

UNE SPECIFICATION FONCTIONNELLE DE LA CLASSE VIRTUELLE DES OBJETS CONÇUS DANS UN ATELIER DE DAO

Roger NOUSSI
Etienne NOGNENG

Institut Africain d'informatique
BP 2263
Libreville - GABON

Résumé

En partant d'une méthodologie d'analyse orienté objet, cet article propose une spécification d'un système de DAO utilisant le formalisme des systèmes FP [3] de John Backus. Du point de vue de la spécification cette démarche offre la possibilité d'avoir une description paramétrique des objets conçus et l'utilisation de l'algèbre des systèmes FP comme base théorique pour la manipulation des spécification obtenues. Par rapport à la problématique de la CAO, on peut intégrer à ces spécifications des paramètres relatifs à l'usinage et à l'assemblage.

Mots clés: spécification, analyse orientée objet, systèmes FP, CAO, DAO.

INTRODUCTION

A travers la représentation des connaissances et les méthodologies de conception, la notion d'objet s'affirme comme un élément incontournable dans la conception et la réalisation des systèmes informatiques. Du point de vue de la spécification, les avantages les plus souvent mentionnés [1] [2] des méthodologies orientées objet sont:

- la maîtrise de la complexité et le partage des informations; ces deux avantages sont obtenus à travers les principes de généricité et de modularité.
- la facilité de description paramétrée des systèmes; cette facilité est obtenue à travers l'expression polymorphe des méthodes et des messages.

Ces avantages sont malheureusement atténués par les limitations suivantes:

- l'absence d'une base formelle qui entraîne une limitation de la maîtrise sémantique des spécifications.
- la difficulté d'une spécification dynamique des classes; en effet, dans les systèmes basés sur la notion d'objet, les classes doivent être définies à priorie alors que dans certains domaines comme la CAO, on ne connaît pas à l'avance toutes les caractéristiques de l'objet à concevoir.
- le polymorphisme offert dans les systèmes à objet reste assujéti au système de programmation sous-jacent; par exemple, dans les extensions objet d'un système algorithmique, le polymorphisme se limite à un choix dynamique parmi des méthodes qui doivent être programmées à l'avance.

Ces limitations des systèmes à objet, et surtout leur dépendance par rapport à un système hôte pose pour chaque utilisation le problème du choix adéquat d'un système hôte. Le problème qui nous intéresse ici est celui de la définition d'un formalisme de spécification de logiciel. Sur ce point, les systèmes FP [3][4] possèdent des caractéristiques complémentaires de ceux mentionnées pour la notion d'objet et qui s'avèrent fondamentaux pour la maîtrise de la sémantique des systèmes spécifiés. Ces caractéristiques sont:

- une bonne base formelle fondé sur la "structure algébrique" de l'espace des fonctions, qu'on peut utiliser pour les transformations et la validation des spécifications.
- une expression structurelle des programmes qui peut servir d'outil pour la spécification dynamique des classes d'objets.
- la possibilité d'une spécification paramétrée et dynamique des objets.

L'objectif de ce travail est de présenter une spécification "objecto-fonctionnelle" d'un atelier de CAO. Cet atelier est décrit sous la forme d'un système FP dont les

objets sont les classes identifiées suivant une méthodologie de conception orientée objet. Notre choix se justifie par le fait que les systèmes de CAO ont l'avantage d'avoir une finalité plus précise que les systèmes d'information par exemple; ceci nous permet de mieux cibler nos exemples sans perte de généralité. Par ailleurs, les systèmes de CAO utilisent deux variétés d'objets: les objets connus à priori et dont la structure n'évolue pas et l'objet en cours de conception dont la structure varie pendant le fonctionnement du système. Après une présentation générale des systèmes FP, nous montrons comment ces systèmes peuvent être utilisés pour la spécification de logiciel en l'appliquant au cas d'un atelier de CAO. Cette spécification fait ressortir un avantage fondamental des systèmes FP quant à leur utilisation pour spécifier un système à objet car on arrive par cette démarche à spécifier des objets dont la sémantique varie en cours d'utilisation du système et que nous avons nommé objet virtuelle (objet en cours de conception).

1 - LES SYSTEMES FP

Dans ce paragraphe, nous présentons les concepts de base qui régissent les systèmes FP. Cette présentation a pour objectif principal l'illustration de la clarté de leur base formelle. Le lecteur intéressé pourra consulter [3] et [4] pour plus de détail.

1.1 - LES CONSTITUANTS D'UN SYSTEME FP

Un FP système est composé des éléments suivants:

a) un ensemble O d'objets

Etant donné un ensemble A d'atomes, on peut définir à partir de A un ensemble O d'objets d'un système FP de la façon suivante:

- $A \subset O$

- toute liste dont les éléments appartiennent à O est un élément de O . Les listes sont dénotées par la suite de leurs éléments placés entre \langle et \rangle

- $\Phi \in O$ Φ dénote la liste vide

- $\perp \in O$ \perp peut être présenté comme un élément absorbant pour l'ensemble des fonctions du système FP. Par ailleurs, étant donné un objet

$o = \langle x_1, \dots, x_n \rangle$ pour lequel $\exists i$ tel que $x_i = \perp$, alors $o = \perp$. Ce symbole dénote aussi l'indétermination.

l'ensemble O ainsi présenté est défini comme une liste sur un ensemble d'atomes A ; de ce fait des concepts aussi variés que des vecteurs, des matrices

etc...peuvent être transcrits comme éléments de O , dès que l'ensemble des atomes correspondant est connu.

Ainsi, si on prend $A = \{ 5, a, b, 1, 2, c, 3, 4, v, 6, T, F \}$
 (T et F sont des valeurs de vérité)

$$5 \in O$$

$$a \in O$$

$$\langle 2, 5, a, c \rangle \in O$$

$$\langle c, \langle 3, b, 7 \rangle, \langle 8, 9 \rangle, v \rangle \in O$$

b) Un ensemble F de fonctions primitives

La donnée fondamentale d'un système FP est l'ensemble F des fonctions primitives. Ce sont des opérateurs qui transforment un élément de O en un autre élément de O par le mécanisme d'application que nous présentons ci-dessous. Nous illustrerons la suite de ce paragraphe par un système FP ayant l'ensemble de fonctions primitives suivant:

$$F = \{ +, *, Id, Eq, \} \quad \text{où}$$

- $+$ dénote l'addition habituelle de deux valeurs numériques;
 ainsi l'expression $a+b$ sera notée $+\langle a, b \rangle$
- $*$ dénote la multiplication habituelle de deux valeurs numériques
- Id dénote la fonction identité
- Eq dénote l'égalité

c) Un mécanisme d'application

Le modèle opérationnel d'un système FP est fondé sur l'application de fonctions; ainsi, si f est une fonction et x est un objet,

$$f:x \quad \text{est une application de fonction}$$

cette application de fonction dénote l'objet résultant de la transformation de x par f .

Si on considère un système FP constitué des ensembles O et F ci-dessus

$$+ : \langle 2, 3 \rangle = 5$$

$$Id : x = x \quad \text{quel que soit } x$$

$$Eq : x \begin{cases} = T \text{ si } (x = \langle y, z \rangle \text{ et } y = z) \\ = F \text{ si } (x = \langle y, z \rangle \text{ et } y \neq z) \\ = \perp \text{ si } (x \neq \langle y, z \rangle) \end{cases}$$

d) Un ensemble C de constructeurs de fonction

Les constructeurs de fonction peuvent être considérés comme des fonctions de fonction; c'est à dire des fonctions dont les paramètres sont des fonctions ou des objets et dont le résultat est une fonction. L'ensemble de constructeurs le plus couramment utilisé est

$$C = \{o, [...], \rightarrow., \alpha, / \}$$

Dans cet ensemble on trouve:

- la composition o $f \circ g : x \equiv f:(g:x)$
- la construction $[...]$ $[f_1, f_2 \dots f_n] : x \equiv \langle f_1 : x, f_2 : x, \dots f_n : x \rangle$
- la condition $(p \rightarrow f; g) : x \equiv$ si $(p : x = T)$ alors $f : x$
si $(p : x = F)$ alors $g : x$
- la multi - application α $\alpha f : x \equiv$ si $(x = \phi)$ alors ϕ
si $(x = \langle x_1, x_2, \dots, x_n \rangle)$ alors
 $\langle f : x_1, f : x_2, \dots f : x_n \rangle$
- l'insertion $/$ $/f : x \equiv$ si $(x = \langle x_1 \rangle)$ alors x_1
si $(x = \langle x_1, x_2, \dots, x_n \rangle$ et $n > 1$)
alors
 $f : \langle x_1, /f : \langle x_2, \dots, x_n \rangle \rangle$
dans tous les autres cas, $/f : x \equiv \perp$

e) Un ensemble de définition

Une définition est une association d'un nom à une forme fonctionnelle. On peut définir un programme PS de calcul du produit scalaire de deux vecteurs de la façon suivant:

$$\text{def PS} = (/+)o(\alpha*)o\text{Trans}$$

trans est la fonction qui ayant pour argument un objet représentant une matrice, calcule sa transposée

Un exemple d'application de la fonction PS est

$$\begin{aligned}
\text{PS:} & \langle\langle 1,2,3 \rangle, \langle 6,5,4 \rangle\rangle \\
& = (/+)o(\alpha*)otrans:\langle\langle 1,2,3 \rangle, \langle 6,5,4 \rangle\rangle \\
& = (/+)o(\alpha*):(trans:\langle\langle 1,2,3 \rangle, \langle 6,5,4 \rangle\rangle) \\
& = (/+)o(\alpha*):\langle\langle 1,6 \rangle, \langle 2,5 \rangle, \langle 3,4 \rangle\rangle \\
& = (/+):(\alpha*:\langle\langle 1,6 \rangle, \langle 2,5 \rangle, \langle 3,4 \rangle\rangle) \\
& = (/+):\langle *:\langle 1,6 \rangle, *:\langle 2,5 \rangle, *:\langle 3,4 \rangle\rangle \\
& = (/+):\langle 6, 10, 12 \rangle \\
& = 28
\end{aligned}$$

1.2 - ALGÈBRE DES SYSTÈMES FP

L'algèbre d'un système FP est composé d'un ensemble de lois exprimant l'équivalence entre deux expressions fonctionnelles. Si on considère un système FP comme un environnement de programmation, les règles utilisées peuvent permettre d'écrire et résoudre des équations dont les inconnues sont des programmes. Le passage d'un problème (d'une spécification) à un programme suit une démarche scientifique et non plus "artistique" [5]. Dans cette démarche, le programmeur utilise le langage de programmation lui-même pour démontrer des assertions sur le programme. Ce qui rend la démarche plus naturelle que dans le cas de l'axiomatique de Hoare par exemple.

Quelques lois de l'algèbre d'un système FP

Définition 1

étant donnée deux fonctions f et g ,
 $f \leq g \Leftrightarrow \forall x \in O, f:x = \perp$ ou $f:x = g:x$

Définition 2

étant donnée deux fonctions f et g ,
 $f \equiv g \Leftrightarrow f \leq g$ et $g \leq f$

Nous donnons ci-dessous à titre illustratif quelques unes des lois de l'algèbre des programmes d'un système FP

- $[f_1, f_2, \dots, f_n]og \equiv [f_1og, f_2og, \dots, f_nog]$
- $\alpha fo[g_1, \dots, g_n] \equiv [fog_1, \dots, fog_n]$
- $(p \rightarrow f;g)oh \equiv poh \rightarrow foh;goh$
- $ho(p \rightarrow f;g) \equiv p \rightarrow hof;hog$
- $\perp of \equiv fo\perp \equiv \perp$
- $\alpha(fog) \equiv \alpha fo\alpha g$

1.3 SYSTEME FP VU COMME LANGAGE DE PROGRAMMATION

Un système FP peut être vu comme un langage de programmation si on considère que:

- une fonction est un programme,
- un objet x est le contenu de la mémoire,
- l'expression $f;x$ est le contenu de la mémoire après que f ait été activée avec x en mémoire,
- les fonctions primitives sont les instructions du langage de programmation,
- les constructeurs de fonction sont les structures de contrôle.

1.4 - SYSTEME FP VU COMME FORMALISME DE SPECIFICATION

Une expression FP est un concept structuré, construit à partir de fonctions de base que l'on combine avec les formes fonctionnelles. Dans sa formulation, une expression FP ne fait pas référence à des données ou variables spécifiques. Ceci permet d'utiliser l'algèbre du système FP pour résoudre des équations dont les inconnues sont des programmes.

2 LES ATELIERS DE CAO

L'utilisation des logiciels de DAO à pour finalité la représentation des systèmes mécaniques, des plans architecturaux, des circuits électriques et de façon générale des systèmes conçus selon les normes et les règles du domaine d'application de l'utilisateur [6].

Cependant, la plupart de ces logiciels ne manipulent et reconnaissent essentiellement que des éléments de base de la géométrie: point, droite, arc de cercle, arc d'ellipse.

Quant à l'utilisateur de ces logiciels, il manipule bien au delà des entités citées ci-dessus, un grand nombre d'autres objets issus de son domaine d'activité, et de sa créativité; c'est ce qui explique la spécialisation par domaine d'activité (à quelques exceptions près) des logiciels de CAO [6][10].

Du fait de la complexité du processus de conception, les problèmes de spécification sont fondamentaux dans le domaine de la CAO mais ne sont pas souvent résolus de façon appropriée [8]. Les conséquences d'une mauvaise spécification apparaissent de façon plus évidente dans la partie DAO comme l'illustre l'exemple ci-dessous.

Considérons la section droite d'une poutre rectangulaire. Elle est représentée fig. 1 dans un système de DAO par la surface hachurée délimitée par les segments s1, s2, s3 et s4.

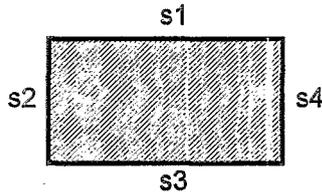


fig. 1

Dans la plupart de logiciels cette section n'est spécifiée que par la donnée de cette liste de segments [9]. L'entité section n'existant pas en tant que telle, on peut imaginer et autoriser la destruction d'un des segments (s3 par exemple).

On obtient ainsi la fig. 2 qui, pour l'utilisateur ne représente plus rien. Le système ne se rend pas compte de cette violation de l'intégrité de l'objet, puisque celui-ci n'est pas représenté de façon spécifique.

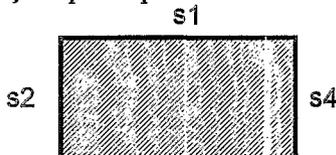


fig. 2

Le fait que la plupart des logiciels de DAO autorisent ce genre d'opération montre bien que la spécification utilisée est incomplète.

L'objectif d'un logiciel de DAO est de donner la représentation la plus fidèle possible de l'objet conçu par l'utilisateur; pour cela, il est important de définir un bon modèle de représentation des objets manipulés, (ce qui explique notre choix de la représentation orientée objet), ainsi qu'un moyen efficace d'utilisation des objets de base offerts par le logiciel pour représenter ceux créés par l'utilisateur, (d'où notre choix du langage FP comme moyen de spécification).

2.1 - Représentation orientée objet d'un atelier de DAO (2D)

a) Les objets manipulés

Les objets manipulés dans un logiciel de DAO sont tous construits à partir:

- des entités géométriques de base que sont le point, le segment de droite, l'arc de cercle et l'arc d'ellipse;
- des courbes d'équation connues (cas de logiciels plus élaborés);
- du texte.

L'utilisateur se sert de ces entités de base pour représenter des objets variés issus de son domaine d'application et de sa créativité. Sur de tels objets l'utilisateur peut avoir besoin de représenter des vues, des coupes, de la cotation ou des hachures; il a aussi besoin de représenter de la nomenclature.

b) Représentation des objets manipulés

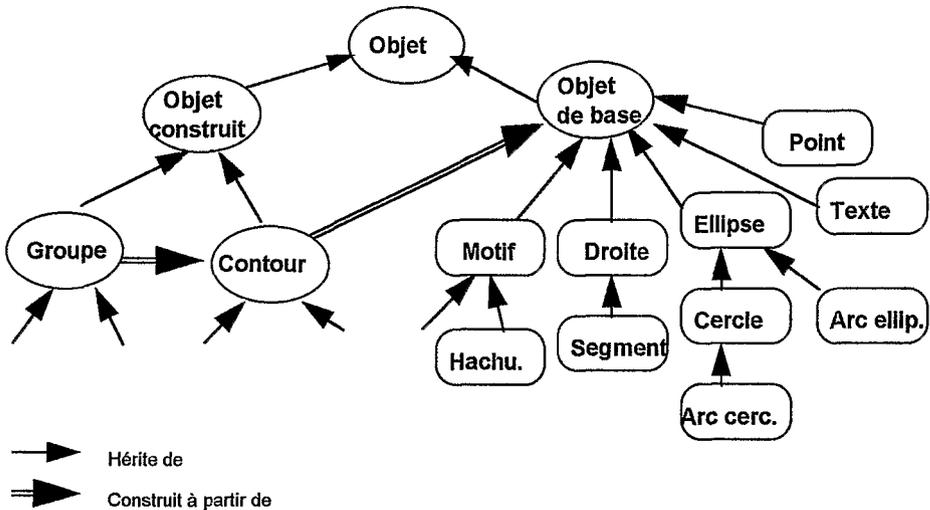


fig. 3

Hiérarchie des classes d'objet

La hiérarchie des classes représentées (fig. 3) ci-dessus fait ressortir les cinq méta-classes suivantes:

- *les objets de base*

Cette méta-classe renferment l'ensemble des entités que propose le logiciel comme base de la construction.

- *les objets construits*

Ce sont les objets qui ne font pas partie de ceux offerts par le logiciel; ils se composent de deux autres méta-classes qui sont la classe contour et la classe groupe.

- *les contours*

Un contour est un objet constitué d'un arc de cercle, d'un arc d'ellipse ou de plusieurs entités géométriques de base qui se touchent. On distingue les contours fermés (délimitant les surfaces) et les contours ouverts. Les vues et les coupes sont des contours

- *les groupes*

Cette classe est faite d'objets constitués de plusieurs contours qui peuvent être soit un assemblage (sens mécanique du terme) soit une pièce. Une côte par exemple est un groupe.

En revenant sur le problème exposé au paragraphe précédent, on constate que la section droite représentée peut être rattachée à la classe des contours fermés; ainsi, la suppression du segment s3 n'est plus anodine; car l'objet résultant ne peut plus appartenir ni à la classe des contours fermés, ni à l'une de ses super-classes.

3 - SYSTEME FP D'UN ATELIER DE DAO (2D)

3.1 - Du modèle objet au système FP

Dans le paragraphe précédent, nous avons présenté le modèle objet d'un atelier de DAO. Ce modèle est le résultat d'une démarche d'analyse et d'une méthode de conception établie et normalisée à savoir l'analyse et la conception orientée objet. Malheureusement, cette méthode ne repose pas sur une base formelle assez robuste permettant notamment de vérifier la complétude et la consistence du modèle proposé. Dans ce paragraphe, nous proposons un système FP équivalent au modèle obtenu ci-dessus. Les règles de passage du modèle objet au système FP sont les suivantes:

1) Une **classe** est équivalente à une **fonction**. En effet de même que les classes sont des moules qui permettent de fabriquer des instances, la fonction est l'opérateur qui permet de fabriquer une valeur en utilisant le mécanisme d'application.

2) L'**instanciation** correspond au mécanisme d'**application**

3) Envoyer un message F à un objet b de la classe G revient à évaluer l'expression (FoG):b

4) la relation de **composition** entre objet est traduite par le **constructeur** [...]

3.2- L'ensemble O des objets du système

Tout objet de notre système peut être décrit par une liste de valeurs réelles; d'où l'ensemble A des atomes: $A=\mathbb{R}$.

On définit par la suite l'ensemble B des entités de base ci-dessous.

$B = \{\text{point, droite, cercle, segment de droite, arc de cercle, texte}\}$.

B est inclus dans O; En effet, les éléments de B pourront être représentés dans O de la façon suivante:

- point: $\langle x_1 \ x_2 \rangle$ $x_i \in A \quad i = 1, 2$

x_1 et x_2 désignent les coordonnées du point.

- droite: $\langle \alpha \ \beta \ \gamma \rangle$ ou $(\alpha, \beta, \gamma) \in \mathbb{R}^3$ avec $\alpha x + \beta y - \gamma = 0$

- cercle:

$\langle \langle x_c \ y_c \rangle \ r \rangle$ vérifiant $(x - x_c)^2 + (y - y_c)^2 - r^2 = 0$

- Segment de droite: $\langle p_1 \ p_2 \rangle$; p_1 et p_2 sont des points qui désignent les extrémités du segment.

- Arc de cercle:

$\langle p_c \ r \ p_1 \ p_2 \rangle$ avec p_c, p_1, p_2 des points et r réel positif. P_c

désigne le centre de l'arc de cercle, r son rayon, P_1 et P_2 ses deux extrémités respectives (première et deuxième) en tournant dans le sens trigonométrique.

- texte: $\langle p \ \alpha \ l \ h \rangle$; le texte est entièrement défini par un rectangle orienté positivement dans le sens de l'écriture; le texte standard étant horizontal de la gauche vers la droite. Par rapport au texte standard, p désigne le coin bas gauche du rectangle, α l'angle d'orientation de l'écriture, l la longueur du texte en nombre de caractères et h la hauteur des caractères.

L'ensemble B est un système générateur de toute construction (2D) réalisable dans un processus de conception [6][8]

3.3 - L'ensemble F des fonctions

Nous remarquons que chacune des entités de l'ensemble B, choisies pour la spécification de notre système peut être entièrement définie par la donnée d'un certain nombre de points. Ainsi, notre présentation se limitera à fournir des fonctions bien étudiées de recherche de ces points pour chaque entité. Par la suite, une simple composition de ces fonctions nous permettra de préciser chaque entité, et partant de là, chaque objet représenté.

Dans les exemples qui suivent, P désigne l'ensemble des points et R l'ensemble des réels

Parmi ces fonctions, on distingue:

- les fonctions de construction des droites et des cercles

Exemple: la fonction droite ci-dessous

D r o i t e

$$P^2 \rightarrow R^3$$

$$(p_1, p_2) \mapsto (\alpha, \beta, \gamma) \quad \text{avec} \quad \alpha x + \beta y - \gamma = 0$$

- les fonctions de construction des points (par digitalisation, intersection de deux entités, point particulier d'une entité)

Exemple: la fonction inter ci-dessous

I n t e r

$$B \times B \rightarrow P$$

(D1, D2) \mapsto p p est le point intersection des droites D1 et D2
 (C1, C2) \mapsto p p est le point intersection des cercles C1 et C2
 (C, D) \mapsto p p est le point intersection de la droite D avec le cercle C.

- les fonctions de positionnement d'entité

Exemple: La fonction ortho

o r t h o

$$B \rightarrow R^2$$

$$D(\alpha, \beta, \gamma) \mapsto v \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad \text{Vecteur unitaire orthogonale à}$$

la droite D

- les fonctions de duplication d'entité

Exemple: sym(p)

s y m (p)

$$O \rightarrow O$$

$x \mapsto x'$ avec x' symétrie de x par rapport au point p

- la fonction de destruction.

3.4 - L'ensemble C de compositeurs de fonctions:

$$C = \{\text{distant}, \text{direction}, \text{symétrie}, \text{rotation}\}$$

D i s t a n t

$$R \cup P \rightarrow F$$

$$a \mapsto \text{distant}(a)$$

a est un réel qui représente une valeur

de distance

$$p \mapsto \text{distant}(p) \quad p \text{ représente le centre d'un cercle}$$

D i r e c t i o n

$$R \rightarrow F$$

$$a \mapsto \text{direction}(a)$$

a est un réel qui représente une valeur

d'angle

S y m é t r i e

$$B \rightarrow F$$

$$p \mapsto \text{sym}(p)$$

fonction symétrie par rapport au point p

$$D \mapsto \text{sym}(D)$$

fonction symétrie par rapport à la droite

D

R o t a t i o n

$$P \times R \rightarrow F$$

$$(p, a) \mapsto \text{rot}(p, a)$$

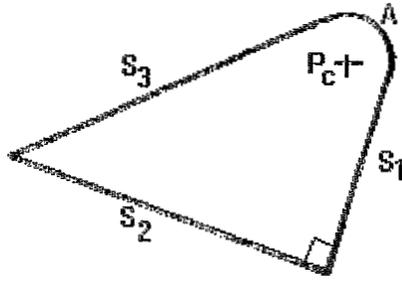
fonction rotation d'angle a autour du

point p

4 - EXEMPLE

4.1 - Le problème

On se donne à dessiner l'objet représenté figure ci-dessous: sorte de triangle rectangle avec un raccordement à l'un de ses sommets.



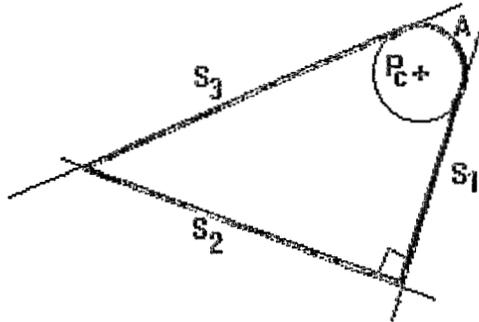
Les données sont les suivantes:

- 1) L'arc de cercle de raccordement est de rayon r .
- 2) Le segment S_2 forme un angle droit avec S_1
- 3) Le segment S_2 se trouve à une distance d du centre du cercle.
- 4) Le segment S_2 est de longueur L .

Ces données sont suffisantes pour construire cet objet.

La saisie de ces données peut se faire par le biais d'un sous langage [11] adapté au domaine d'application.

4.2 - Une procédure de construction:



- 1) Générer le cercle: $C = \text{distant}(r)(\text{digit})$
% digit retourne le point position de la souris
- 2) Créer la droite support de S_1 : $D1 = \text{droite}(\text{proj}_p(\text{digit}, C))$
- 3) Créer la droite support de S_2 :
 $D2 = \text{droite}(\text{proj}_o(\text{distant}(d)(\text{centre}(C)), D1))$
- 4) Déterminer le point de rencontre de S_1 et S_2 : $P1 = \text{inter}(D1, D2)$.
- 5) Déterminer le point de rencontre de S_2 et S_3 :
 $P2 = \text{inter}(D2, \text{distant}(L)(P1))$.
- 6) Déterminer droite support de S_3 : $D3 = \text{droite}(\text{proj}_p(P2, C))$.
- 7) Générer l'arc utile $A = \text{arc}(\text{inter}(D1, C), \text{inter}(D3, C), C)$.
- 8) Générer le segment $S1 = \text{segment}(\text{inter}(D1, D2), \text{inter}(C, D1), D1)$.
- 9) Générer le segment $S2 = \text{segment}(\text{inter}(D1, D2), \text{inter}(D2, D3), D2)$.

10) Générer le segment $S3 = \text{segment}(\text{inter}(D2, D3), \text{inter}(C, D3), D3)$.

En reprenant les points 7, 8, 9 et 10 de la procédure de construction ci-dessus, on constate qu'on obtient un objet ayant les attributs A, S1, S2, S3; cet objet est rattaché à la méta-classe des contours fermés. A cet objet on peut associer la spécification fonctionnelle suivante:

```
[arc(inter(D1,C),inter(D3,C),C),  
segment(inter(D1,D2),inter(C,D1),D1),  
segment(inter(D1,D2),inter(D2,D3),D2),  
segment(inter(D2,D3),inter(C,D3),D3)]
```

CONCLUSION

Nous avons proposé dans cet article une spécification objet-fonctionnelle d'un atelier de DAO; dans ce travail nous avons tiré de la démarche objet la méthode d'analyse et la capture de la sémantique des objets conçus. L'aspect fonctionnelle nous permet d'avoir une spécification paramétrée et nous offre la possibilité d'utiliser l'algèbre des systèmes FP comme base théorique pour la manipulation des spécifications obtenues. D'autre part l'utilisation d'un formalisme de type fonctionnel laisse entrevoir la perspective de réaliser dans le treilli des objets du système un partage d'informations plus dynamique que celui autorisé par le mécanisme d'héritage.

BIBLIOGRPHIE

- [1] *Conception orienté objet et application*
par GRADY Booch ed. Addison Wesley - 1992
- [2] *Conception et programmation par objet*
par J. FERBER ed. Hermes 1992
- [3] *Can programming be liberated from the Von Neuman style ?
A functional style and its algebra of program.*
par J. BACKUS Com. of the ACM août 1978 vol 21 n° 8
- [4] *Theoretical foundations of computer science*
par Dino MANDRIOLI et Carlo GHEZZI
ed. John Willey and sons 1987

- [5] *The science of programming*
par D. GRIES ed. Springer - Verlag
- [6] *Techniques graphiques interactives et CAO*
par Y. GARDAN, M. LUCAS ed. HERMES
- [7] *Participation à la conception du logiciel LOTREC (de DAO)
pour le développement de SICAM.*
par E. NOGNENG Rapport de stage
- [8] *Définition de systèmes mécaniques sur microordinateurs:
Réalisation d'une structure informatique adaptée.*
par E. NOGNENG Thèse
- [9] *C.A.O. en mécanique (conseils essentiels)*
La CAO potentialités et composants des systèmes
par la Fédération des Industries Mécaniques et Transformatrices
des Métaux (F.I.M.T.M) 1985
- [10] *Introduction à la CAO 2° édition*
par N. GIAMBIASI, J. Claude RAULT, J. Claude SABONNADIÈRE
ed. HERMES
- [11] *The significance of sublanguage for automatic translation
in machine translation,*
edited by Sergei NIRENBURG
Studies in natural language processing