

# Mesure de la taille fonctionnelle des logiciels temps réel

JEAN-MARC DESHARNAIS, ALAIN ABRAN ET DENIS SAINT-PIERRE

**Résumé :** Les points de fonction constituent la mesure de la taille fonctionnelle la plus employée dans le domaine des systèmes d'information. Ses limites quant au logiciel temps réel ou embarqué ayant été souvent dénoncées, cette mesure n'est pas à l'heure actuelle jugée appropriée pour estimer l'effort relatif à un logiciel temps réel. En 1997, une nouvelle extension à la technique des points de fonction (intitulée points de fonction étendus -PFE) a été introduite pour mesurer la taille fonctionnelle de logiciels temps réel et pallier les faiblesses les plus évidentes des points de fonction de l'IFPUG tout en conservant la compatibilité avec les points de fonction classiques pour le logiciel des systèmes d'information. La technique des points de fonction étendus a été retenue comme standard de mesure pour le logiciel temps réel par l'International Software Benchmarking Standards Group. Le présent article présente les concepts de base de cette extension et donne des résultats d'essais d'utilisation dans différents contextes industriels. La capacité de la technique à traduire la taille fonctionnelle du logiciel temps réel est illustrée par des mesures, effectuées à l'aide des méthodes PFA et FPE, sur les mêmes produits logiciels. Un retour d'expérience issu de l'industrie est aussi présenté.

## I. INTRODUCTION

Étant donné son importance dans les produits et les services, le logiciel est devenu un poste majeur des budgets des entreprises. Comme pour tout autre poste budgétaire, il est important d'en contrôler les dépenses, d'analyser l'efficacité des sommes allouées au développement du logiciel et de comparer à ce qui se fait de mieux dans le domaine. À cette fin, des mesures sont nécessaires pour analyser à la fois la qualité et la productivité du développement et de la maintenance. D'une part, des mesures techniques sont requises pour mesurer, du point de vue du développeur, les performances techniques des produits ou des services et, d'autre part, des mesures fonctionnelles sont nécessaires pour mesurer ces mêmes performances du point de vue de l'utilisateur et analyser la productivité. Les mesures fonctionnelles doivent être indépendantes des décisions techniques du développement et de la mise en œuvre. Elles peuvent être utilisées pour comparer les pro-

ductivités permises par différentes techniques et technologies.

Une telle méthode de mesure fonctionnelle, l'analyse des points de fonction (dite FPA en anglais), est disponible pour les systèmes d'information, domaine où elle a été utilisée intensivement en analyse de la productivité et en estimation [2, 6, 7, 20]. Cette méthode permet de capturer convenablement les caractéristiques fonctionnelles du logiciel des systèmes d'information. Toutefois, il lui est reproché de ne pas être applicable universellement à tous les types de logiciel [4, 8-10, 14-16, 18, 25]. Par exemple, elle ne permet pas de traduire toutes les caractéristiques du logiciel temps réel. Lorsqu'elle est appliquée à cette catégorie de logiciel, elle fournit bien des chiffres mais ces derniers ne constituent pas une mesure pertinente de la taille. Pour cette raison, jusqu'à de façon récente il n'y avait pas d'équivalent à FPA pour le logiciel temps réel permettant de comparer des productivité

et de développer des modèles d'estimation reposant sur la taille complète du logiciel temps réel.

Le présent article décrit les travaux effectués par le Laboratoire de Recherche en Gestion du Logiciel de l'Université du Québec à Montréal et par son partenaire industriel SELAM en vue d'adapter la méthode FPA au logiciel temps réel.

**2. LES LIMITES DE LA MÉTHODE FPA VIS-À-VIS DU LOGICIEL TEMPS RÉEL**

**2.1 LES LIMITES VIS-À-VIS DES DONNÉES**

Il existe deux types de structure de données de contrôle : les groupes de données à occurrences multiples et les groupes de données à occurrence simple. Les premiers groupes de données peuvent avoir plusieurs exemplaires d'un même type d'enregistrement, alors que les groupes du second type n'ont qu'un seul exemplaire d'un même enregistrement. Le logiciel temps réel contient typiquement un grand nombre de groupes de données à occurrence unique difficiles à regrouper dans les fichiers internes et externes logiques de l'IFPUG (respectivement. Internal Logical Files et External Logical Files). Une extension est alors nécessaire pour mesurer de façon pertinente les groupes de données à occurrence unique.

**2.2 LES LIMITES VIS-À-VIS DES TRANSACTIONS**

Les processus du logiciel temps réel partagent une même caractéristique transactionnelle : le nombre de leurs sous-processus varie largement. Une technique de mesure fonctionnelle pour le temps réel doit tenir compte du fait que certains processus n'ont que peu de sous-processus, alors que d'autres en ont un grand nombre. Les deux exemples suivants illustrent ce phénomène.

*Exemple 1 : Un processus de contrôle de température de moteur (cas d'un nombre restreint de sous-processus)*

Le but premier de ce processus est d'activer un système de refroidissement lorsque c'est nécessaire. Un capteur fournit la température à l'application (sous-processus 1). La température est comparée au seuil de température de surchauffe (sous-processus 2). Enfin, un message d'activation est envoyé au système de refroidissement si nécessaire (sous-processus 3). Dans cet exemple, le processus de contrôle de température a trois sous-processus (voir tableau I). Ce processus n'est pas une application ; il ne constitue que l'un des nombreux processus d'une application de contrôle de moteur. Selon les règles de l'IFPUG, l'application n'atteint pas un état cohérent tant que tous les sous-processus du processus de contrôle de température ne sont pas terminés. En conséquence, n'est compté qu'un processus élémentaire [12] et alors qu'une seule fonction.

Processus	Description des sous-processus	Nombre de sous-processus
Contrôle de moteur	Prise de la température	1
	Lecture du seuil pour la comparaison	1
	Envoi du message d'activation	1
	Total	3

*Exemple 2 : Un processus de diagnostic de moteur (cas d'un grand nombre de sous-processus)*

**Tableau 1 : Les sous-processus de l'exemple 1**

Le but premier de ce processus est d'activer une alarme lorsque nécessaire. Quinze capteurs sur le moteur (tous différents) envoient des données au processus de diagnostic (quinze sous-processus, un sous-processus unique pour chaque type de capteur). Pour chaque capteur, l'ensemble des données externes reçues est comparé à des valeurs de seuil extraites d'un fichier interne, un fichier unique pour chaque type de capteur (quinze autres sous-processus, un processus unique pour chaque type de capteur). Dans certaines conditions, une alarme sur le tableau de bord peut être déclenchée (un sous-processus). Dans cet exemple, le processus de diagnostic comprend 31 sous-processus (voir Tableau II). Ce processus n'est pas une application ; il n'est seulement que l'un des nombreux processus d'une application de diagnostic de moteur. Selon les règles standard de l'IFPUG, l'application n'atteint pas un état cohérent tant que tous les sous-processus du processus de diagnostic ne sont pas terminés.

Processus	Description des sous-processus	Nombre de sous-processus
Diagnostic de moteur	Entrée des données	15
	Lecture des seuils pour la comparaison	15
	Envoi du message d'alarme	1
	Total	31

**Tableau 2 : Les sous-processus de l'exemple 2**

Selon les règles standard de l'IFPUG, seulement quelques points de transaction seront comptés, car les fonctions transactionnelles reposent sur des processus élémentaires plutôt que sur des processus. En conséquence, avec les procédures de mesure actuelles de l'IFPUG les systèmes des deux exemples ci-dessus auront la même taille fonctionnelle.

**3. MESURE DE LA TAILLE FONCTIONNELLE DU LOGICIEL TEMPS RÉEL**

Les points de fonction étendus constituent une mesure de la taille fonctionnelle du point de vue de l'utilisateur et ne tiennent pas compte des caractéristiques techniques du logiciel. Cette mesure a été conçue à la fois pour le logiciel des systèmes d'information et le logiciel temps réel.

Les deux exemples donnés plus haut montrent que, pour mesurer de façon pertinente les caractéristiques d'un nombre variable de sous-processus, le concept de processus élémentaire défini par l'IFPUG est inapproprié, car la fonctionnalité au niveau du sous-processus devrait aussi prendre en compte la mesure de la taille

► fonctionnelle du point de vue de l'utilisateur. En conséquence, une nouvelle base de mesure a été introduite avec les points de fonction étendus, conjointement avec des types de fonction transactionnelle et de données associées plus proches des caractéristiques des données de contrôle.

Les deux nouveaux types de fonction de données de contrôle introduits ont une structure similaire à celle des types de fonction de données de l'IFPUG :

- *Groupe de contrôle modifié (Updated Control Group ou UCG)* : un UCG est un groupe de données de contrôle modifiées par l'application à mesurer. Il est identifié du point de vue fonctionnel. Les données de contrôle restent vivantes pour plus d'une transaction<sup>3</sup>.
- *Groupe de contrôle à lecture seule (Read-only Control Group ou RCG)* : un RCG est un groupe de données de contrôle utilisées, mais non modifiées, par l'application mesurée. Il est identifié du point de vue fonctionnel. Les données de contrôle restent vivantes pour plus d'une transaction.

Les quatre types de fonction transactionnelle de contrôle nouveaux s'adressent aux sous-processus du logiciel temps réel :

- *Entrée de contrôle externe (External Control Entry ou ECE)* : un ECE est un sous-processus unique. Il est identifié d'un point de vue fonctionnel<sup>4</sup> et traite les données de contrôle venant de l'extérieur des frontières de l'application. Il s'agit du niveau de décomposition le plus bas d'un processus agissant sur un groupe de données. En conséquence, si un processus introduit deux groupes de données, il y a au moins deux ECE. Dans l'exemple du diagnostic de moteur, quinze capteurs envoient des données à l'application (les données de contrôle traversent la frontière de l'application) ; comme il n'y a qu'un seul processus pour chaque capteur, il y a quinze ECE.
- *Sortie de contrôle externe (External Control exit ou ECX)* : un ECX est un sous-processus unique. Il est identifié du point de vue fonctionnel et traite les données allant vers l'extérieur de la frontière de l'application. C'est le niveau de décomposition le plus bas d'un processus agissant sur un groupe de données. En conséquence, si un processus émet deux groupes de données, il y a au moins deux ECX. Dans l'exemple du diagnostic de moteur, le sous-processus qui émet un message vers le tableau de bord (données de contrôle

envoyées à l'extérieur de la frontière de l'application) est un ECX.

- *Lecture de contrôle interne (Internal Control Read ou ICR)* : un ICR est un sous-processus unique. Il est identifié du point de vue fonctionnel et lit des données de contrôle. C'est le niveau de décomposition le plus bas d'un processus agissant sur un groupe de données. En conséquence, si un processus lit deux groupes de données, il y a au moins deux ICR. Dans l'exemple du diagnostic de moteur, le sous-processus qui lit les valeurs des seuils est un ICR. Dans cet exemple, quinze sous-processus uniques lisent différentes sortes de valeurs de seuil à différents moments dans un but de comparaison ; en conséquence, il y a quinze ICR.

- *Écriture de contrôle interne (Internal Control Write ou ICW)* : un ICW est un sous-processus unique. Il est identifié du point de vue fonctionnel et écrit des données de contrôle. C'est le niveau de décomposition le plus bas d'un processus agissant sur un groupe de données. En conséquence, si un processus écrit dans deux groupes de données, il y a au moins deux ICW

La figure 1 schématise les relations entre les fonctions transactionnelles de contrôle et le processus de contrôle.

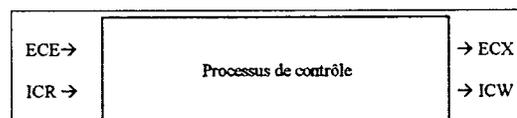


Figure 1 : Les fonctions transactionnelles de contrôle

Comme pour l'IFPUG, tous les nouveaux types de fonctions reposent sur l'aspect fonctionnel de l'application plutôt que sur les aspects techniques de cette dernière.

#### 4. RETOUR D'EXPÉRIENCE ISSU D'ESSAIS SUR LE TERRAIN DANS L'INDUSTRIE

Les chapitres précédents ont donné un aperçu des concepts et définitions des points de fonction étendus ainsi que deux exemples. Un certain nombre d'applications industrielles ont été mesurées à l'aide des points de fonction étendus au cours d'essais sur le terrain portant sur des systèmes logiciels industriels en Amérique du Nord et en Asie. Ce qui suit présente les résultats obtenus.

##### 4.1 ESSAIS INITIAUX

Préalablement à sa publication en 1997, la méthode a été expérimentée. Un premier ensemble d'essais ont été conduits par les auteurs de la méthode eux-mêmes ; un deuxième par un partenaire industriel avec l'aide de l'équipe de recherche. Ces premières expéri-

mentations ont conduit aux observations suivantes :

- *Facilité de compréhension* : Toutes les mesures de points de fonction étendus ont été effectuées avec l'aide d'experts en applications temps réel (ingénieurs de l'industrie familiers des applications mesurées). Une fois qu'ils avaient compris les définitions des nouveaux types, ces experts n'avaient aucun problème pour identifier ces derniers. En fait, après un jour entier consacré à mesurer à l'aide des points de fonction étendus, les experts d'application étaient capables de mesurer avec une aide minimale de la part d'experts en estimation de taille fonctionnelle connaissant les points de fonction étendus.
- *Effort de mesure* : D'après les essais conduits sur le terrain, l'effort de mesure à l'aide des points de fonction étendus est similaire à celui connu avec l'analyse des points de fonction (FPA). Certes, davantage de fonctions doivent être identifiées avec les points de fonction étendus, mais, étant plus simples<sup>5</sup>, les nouvelles fonctions transactionnelles des points de fonction étendus sont identifiées plus rapidement. Par ailleurs, les experts des applications semblent requérir moins d'aide de la part des experts en estimation de taille fonctionnelle, à nouveau en raison du fait que les fonctions transactionnelles des points de fonction étendus sont plus simples à manipuler.
- *Importance de la documentation* : Une source d'informations fonctionnelles doit être disponible pour mesurer une application. Ces informations fonctionnelles sont fournies par les experts des applications ou par la documentation associée aux applications. D'après les essais conduits sur le terrain, les informations fonctionnelles habituellement disponibles dans l'industrie sont adéquates pour mesurer à l'aide des points de fonction étendus.
- *Mesures précoces de points de fonction étendus* : Il est possible de mesurer les types de points de fonction étendus à un stade précoce du développement pourvu que les exigences fonctionnelles soient documentées. Malgré des types de fonctions additionnels, le niveau de détail des points de fonction étendus est similaire à celui de l'IFPUG. Pour utiliser la méthode de l'IFPUG aux premières étapes du développement, on doit habituellement faire une approximation sur les mesures, car toutes les informations fonctionnelles ne sont pas disponibles. Au fil des ans, des méthodes d'approximation pour les mesures aux premières étapes du développement ont été conçues ;

des méthodes d'approximation similaires pourraient être développées pour les points de fonction étendus<sup>6</sup>.

#### 4.2 AUTRES ESSAIS SUR LE TERRAIN

D'autres essais sur le terrain ont été conduits postérieurement à la publication de la méthode des points de fonction étendus et leurs résultats ont été publiés [7, 22, 23]. Ces essais industriels additionnels ont permis de comparer les tailles obtenues avec les points de fonction étendus à celles obtenues avec l'analyse des points de fonction. Toutes les procédures de mesure ont été conduites par un même expert en mesure, expert possédant une expérience de douze ans dans la mesure de taille fonctionnelle, éliminant ainsi les divergences entre des experts en mesure.

Le tableau 3, dû à S. Oigny [23], donne les résultats pour cinq logiciels distincts issus d'une entreprise du secteur des télécommunications. Ces logiciels ont été mesurés en utilisant à la fois les règles IFPUG 4.0 et celles de la méthode des points de fonction étendus. Parmi ces sept logiciels, quatre étaient représentatifs des systèmes temps réel, deux l'étaient des systèmes d'information et l'un relevait principalement des systèmes d'information mais comportait certaines fonctionnalités de temps réel. La distinction entre logiciels de système d'information et logiciels temps réel repose sur la connaissance des experts fonctionnels et des praticiens au sein de l'entreprise.

LOGICIEL	TYPE	TAILLE FPA	TAILLE PFE	DIFFÉRENCE	ÉCART RELATIF
A	Temps réel	210	794	584	74 %
B	Temps réel	115	183	68	37 %
C	Temps réel	0	2604	2604	100 %
D	Temps réel	43	318	275	86 %
E	SI + Temps réel	764	791	27	3 %
F	SI	272	676	404	60 %
G	SI	878	896	18	2 %

De ce tableau on peut remarquer les faits suivants :

La différence de taille est significative dans tous les cas où le logiciel comporte de nombreux processus temps réel (logiciels A, B, C et D). La taille selon les points de fonction étendus est considérablement plus élevée que celle fournie par l'analyse des points de fonction (PFA).

Le logiciel C n'a absolument pas pu être mesuré avec l'analyse des points de fonction (PFA), car sa fonctionnalité n'entraîne pas avec clarté dans les catégories de types de fonction de cette méthode. Il ne relevait pas non plus des définitions de processus élémentaires des règles IFPUG 4.0.

Les résultats concernant la taille sont similaires quand les deux méthodes sont appliquées à un logiciel de type système d'information (logiciels E et G).

## 6. INTÉGRATION AVEC LA MÉTHODE FPA DE L'IFPUG

Pour les entreprises possédant un recueil important de mesures de taille effectuées à l'aide de la méthode de l'IFPUG, il est important de maintenir la cohérence et de préserver les investissements passés. En fait, comme la méthode PFE est une extension de la méthode de l'IFPUG, toutes les règles de l'IFPUG peuvent être incluses dans cette nouvelle technique de mesure. Toutefois, les quelques règles IFPUG concernant les concepts de contrôle doivent être remplacées par les nouveaux types de fonction de la méthode PFE. Ainsi, le décompte pour une application mesurée avec la méthode PFE peut être relié à celui obtenu avec la méthode PFA de la façon suivante :

$$\begin{aligned} \text{PFE} &= \text{PF Gestion} + \text{PF Contrôle} \\ \text{PFE} &= (\text{FPA-Information de contrôle}) + \text{PF Contrôle} \end{aligned}$$

La différence entre la méthode IFPUG et l'extension proposée ici réside alors dans les nouveaux types de fonction. Ces nouveaux types sont utilisés seulement pour les données de contrôle et les processus de contrôle. Les autres types de données et de processus, appelés ici données de gestion et processus, sont mesurés avec la méthode IFPUG (voir figure 2 et tableau 4). Bien entendu, il est important de signaler quelle stratégie a été utilisée lorsque les résultats sont fournis et sont utilisés à des fins de comparaison.

Types de fonctions de gestion de la méthode FPA :

Internal Logical Files (ILF)  
External Interface File (EIF)  
External Input (EI)  
External Output (EO)  
External Inquiry (EQ)

Types de fonctions de contrôle de la méthode PFE (nouveau)

Updated Control Group nouveau type de fonction, similaire à ILF  
Read-only Control group nouveau type de fonction, similaire à EIF  
External Control Group nouveau type de fonction, similaire à un sous-ensemble de EI  
External Control Exit nouveau type de fonction, similaire à un sous-ensemble de EO/EQ  
Internal Control Read nouveau type de fonction, similaire à un sous-ensemble de EI/EO/EQ  
Internal Control Write nouveau type de fonction, similaire à un sous-ensemble de EI

Tableau 4 : liste des types de fonction intégrés (FPA et PFE)

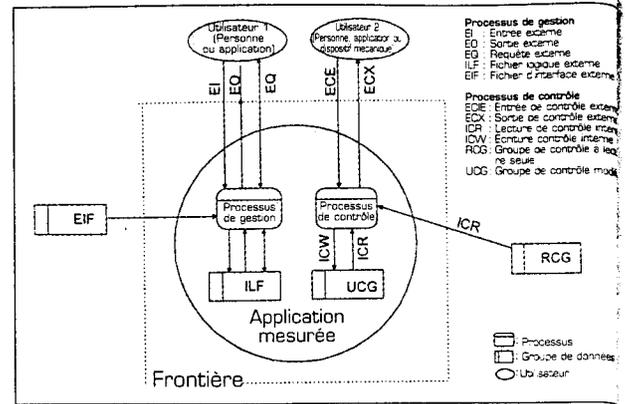


Figure 2 : Schéma des types de fonction FPA + PFE

## 6. CONCLUSION

Considérant les inconvénients de la méthode de mesure classique des points de fonction rencontrés lorsque cette dernière est utilisée dans le cas de logiciels temps réel, la méthode des points de fonction étendus a été proposée comme méthode de mesure de taille fonctionnelle, méthode spécifiquement conçue pour ce type de logiciel.

Les résultats de premiers essais sur le terrain ont montré : 1) l'adéquation de la méthode des points de fonction étendus en termes de couverture fonctionnelle vis-à-vis du logiciel à mesurer ; 2) la nature de la différence observée dans les tailles fonctionnelles mesurées respectivement avec les deux méthodes lorsqu'elles sont appliquées simultanément à un même logiciel temps réel ou embarqué ; 3) la reconnaissance, par les praticiens, d'un bon niveau d'applicabilité dans les contextes industriels où ces types de logiciels sont développés et maintenus. De plus, cette nouvelle méthode a été reconnue par le Comité IBSG (International Benchmarking Standards Group), un organisme international dédié au benchmarking.

Un rapport public, en français, anglais et japonais, détaillant les concepts, les règles et les procédures de mesure est accessible sur les sites suivants :

[www.lrgl.uqam.ca/ffp.html](http://www.lrgl.uqam.ca/ffp.html)  
[www.lmagl.qc.ca/rtreport.pdf](http://www.lmagl.qc.ca/rtreport.pdf)

## 7 NOTES

- (1) Par exemple, une application de contrôle de moteur pourrait avoir un groupe de données de contrôle contenant des informations sur chaque cylindre (numéro du cylindre, instant de l'allumage, pression, etc.). Un tel groupe de données a une structure d'occurrences multiples (un enregistrement par cylindre). En d'autres termes, l'enregistrement « cylindre » est répété plus d'une fois.
- (2) Ceci signifie que le groupe de données figure dans les exigences relatives à l'application, en supposant que ces dernières soient complètes.

- (3) Dans l'exemple du chapitre 2, les données du capteur entrées ne sont vivantes que pour une transaction à la fois, car après le processus de diagnostic, le système ne les mémorise plus. Par contre, les données relatives aux seuils sont réutilisées pour chaque nouvelle entrée et, en conséquence, restent vivantes pour plus d'une transaction.
- (4) Ceci signifie que le sous-processus apparaît, en supposant qu'elles soient complètes, dans les exigences relatives à l'application.
- (5) Les types de fonction FFP sont plus simples, car ils sont relatifs à un seul sous-processus. Les types de fonction de l'IFPUG peuvent comprendre de nombreux sous-processus et les regrouper en processus élémentaire peut être consommateur de temps.
- (6) Bien entendu, on ne doit pas s'attendre, à ce stade, à ce que les méthodes d'approximation des points de fonction étendus soient aussi matures que celles de l'analyse des points de fonction (FPA).

## 8. RÉFÉRENCES

- [1] A. Abran : *Analysis of the measurement process of Function Point Analysis* ; Thèse de Doctorat, Département de Génie Électrique et de Génie Informatique, École Polytechnique de Montréal, 1994, 405 pages
- [2] A. Abran et P. N. Robillard : *Function Point Analysis: An empirical study of its measurement processes* ; IEEE Transactions on Software Engineering, vol. 22, n° 12, pp. 895-909, décembre 1996
- [3] A. J. Albrecht : *AD/M productivity measurement and estimate validation* ; IBM Corporate Information Systems, Ibm Corp., Purchase, NY, mai 1984
- [4] S. D. Conte, V. Y. Shen et H. E. Dunsmore : *Software engineering metrics and models* ; Benjamin Cummins Publishing, 1986, 396 pages.
- [5] J. E. Cooling : *Software design for real-time systems* ; Chapman and Hall, 1991
- [6] J.-M. Desharnais : *Statistical analysis on the productivity of data processing with development projects using the function point technique* ; Université du Québec à Montréal, 1998
- [7] J.-M. Desharnais et P. Morris : *Comparison between FPA and FFP: a field experience* ; 8th international Workshop on Software Measurement, Magdebourg, 17-18 septembre 1998
- [8] S. Galea : *The Boeing Company: 3D Function Point Extensions. V. 2.0, Release 1.0* ; Boeing Information and Support Services, Research and Technology Software Engineering, juin 1995.
- [9] R. B. Grady : *Practical software metrics for project management and process improvement* ; Prentice Hall, New Jersey, 1992, 270 pages.
- [10] B. Hetzel : *Making software measurement work* ; QEB Publishing Group, 1993, 290 pages.
- [11] IEEE : *IEEE Standard Computer Dictionary: A compilation of IEEE standard computer glossaries* ; IEEE Std 610-1990, IEEE, 1990
- [12] IFPUG : *Function point counting practices manual, release 4.0* ; IFPUG Standards, 1994
- [13] V. Illingworth (réd.) : *Dictionary of computing (3<sup>e</sup> édition)* ; Oxford University Press, 1991, 510 pages
- [14] D. C. Ince : *History and industrial applications* ; in N. E. Fenton, *Software metrics: A rigorous approach*, Chapman & Hall, 1991, 337 pages.
- [15] C. Jones : *A short history of function points and feature points* ; Software Productivity Research Inc., Cambridge, Mass., 1988.
- [16] C. Jones : *Applied software measurement - Assuring productivity and quality* ; McGraw-Hill, 1991, 493 pages.
- [17] C. Jones : *Applied software measurement - Assuring productivity and quality* ; McGraw-Hill, 1996, 618 pages.
- [18] S. H. Kan : *Metrics and models in software quality engineering* ; Addison-Wesley, 1993, 344 pages.
- [19] C. F. Kemerer : *Reliability of function point measurement: a field experiment* ; Communications of the ACM, vol. 36, pp. 85-97, 1993
- [20] B. Kitchenham : *Making process predictions* ; in N. E. Fenton, *Software metrics: A rigorous approach*, Chapman and Hall, 1991, 337 pages
- [21] P. Laplante : *Real-time systems design and analysis: An engineer's handbook* ; The Institute of Electrical and Electronics Engineers, 1993, 339 pages
- [22] P. Morris et J.-M. Desharnais : *New methods for measuring outsourcing contracts* ; 10th Year Software Measurement Conference, UK Measurement Association (UKSMA), Londres, 28-30 octobre 1998
- [23] S. Oligny, A. Abran, J.-M. Desharnais et P. Morris : *Functional size of real-time software: overview of field tests* ; Proceedings of the 13th International Forum on COCOMO and software cost modeling, Los Angeles, Californie, 6-8 octobre 1998.
- [24] J. A. Stankovic et K. Ramamritham : *Tutorial hard real-time systems* ; IEEE Computer Society Press, 1988, 618 pages
- [25] S. A. Whitmire : *3-D function points: Scientific and real-time extensions to function points* ; Proceedings of the 1992 Pacific Northwest Software Quality Conference.