plus de difficultés, bien qu' en en-tête il porte la signature de *Matra Division*. Il a cependant été lu par Cinema 4D V6, Solidview2004, SimVRML Viewer, Accutrans et Model Press. Les formats **IGES**, **STEP** et **BREP**, eux, sont lus par les applications déjà citées plus quelques autres. Sous Linux, faute de programmes , l'essai s'est limité à l'ouverture du **VRML** et de l'**IGES**. **VRML** a été lu par Blender et Sim VRMLViewer Pro.
- IGES** a été lu par deux applications métier **DAO/FAO** : gCAD3D et Weber Synergy.

L'illustration "*Relecture des fichiers transférés*" regroupe quelques copies d'écran montrant que la géométrie du modèle est conservée après son passage dans **GraphiteOne**.



On peut raisonnablement en conclure que **GraphiteOne** possède un esprit d'ouverture suffisamment grand pour collaborer avec les autres outils de type **MCAD** (*Mechanical ComputerAided Design*) disponibles sur le marché. Il s'avère donc être une solution envisageable, nécessitant cependant, pour un choix définitif, qu'on le teste plus en profondeur. Ce n'est pas l'objectif ici fixé.

5 L'examiner en surface... :

L'interface est sobre. Elle présente au démarrage un espace de travail, surmonté d'une barre d'icônes et bordé à sa droite de plusieurs palettes de menus littéraux, bien plus explicites que des pictogrammes. Il n'y a rien de figé là-dedans : tout est configurable. Les palettes sont détachables et disposables n'importe où. Un clic droit sur la barre supérieure ouvre un menu contextuel permettant de choisir les palettes que

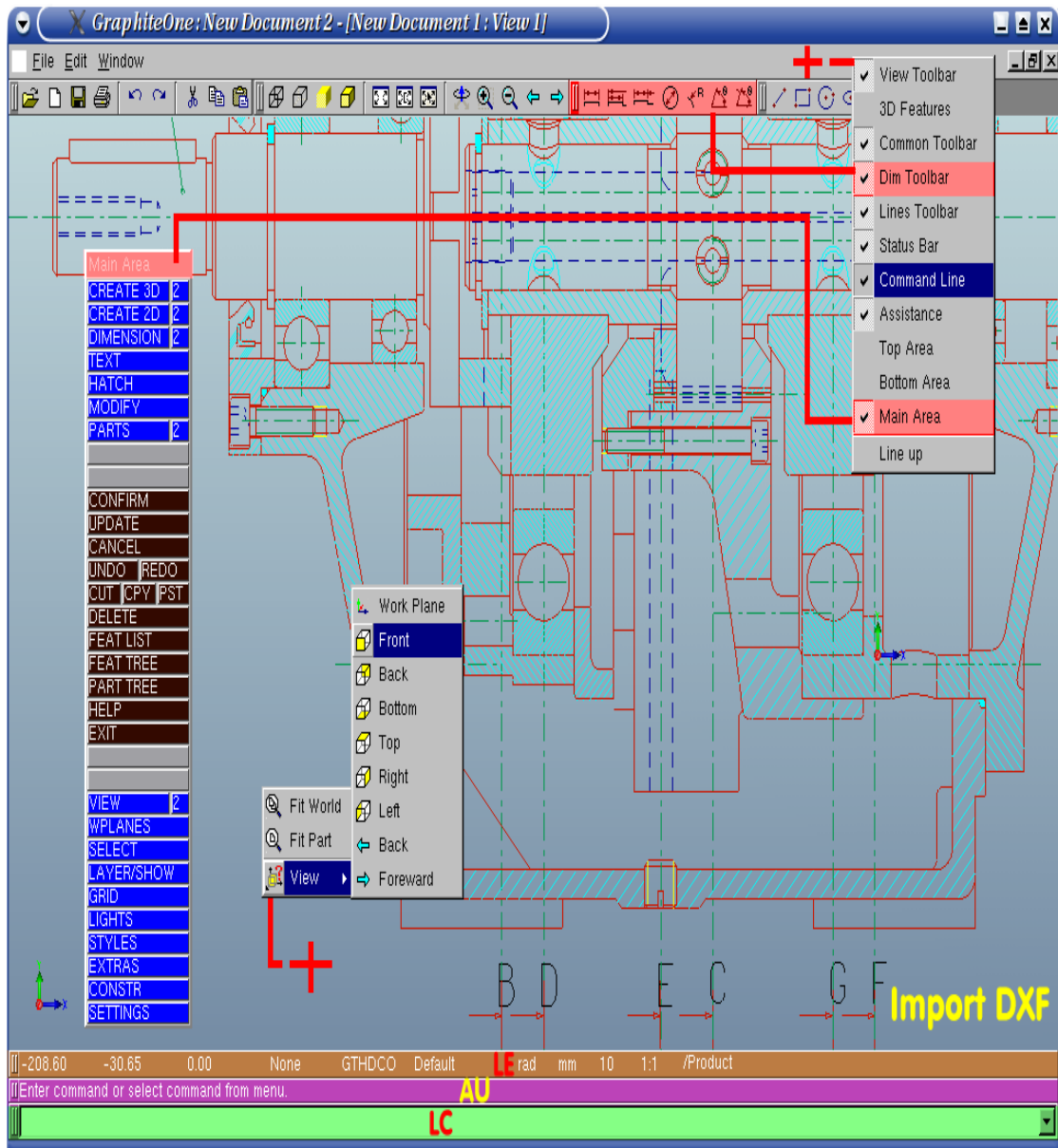
l'on veut ouvrir ou fermer. Un clic droit sur l'espace de travail ouvre un menu différent selon qu'une entité est sélectionnée ou non. Au bas de l'écran, trois lignes de la plus haute importance:

- **LC** : *Ligne de commande*, où seront entrées les commandes, comme sous **AutoCad**, ou les coordonnées et dimensions.

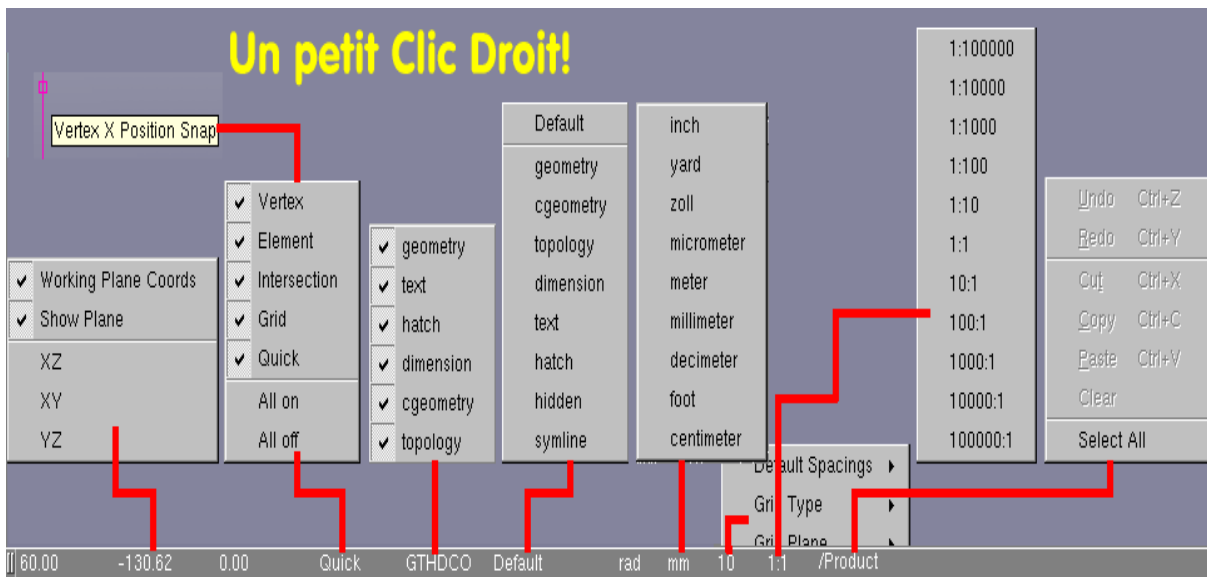
- **AU** : *Assistance utilisateur*. Le système délivre ici différentes informations, erreurs ou attente de commandes.

- **LE** : *Ligne d'état*. Chacune des indications ici inscrites permet l'accès à un menu spécifique si on la sollicite d'un clic droit. C'est là que l'on gère les calques (*Layers*), que l'on paramètre le système de mesure, l' accrochage (*Snap*) etc.,. Il sera question de cette ligne un peu plus avant.

Voir l'illustration : "*Import DXF*".



Les menus contextuels, traditionnellement ouverts d'un clic droit, sont donc présents comme il vient d'être dit. Le clic droit est également la commande par laquelle on ouvre tous les menus dissimulés dans la ligne d'état, derrière les indications : *GTHDCO, Default, Rad, mm, 10, 1:1* etc.,. On y aura souvent recours pour changer de *calque de travail*, le rendre visible ou non, changer le système de mesure, l'échelle, les paramètres de la grille. L'illustration : "*Un petit Clic Droit*" se passe de commentaires.



Il vient d'être fait mention à plusieurs reprises de : *calques*. Un dessin technique est en effet organisé en couches transparentes superposées contenant chacune des entités de type différent. Une habitude issue d'**AutoCad** veut que la décomposition d'un dessin en *calques* (*Layers, couches, plans...*) découle du type de trait que l'on y dépose, auquel, de plus, on affectera une couleur correspondant à une pointe donnée du traceur : *trait plein continu, axe mixte fin* etc.,. Cette habitude est discutable de nos jours, et peut être totalement ignorée par tous ceux qui utiliseront une imprimante plutôt qu'un traceur pour réaliser une sortie papier de leur travail. En tout cas, **GraphiteOne** reste classique dans ce domaine, et c'est ainsi que l'on dispose de *calques* qui contiendront chacun des entités de type différent.

Changer de calques peut s'opérer, comme on l'a vu, d'un clic droit sur l'entrée adéquate de la *ligne d'état*, ou plus rapidement encore, en désignant l'un des quatre types de trait de la palette *Top Area* :

- *Default*, trait plein.
- *Symline*, trait d'axe.
- *Dimension*, cotation.
- *Hidden*, interrompu court des arêtes cachées.

Les attributs de ces types de trait sont modifiables via la fonction *STYLES* de la palette *Main Area*. Voir illustration : "*Calques Rapides*".



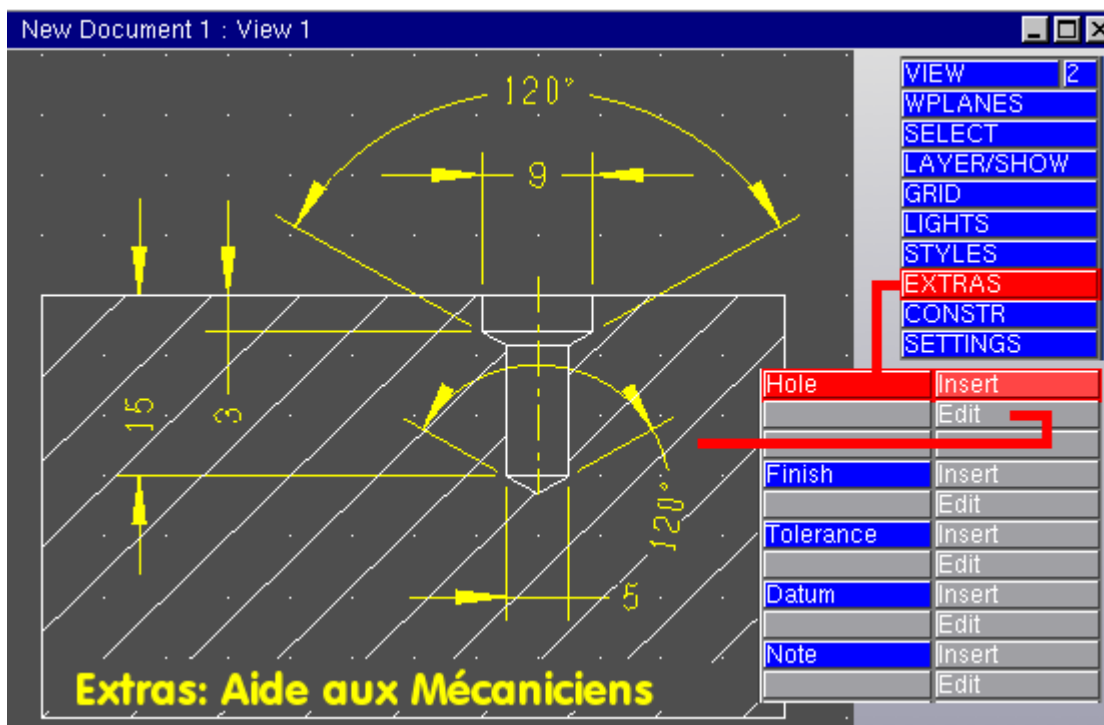
7 ... et en profondeur :

7.1 Les Extras :

L'intérêt d'un programme tient en sa spécificité. Qu'offre-t'il que les autres n'ont pas? Là est la question qui devrait décider de son adoption ou de son abandon

Il existe tellement de programme de **DAO**, qu'il serait téméraire de prétendre que les fonctionnalités de **GraphiteOne** ne se rencontrent pas ailleurs. Toutefois, en tant que programme généraliste 2D/3D orienté *dessin de mécanique*, il dispose d'atouts non négligeables. Ainsi, dans la palette principale *Main Area*, on trouve une entrée *EXTRAS*. Activer celle-ci met à disposition dans la palette *Bottom Area* des fonctions précieuses pour le dessinateur en mécanique qui est confronté à des tâches répétitives et fastidieuses : représentation des perçage et taraudages, inscription des états de surface, des tolérances dimensionnelles et géométriques de forme et de position, des notes diverses... Les fonctions *EXTRAS*, par le biais de boîtes de dialogues dédiées, permet de paramétrer chacun de ces éléments pour le disposer dans le dessin sans

qu'il soit nécessaire de le construire. La fonction *Hole* qui autorise le paramétrage et la mise en place d'une dizaine de perçages, borgnes ou débouchants, est un modèle du genre. Il en va de même pour les états de surfaces normalisés *Finish*, les références *Datum*, ou les annotations *Note* simples ou multilignes. Voir illustrations : "*Extras : Aide aux Mécaniciens*"



7.2 Le mode Construction :

En dessin classique au crayon sur calque, on procède de la manière suivante :

- d'abord, une esquisse (*Draft*) est tracée pour définir les contraintes géométriques et dimensionnelles du plan; les droites sont généralement plus longues que nécessaires, et les arcs tangents ou les raccordements ne sont pas limités à leurs points de tangence.
- ensuite, le dessin définitif est tracé en repassant les traits d'esquisse sur les longueurs voulues.
- enfin, le surplus de l'esquisse sur laquelle on s'est appuyé est effacé.

GraphiteOne reprend cette logique à son compte, similaire, comme il a été dit, à celle de **TS Cad/Draw** et, somme toute, analogue aux *Drawing Aids* de **Varicad**.

Lorsque l'on active le mode *CONSTR* et que l'on sélectionne le calque *geometry* (*Constructions géométriques*), les droites seront infinies et toutes les entités portées sur ce calque ne seront pas imprimées (Fonction non testée). C'est donc là que l'on tracera l'esquisse. Le dessin définitif sera effectué sur le calque *geometry* avec les entités disponibles dans *Top Area*, après avoir choisi le mode *CREATE 2D*.

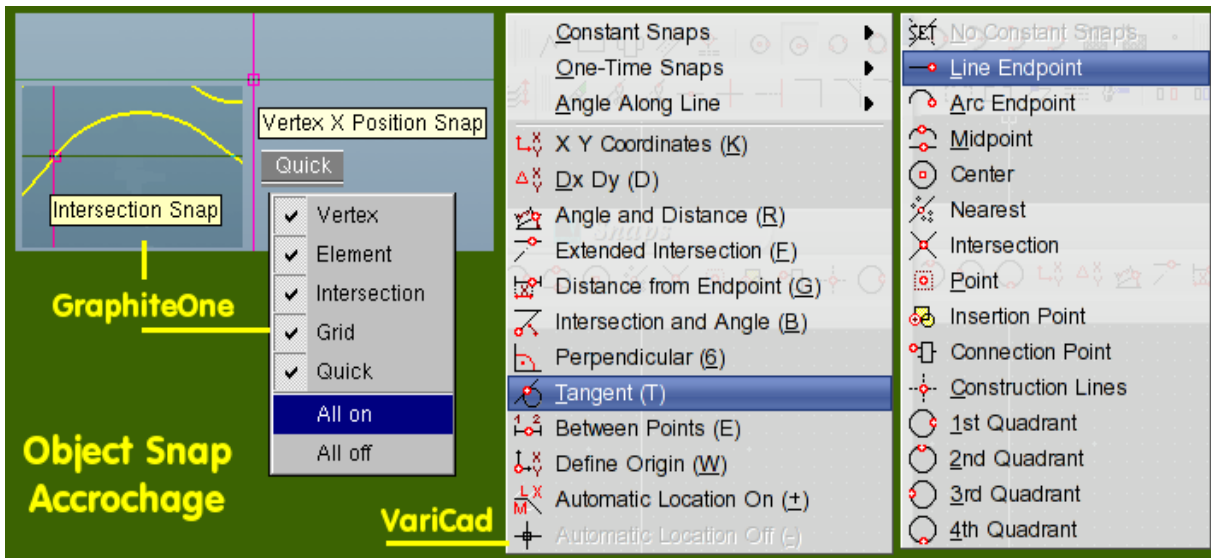
Si les fonctions d'accrochage *Snap*, appelées ailleurs *Osnap* (*Object Snap*), sont activées, le système proposera d'accrocher l'entité en cours de construction à l'un des *points caractéristiques* qu'il connaît, dès lors que le curseur s'en approche. Un *point caractéristique* est une extrémité d'entité, son milieu, le centre d'un cercle, l'intersection de deux entités, un point de la grille magnétique etc., **GraphiteOne** signale par une *infobulle* le point d'accrochage possible. Il suffit de valider la position indiquée si elle convient, ou d'en rechercher une autre.

Beaucoup de programmes disposent d'une batterie de *points caractéristiques* à désigner explicitement pour qu'un point précis soit sélectionné, à moins de disposer d'un mode d'accrochage automatique, auquel cas là aussi le système proposera les accrochages possibles.

En comparaison, **GraphiteOne** semble, apparemment, peu pourvu. Mais l'accrochage *Vertex* inclut extrémité, milieu, centre; *element* est semblable à : *appartenant à*, de **DMT/SolidConcept**; le mode

Quick proposera automatiquement toute les possibilités en les signalant, et l'on se retrouvera dans une situation très proche de celle du mode *Esquisse* de **SolidWorks**, à savoir dans un mode d'*intentions de construction* dans lequel le système devance presque l'utilisateur dans sa démarche.

Voir illustration : "Object Snap, Accrochage", montrant les points caractéristiques reconnus dans **VariCad** et dans **GraphiteOne**.



Après utilisation, on constate que **GraphiteOne** souffre quand même d'un manque dans ce domaine : le point d'accrochage *Tangent*, qui est déficient. En effet, sauf dans le cas de cercles tangents à des droites, où les points de tangences sont perçus comme des points d'intersection, **GraphiteOne** ne trouve pas à s'y accrocher. Il faudra donc recourir à une construction géométrique pour déterminer ces points et les rendre utilisables. Pour cela, on se rappellera que:

- abaisser à partir du centre d'un cercle une droite **Dr1** perpendiculaire à la tangente au cercle, tracera la droite **Dr1** passant par le point de tangence.
- dessiner une droite **Dr2** passant par les centres de deux cercles tangents, tracera la droite **Dr2** passant par le point de tangence des cercles.

7.3 Etude d'un cas :

Un cas concret rendra le discours plus clair. Toutes les explications sont à suivre sur l'illustration : "Tracés d'Arcs Tangents à partir d'une esquisse".

7.3.1 Esquisser :

- Sélectionner le mode *CONSTR*.
- Sélectionner le calque *cgeometry* (ce n'est pas une obligation : une esquisse peut se situer sur n'importe quel calque). Rappelons que l'on sélectionne un calque d'un clic droit sur la mention *Default* de la *ligne d'état*.
- Dans *Bottom Area*, *Menu 2*, sélectionner *Cent Pt/R* et tracer un cercle. Dans *Menu 1*, sélectionner *TanArc/Pt*, désigner un point du cercle (**Pt A**) puis un autre point de l'espace de travail : la droite **D1** est dessinée.
- Dans *Menu 1*, choisir *Perpend*, désigner la droite **D1** n'importe où, puis un autre point quelconque: la droite **D2**, perpendiculaire à **D1** est dessinée.
- Cliquer sur *Menu 2*, choisir *Tan2Pt/R*, désigner le 1^o Cercle (**Pt B**) et **D2** (**Pt C**), déplacer la souris jusqu'à obtenir un cercle dont le rayon convienne, ou entrer une valeur de rayon dans la *ligne de commande*.

7.3.2 Pallier (l'inconvénient) :

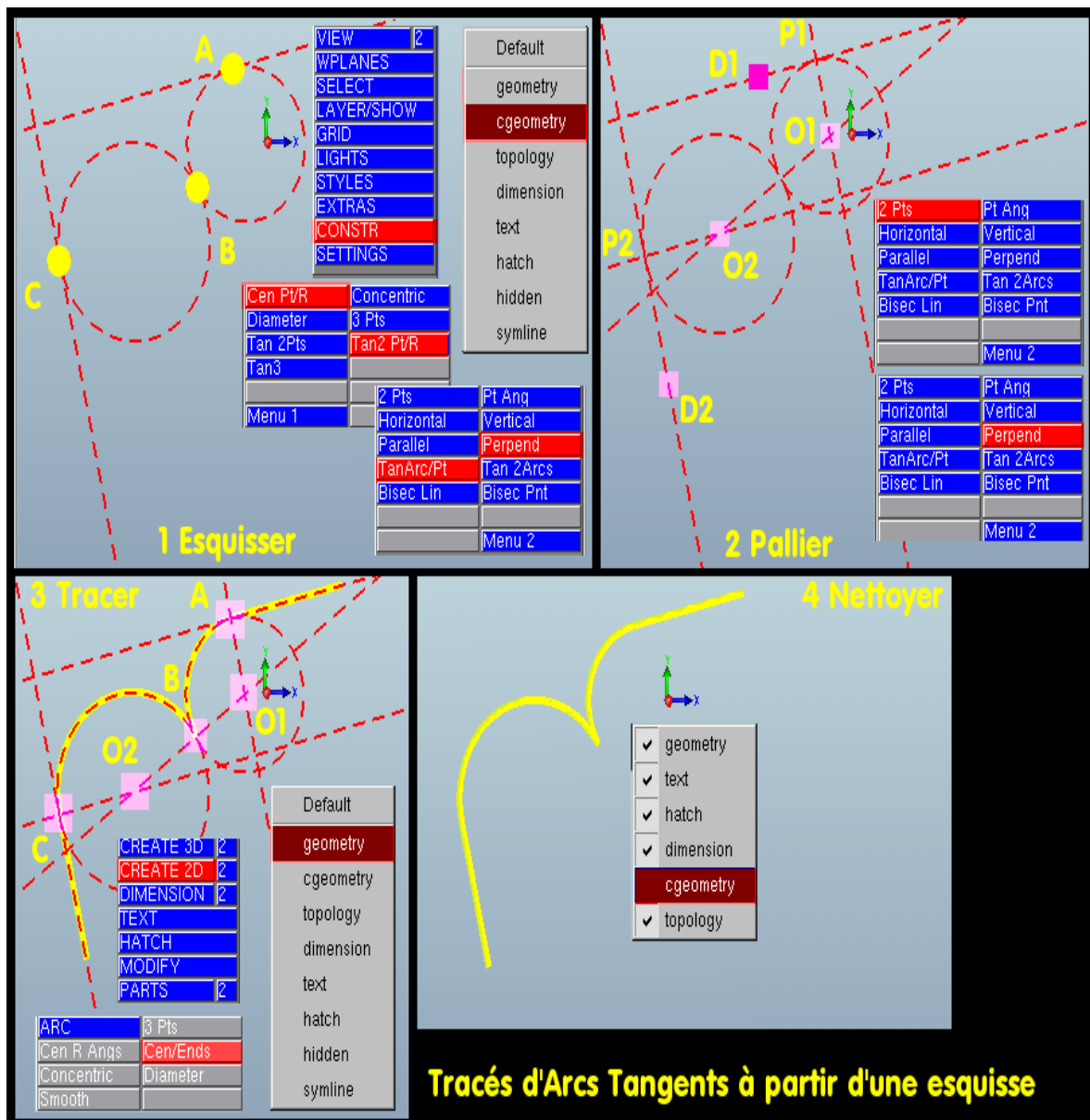
- Dans *Menu 1*, sélectionner *2 Pts*, désigner **O1** et **O2** (le système indique qu'il a trouvé le centre en changeant le curseur en carré magenta et en indiquant dans la ligne d'état : *vertex*). La ligne des centres passant par **B** est dessinée.
- Sélectionner *Perpend*, désigner **D1** et **O1**: **P1** est tracée, elle passe par **A**, point de tangence. Désigner **D2** et **O2**, pour tracer la droite **P2**, qui passe par **C**.

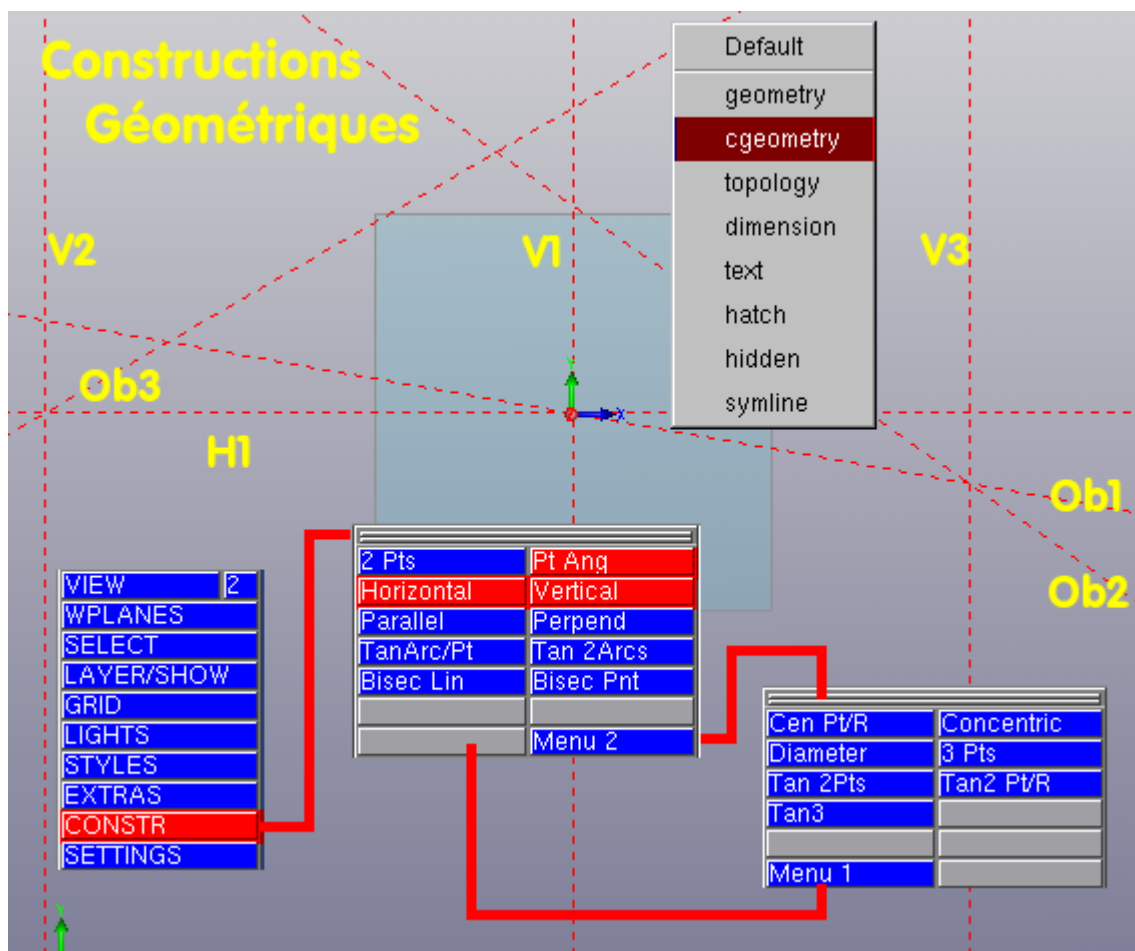
7.3.3 Tracer :

- Sélectionner le calque geometry.
- Sélectionner *CREATE 2D* dans *Main Area*.
- Sélectionner *Cent/Ends* dans *Top Area*.
- Dessiner les arcs en désignant : **O1**, puis **A** et **C**, et **O2**, puis **B** et **C**. Les arcs sont dessinés. Compléter au besoin.

7.3.4 Nettoyer :

Deux solutions sont à disposition : soit effacer l'esquisse, soit rendre invisible le calque *cgeometry*. C'est fini. Tout ceci est beaucoup plus long à décrire qu'à réaliser dès que l'on a compris le principe mis en oeuvre, auquel on recourra dans le didacticiel proposé plus avant.

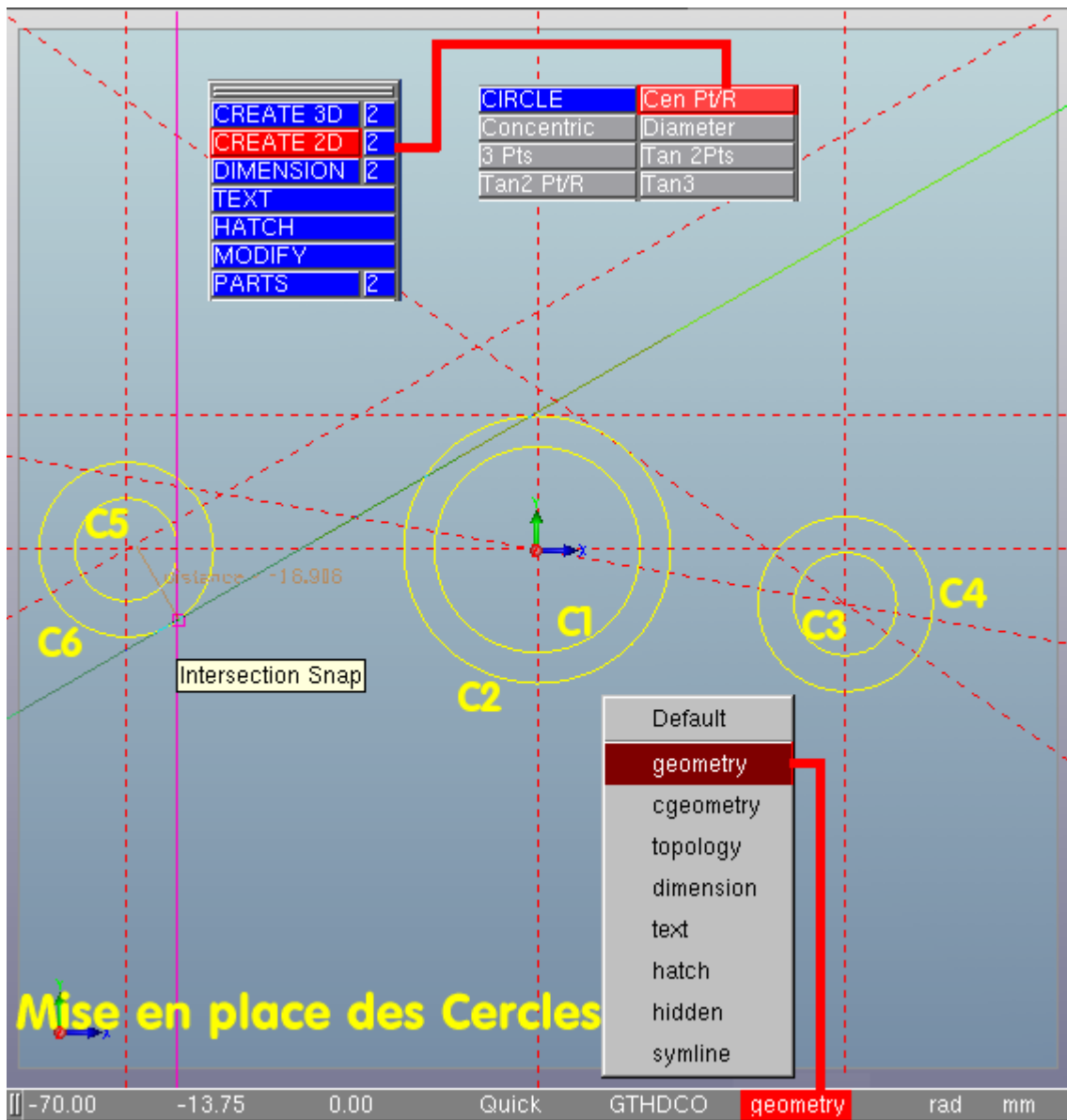




8.2 Mise en place des cercles :

En mode *CREATE 2D*, dans *Top Area*, sélectionner *Cen Pt/R*, et dessiner les cercles **C1** ... **C6** sur le calque *geometry* tels que:

- Cercle **C1**, désigner le point $\langle 0,0 \rangle$, puis entrer $\langle 20 \rangle$, sous-entendu : Rayon=20.
- Cercle **C2**, en entrant $\langle 26 \rangle$, sans changer de commande.
- Sélectionner de nouveau *Cen Pt/R*, designer l'intersection **V3/Ob1**, et entrer successivement $\langle 10 \rangle$ et $\langle 17 \rangle$. Les cercles **C3** et **C4** sont dessinés.
- Sélectionner encore *Cen Pt/R*, designer l'intersection **V2/H11**, et entrer successivement $\langle 10 \rangle$ et $\langle 17 \rangle$. Les cercles **C5** et **C6** sont dessinés. Voir illustration : "Mise en place des Cercles". On appréciera là aussi la *modalité des fonctions* qui permet une productivité certaine.



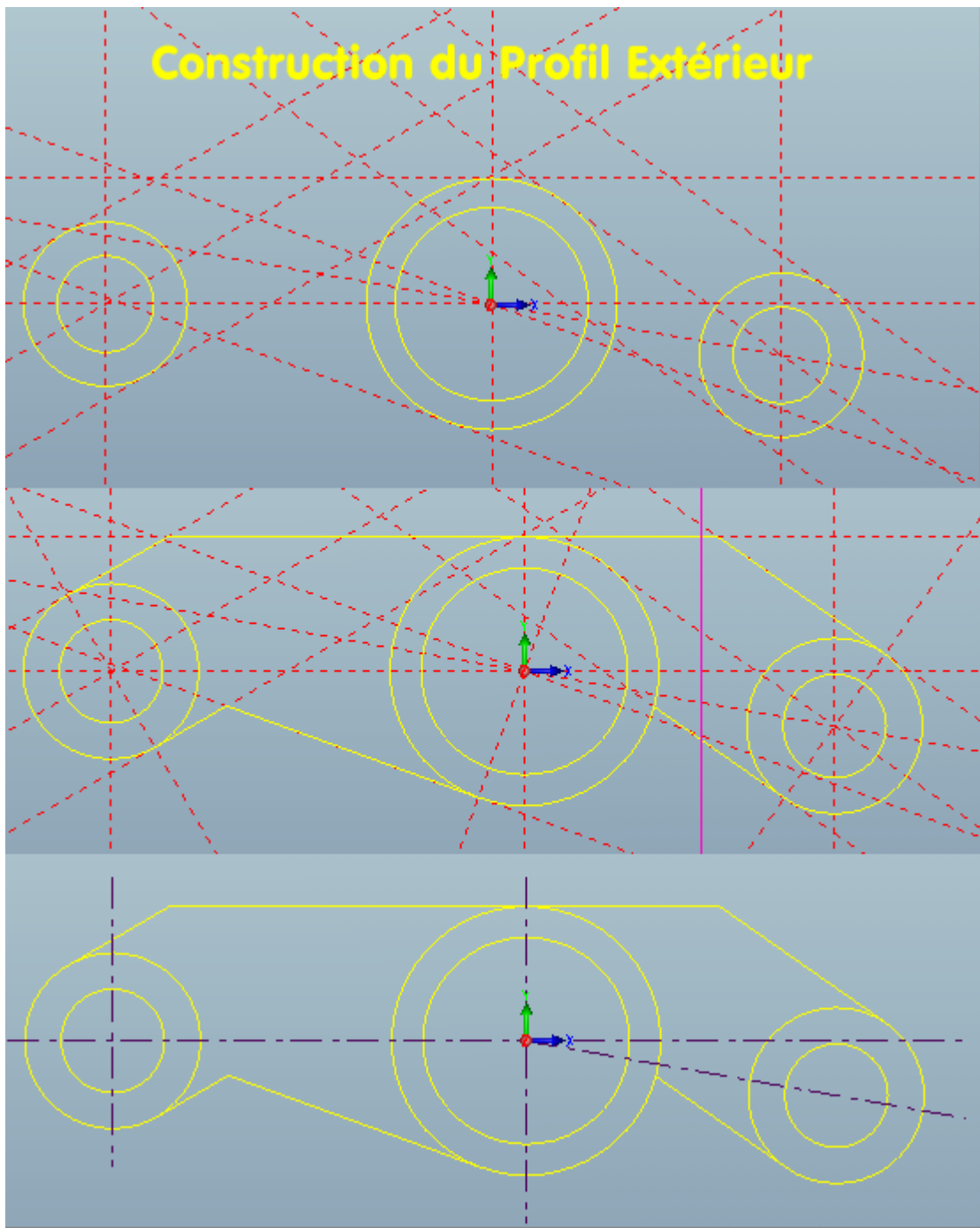
8.3 Tracé du profil extérieur :

Toutes les commandes nécessaires à l'exécution de l'esquisse du profil extérieur et à son tracé définitif ont déjà été utilisées dans les étapes précédentes. Il est donc inutile de les répéter.

Il faut consacrer quatre temps à cette phase:

- Tracé de l'esquisse en mode *CONSTR*, sur le calque *cgeometry*, avec la commande *Parallele*. Bien lire les indications du système sur la *ligne d'assistance*, et entrer les données qu'il réclame pour continuer la manoeuvre.
- Tracé du profil fini en mode *CREATE 2D*, sur le calque *geometry*, avec la commande *2 Pts*, en s'appuyant sur l'esquisse pour définir les accrochages.
- Tracé des axes sur le calque *symline*, grâce à la commande *2 Pts*.
- Suppression de l'esquisse.

Le travail devrait ressembler à celui de l'illustration : "*Construction de Profil extérieur*". Enfin, presque : puisque dans l'exemple présenté, les *attributs* des traits ont été modifiés, on verra comment un peu plus loin.



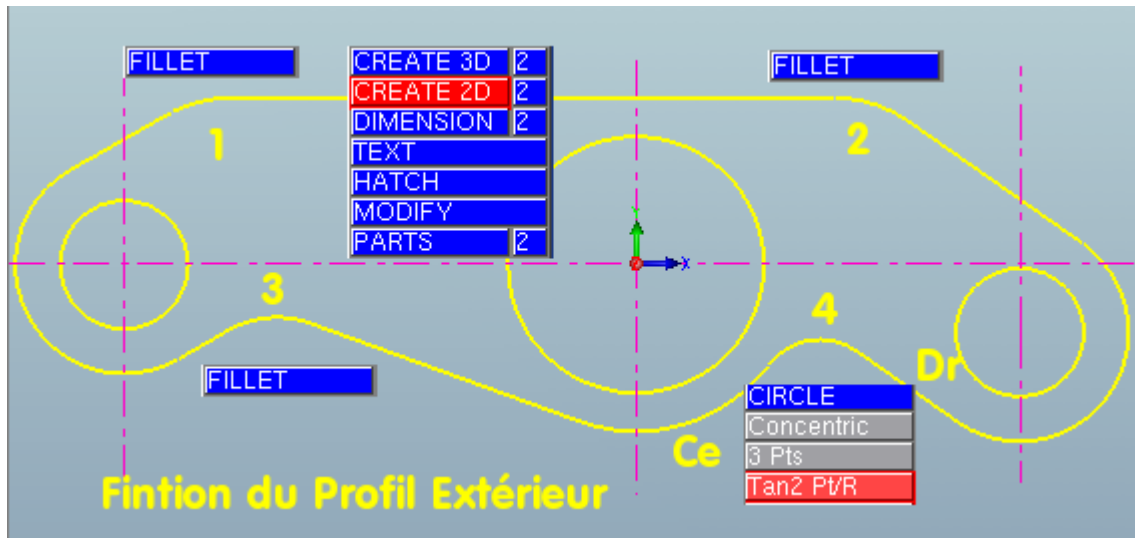
8.4 Finition du profil Extérieur :

On a parlé jusqu'ici de profil fini, mais celui-ci ne sera véritablement fini que lorsque les raccordements auront été ajoutés. Entre deux entités linéaires, des segments donc, le raccordement s'effectue grâce à la commande **FILLET**, qui trace le *congé* (Arc de Raccordement) en éliminant le superflu (*Ragréage*). Entre deux arcs ou un arc et une droite, il n'y a pas possibilité de raccordement avec ragréage, alors que **QCad** fait cela intelligemment, ne ragréant que le segment raccordé au cercle. Dans **GO**, il faut dessiner un cercle tangent, déterminer les points de tangence de la manière déjà décrite, et supprimer le superflu. Donc :

- Pour les raccords 1, 2 et 3, utiliser **CREATE 2D > FILLET**. Entrer la valeur du rayon, cliquer sur le premier segment devant être raccordé, puis sur le second. Cette fonction est *modale*.
- Pour le raccord 4, utiliser **CREATE 2D > Tan2 Pt/R**. Désigner **Dr**, puis **Ce**. Entrer la valeur du rayon, déplacer la souris légèrement. Le cercle tangent est dessiné. Ensuite, en mode **CONSTR**, créer des droites passant par les points de tangence. Enfin, en mode **MODIFY > Cut**, désigner la partie superflue du cercle : elle sera effacée jusqu'aux points de tangence. Avec **DELETE**, effacer les droites de construction. A noter que la fonction *Cut*, agit de manière similaire aux "ciseaux" du mode Esquisse de **SolidWorks** : il

s'agit donc d'un outil puissant.

Voir illustration : "Finition du Profil Extérieur".



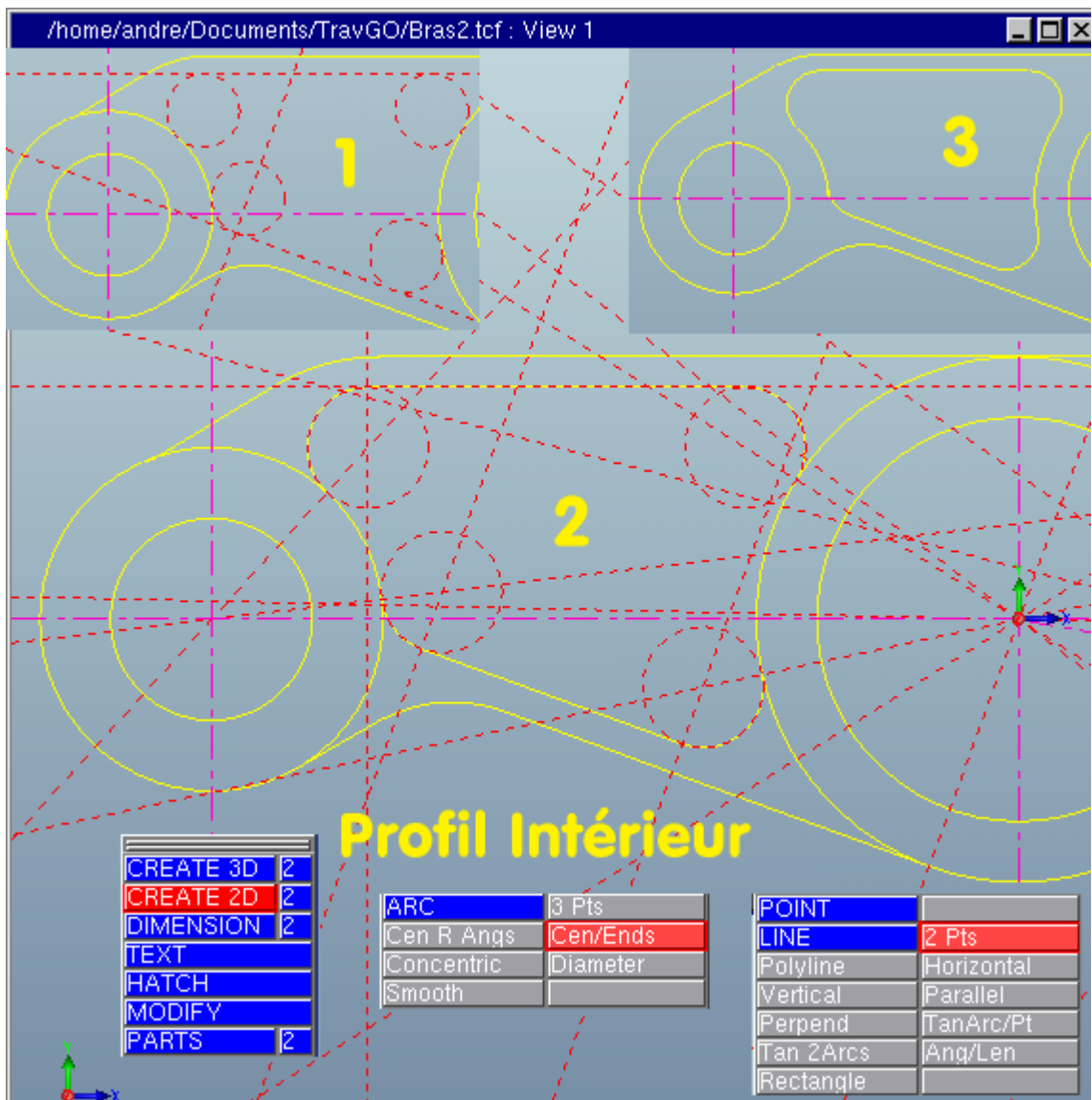
8.5 Profil Intérieur :

Le profil intérieur est constitué de deux *poches* (des évidements) situées de part et d'autre de l'*alésage* central, séparées de l'extérieur par des parois minces de 3 mm.

Sept étapes sont nécessaires:

- création des parois avec des parallèles distantes de 3 mm, en mode *CONSTR*.
- création des cercles tangents de rayon 6 mm.
- création des points de tangence grâce à des droites de construction.
- Création des arcs tangents en mode *CREATE 2D*, en s'appuyant sur l'esquisse.
- Création des segments reliant les arcs, en s'appuyant sur l'esquisse.
- Découpage des portions d'arcs superflues, en mode *MODIFY > Cut*.
- Nettoyage par effacement de l'esquisse.

Voir illustration : "Profil Intérieur", ou la représentation des étapes a été réduite à 3.



8.6 Changer de Style :

En l'absence de paramétrage personnel, les traits affichés à l'écran ou imprimés ont une épaisseur minimale nommée : *épaisseur 0*. On peut vouloir changer cet état de fait afin d'obtenir un résultat conforme aux normes de représentation, où l'épaisseur qualifie les surfaces et les éléments représentés : *surface vue*, *cachée*, *axe*, *délimitation*, *vue fantôme* etc.,

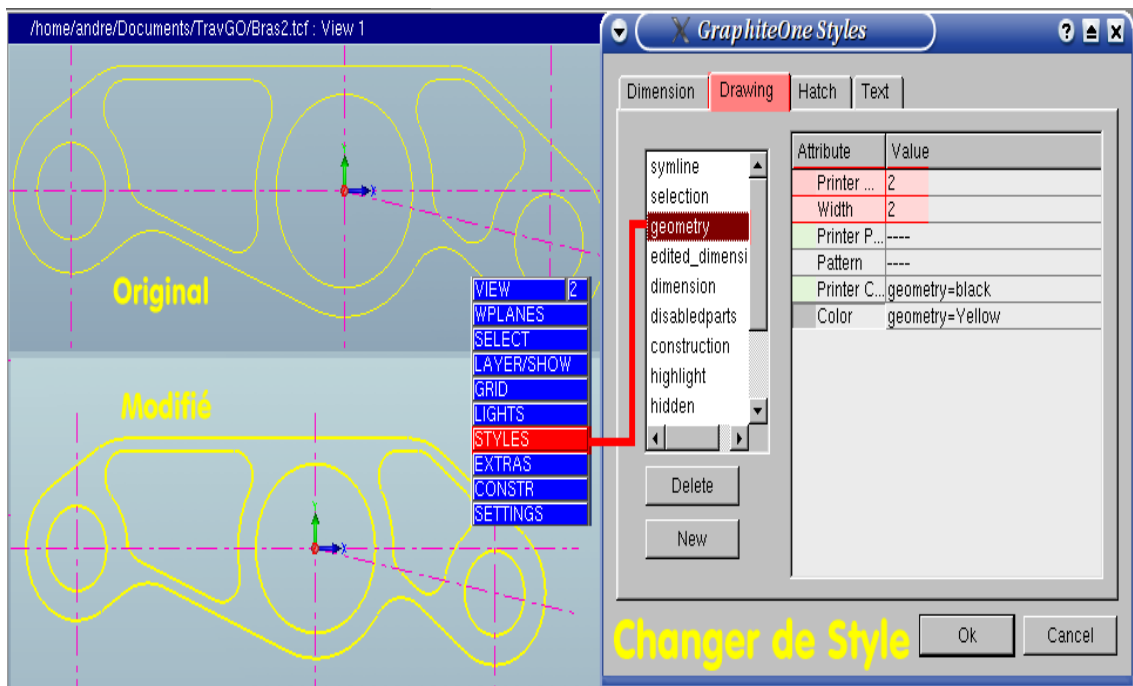
En ce qui nous concerne, le dessin actuel de l'objet ne présente que des surfaces vues : elles doivent donc être dessinées en trait plein d'épaisseur à 0.6 à 0.8.

Sur notre exemple, on va passer d'une épaisseur dite 0 à une épaisseur 2, pour une meilleure lisibilité.

Pour cela, sélectionner **STYLES**, puis **Drawing, geometry**.

Entrer les valeurs **Printer : 2** (inutile si l'on n'imprime pas le dessin), et **Width : 2**, qui conditionne l'aspect à l'écran. On notera que l'on peut modifier ici la couleur vue et la couleur imprimée, le type de trait vu et celui qui sera imprimé.

Notre travail devrait ressembler à celui de l'illustration : "**Changer de Style**".



8.7 Cotation :

Un plan représente généralement un objet destiné à être fabriqué, ce qui ne peut se faire que s'il est dimensionné, c'est-à-dire : *coté*. Par chance, les fonctions de cotation de **GraphiteOne** sont perormantes, souples, paramétrables et très agréables.

On y accède par *DIMENSION* dans *Top Area*. Le système de cotation se choisit en cliquant sur *ANSI*. Apparaissent successivement les différents systèmes disponibles: *ANSI, DIN, JIS, ISO* et *COMPANY*, dont un aperçu est donné à chaque choix dans la fenêtre de visualisation.

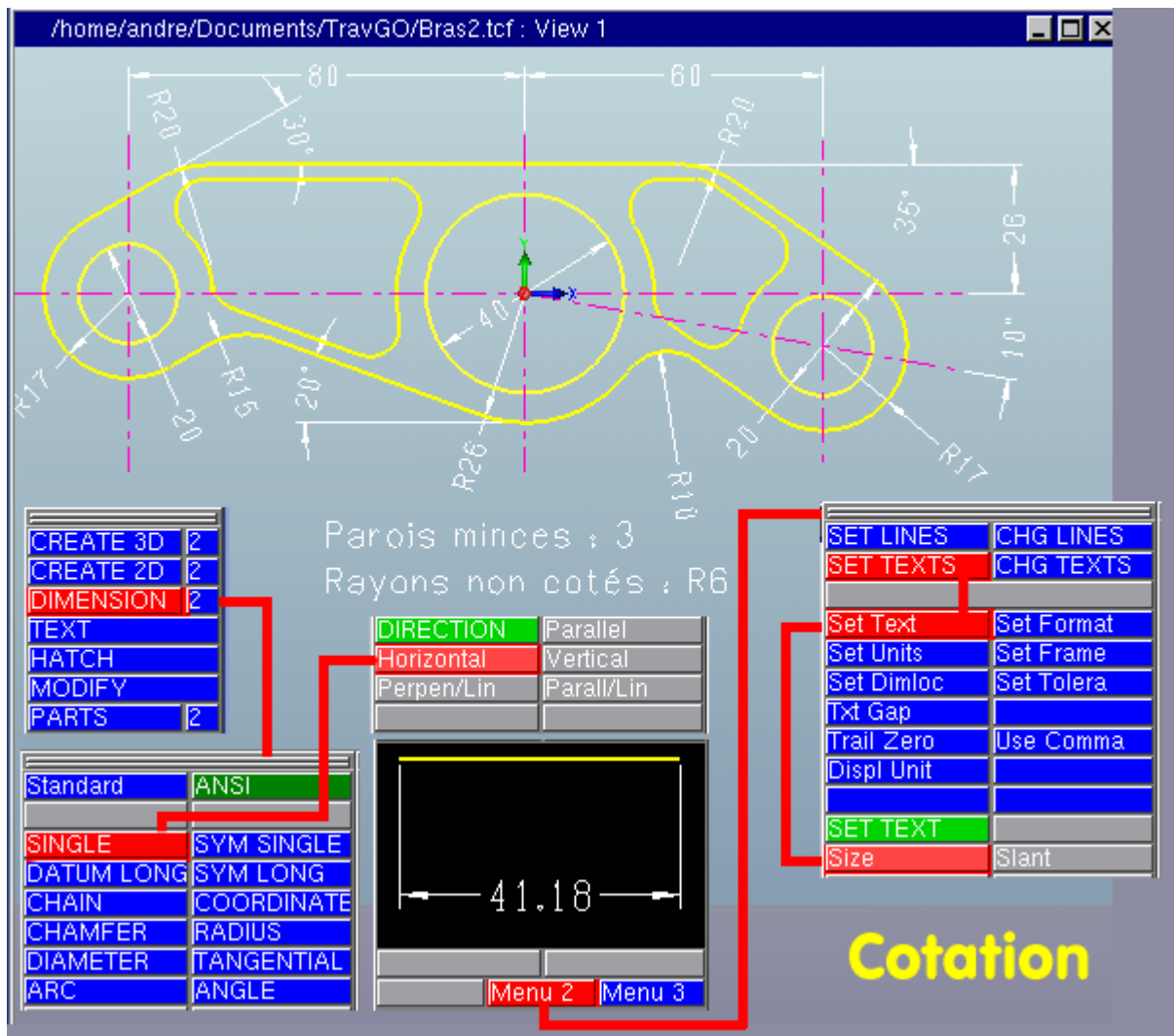
Dans la palette de cotation, *Menu 2* permet de paramétrer le texte de la cote. Le *Menu3* permet d'accéder aux tolérances et autres spécifications d'état de surface, d'annotation, de référence etc.,

Le type de cotation choisie, ici *SINGLE*, permet de définir la *DIRECTION* de la cote : *Horizontal, Vertical, Parallel* etc.,

Les angles, les cercles, les rayons, les chanfreins se cotent avec *ANGLE, DIAMETER, RADUIS, CHAMFER*.

Tout ceci est fort simple et demande seulement un peu de logique. Les cotes s'accrochent automatiquement aux points caractéristiques reconnus par le système, sans qu'il soit besoin de les spécifier. Bien évidemment, le calque actif sera le calque : *dimension*.

Voir résultat de la cotation sur l'illustration : "*Cotation*".



8.8 C'est fini ?

Non : il faudrait encore dessiner la vue en coupe, la hachurer et la coter. Comme il n'y a aucune difficulté à la réaliser, que le procédé de la correspondance des vues a été exposé dans l'article relatif à [QCAD](#), et que chacun devrait en savoir suffisamment maintenant, sa réalisation ne sera pas expliquée. C'est un exercice simple de mise en pratique de l'acquis.

Juste un clin d'oeil pour terminer : [GraphiteOne](#) est-il une solution 3D comprenant un module 2D, ou bien l'inverse? Toujours est-il que les deux méthodes sont à chaque instant disponibles. A preuve : rien ne vous empêche de présenter le plan 2D en vue perspective, ou de profil, ou de derrière, bien que ce soit d'un intérêt très limité. Voir illustration : "2D ou 3D?".

