
TITRE : CALCULETTE ORIENTÉE STYLO POUR MACHINE DE POCHE

Laurent Pasquer¹, Éric Anquetil², Guy Lorette³

¹ : Irisa/INRIA

² : Irisa/INSA de Rennes

³ : Irisa/Université de Rennes 1

Laurent.Pasquer@irisa.fr

IRISA / INRIA Rennes

Campus Universitaire de Beaulieu

Avenue du Général Leclerc

35042 RENNES Cedex - France

Résumé — Dans cet article nous présentons l'état d'avancement d'une calculette pour machine de poche (PDA) basée sur la simple utilisation du stylo. Cette calculette est issue du portage de la boîte à outils logicielle **DigiCarbon** sur machine de poche. **DigiCarbon** a été conçu pour faciliter la construction d'interfaces d'édition de documents structurés simples. Ces interfaces s'appuient sur une interaction complète au stylo : la saisie des symboles se fait par l'écriture, et leur manipulation via des gestes graphiques. D'autre part, **DigiCarbon** se base sur une description grammaticale du document à manipuler. Ainsi, fabriquer une interface pour un nouveau document consiste simplement à définir la structure de ce document, ainsi que les différents symboles et gestes qui serviront à le manipuler.

1. Introduction

Les interfaces orientées stylo constituent une nouvelle modalité d'interaction entre l'homme et la machine. Elles sont de plus en plus utilisées du fait du développement de machines de plus en plus petites (tablet-pc, assistants numériques, téléphones...).

Ces interfaces devraient bientôt offrir à l'utilisateur la possibilité d'effectuer à main levée à l'aide d'un stylet :

1. des saisies naturelles (il peut directement écrire des symboles sur l'écran),
2. des dessins de graphe, croquis, etc.
3. des commandes gestuelles intuitives (sélection, suppression, insertion...).

Pour permettre de construire facilement des interfaces orientées stylo, nous avons conçu la boîte à outils logicielle **DigiCarbon**, programmée en Java.

DigiCarbon permet de construire facilement des interfaces d'édition et de manipulation de documents structurés simples, ces opérations se réalisant à l'aide d'un stylo (commande gestuelle et saisie naturelle).

DigiCarbon est basé sur une double architecture : d'un côté une modélisation géométrique des différents symboles et gestes graphiques possibles, et de l'autre une modélisation structurelle du document à éditer.

DigiCarbon est la suite de nos travaux précédemment menés sur le système SPI : Système de Perception et d'Interprétation [1]. Il est à la fois une version dynamique de ce système (SPI se limitait à l'analyse statique de toute l'écriture) : l'interprétation se fait *au fur et à mesure* des tracés. Mais il est pour l'instant une version simplifiée de SPI : la gestion de plusieurs contextes organisés n'est pas encore implémentée.

Nous avons construit un prototype d'interface de saisie de formules mathématiques simples (calculatrice) en nous appuyant sur DigiCarbon. Ce prototype est en cours de portage sur la machine virtuelle SuperWaba [2]. Ceci nous permettra de le faire fonctionner sur les plateformes PalmOS et PocketPC.

Dans la suite de cet article, nous allons tout d'abord présenter la structure de DigiCarbon, nous décrirons ensuite son utilisation pour construire la calculatrice ainsi que le fonctionnement de celle-ci. Enfin, nous concluons en présentant les principales difficultés du portage vers SuperWaba.

2. DigiCarbon

La boîte à outils DigiCarbon s'appuie sur une représentation du document en deux descriptions complémentaires au format XML :

- d'une part, la description graphique des différents symboles et gestes possibles,
- d'autre part, la description grammaticale de la structure du document.

Nous allons maintenant décrire plus en détail ces deux descriptions.

2.1 Description des symboles et des gestes

Pour permettre à l'utilisateur d'interagir avec l'interface stylo, il faut que celle-ci puisse identifier les différents tracés qu'il va faire.

Il faut donc que le concepteur de l'interface décrive comment peuvent être tracés les différents symboles possibles, ainsi que les différents gestes.

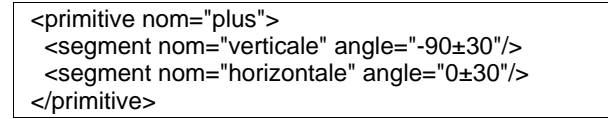
Cette description est basée sur une représentation géométrique simple des tracés. La **figure 1** présente la description du symbole '+'.


Figure 1 — Exemple de description d'un symbole

Tout tracé dessiné par l'utilisateur est segmenté en primitives graphiques, les meilleurs appariements entre ces primitives et les différentes descriptions sont conservés pour être analysés *en-contexte*.

Ainsi, la décision de reconnaissance d'un tracé comme un symbole ou une commande gestuelle est repoussée pour être prise en confrontant ces différentes hypothèses d'identification avec ce que le contexte structurel propose.

2.2 Description de la structure

Pour fournir à l'utilisateur des moyens de construire et manipuler un document structuré, DigiCarbon a besoin que le concepteur de l'interface décrive précisément comment se font toutes ces opérations.

La première étape est de construire la grammaire du document. Cette description est elle-aussi au format XML. La **figure 2** présente la description d'une formule de type somme 'Σ'.

```
<forme nom="somme">
  <composition>
    <formes-obligatoires>
<ref-forme nom-forme="sigma" nom="sigma"/>
    </formes-obligatoires>
    <formes-facultatives>
<ref-forme nom-forme="formule" nom="dedans"/>
<ref-positionnement nom="à droite"
  liste="dedans,sigma"/>
<ref-forme nom-forme="formule" nom="en_haut"/>
<ref-positionnement nom="pos_au_dessus"
  liste="en_haut,sigma"/>
<ref-forme nom-forme="formule" nom="en_bas"/>
<ref-positionnement nom="pos_en_dessous"
  liste="en_bas,sigma"/>
    </formes-facultatives>
  </forme>
```

Figure 2 — Exemple de description d'une formule 'Σ'

Cette description permet de préciser pour chaque objet du document : de quels sous-objets il est composé (*ref-forme*), et aussi comment sont agencés ces objets (*ref-positionnement*).

Ainsi, DigiCarbon sait quels sont les symboles possibles dans un document en cours d'édition. Il peut

donc les *prédire* et de plus, grâce à la description de leur agencement, il sait où l'utilisateur va pouvoir les tracer.

En plus de cette description de la construction du document, des règles de réécriture décrivent les modifications possibles du document, ainsi que les gestes qui les déclenchent.

Ainsi, lorsqu'un tracé a été analysé, DigiCarbon peut savoir si c'est un geste ou un symbole, et va décider lequel en fonction de la justesse de son positionnement dans le document, cette justesse étant fusionnée avec la qualité de son appariement entre le tracé et la forme.

3. Calculette

Le prototype calculette est basé sur une version simpliste d'une possible interface de saisie de formules mathématiques.

Les documents possibles sont composés des symboles et formes suivants :

- les dix chiffres et la virgule,
- les quatre opérations de base : + - / *
- la possibilité de construire des fractions et des sommes,
- un opérateur pour calculer le résultat : =

Quant aux gestes, il sont pour l'instant très simples :

- gestes pour annuler/refaire,
- geste de sélection d'objets,
- geste de suppression de la sélection,

L'utilisateur peut ainsi écrire et modifier des formules simples de manière conviviale à l'aide du stylo. La **figure 3** montre un exemple de formule tracé dans cette interface.

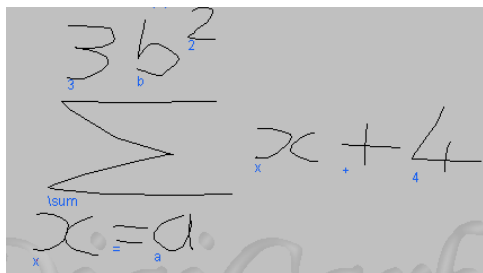


Figure 3 — Capture de l'interface calculette

Pour construire cette interface, nous avons donc décrits tous les symboles et gestes de manière géométrique. Nous avons ensuite construit la description grammaticale des différentes formules possibles (notamment la construction des nombres flottants). Enfin nous avons sommairement décrit les opérations de modification possibles.

Cette interface s'appuie sur la bibliothèque graphique AWT de Java. Elle fonctionne sous la forme d'une application Java, mais est facilement transformable en Applet.

Conclusion

Nous sommes actuellement en train d'effectuer le portage de cette interface sur machine de poche. Pour faire ce portage nous avons décidé de nous appuyer sur la machine virtuelle SuperWaba.

SuperWaba [2] est une machine virtuelle libre et ouverte destinée aux machines de poches. Initialement développée pour le système PalmOS, elle a récemment été adaptée au Pocket-PC.

De par sa vocation à fonctionner dans des environnements réduits, SuperWaba n'est pas entièrement compatible avec Java.

Les caractéristiques les plus gênantes pour notre portage sont les suivantes :

- les processus légers sont implémentés de manière coopérative : il est donc impossible d'effectuer la reconnaissance d'un symbole pendant le tracé d'un autre,
- la réflexion n'est pas encore implémentée : la sérialisation n'est donc pas possible de manière transparente ; nous avons dû développer notre propre mécanisme,
- le modèle Observable/Observer n'est pas implémenté,
- Object.notify() et Object.wait() non plus.

Notre portage est toutefois en bonne voie d'aboutissement. Le principal souci qui nous a été évité est le portage de la phase d'analyse de la description XML.

Il est en effet exclu de faire cette analyse sur la machine de poche car, d'une part, cela pénaliserait l'utilisateur en prenant une place conséquente en mémoire (à la fois la description et l'analyseur XML), et d'autre part cela le pénaliserait en interactivité car l'interface aurait à analyser cette structure.

Notre principe a été dès le départ de séparer l'analyse de la description XML de son utilisation. La première phase d'analyse peut donc se conduire sur une machine puissante, le résultat de cette analyse étant transféré pour être utilisé par l'interface.

Lorsque le portage de cette interface sera terminé, nous pourrons l'utiliser sur d'autres types de documents structurés simples. Pour ce faire, nous n'aurons qu'à analyser la description de ce document et transférer le résultat à l'interface.

Nous pensons notamment développer des interfaces d'édition de formules mathématiques plus complexes, de mini-partitions musicales et de graphes valués.

Bibliographie

- [1] *Système itératif d'interprétation multicontextuelle pour la lecture d'écriture manuscrite*, Laurent Pasquer, Éric Anquetil, Guy Lorette, RFIA'200, p. 347-356.
- [2] *SuperWaba*, Guilherme C. Hazan, www.superwaba.org.