

ASSEMI, Paris 1999

**'COSMIC' - COMmon Software Measurement International  
Consortium**

# **COSMIC FFP - Buts, approche conceptuelle et progrès**

**Charles Symons  
(au nom du groupe COSMIC\*)  
30 septembre 1999**

(\* Alain Abran, Charles Symons, Jean-Marc Desharnais, Peter Fagg, Paul Goodman, Pam Morris, Serge Oligny,  
Jolijn Onvlee, Risto Nevalainen, Grant Rule, Denis St Pierre)

## **Buts de cette présentation**

- **Introduire le projet COSMIC, ses buts, sa structure et son statut**
- **Décrire les principaux principes de la mesure de la taille fonctionnelle des logiciels, pour lesquels il y a consensus.**
- **Inviter les organisations à participer à l'étape d'expérimentation des principes COSMIC**

# Importance de la mesure de la taille des logiciels pour la gestion et les ententes contractuelles

## Pourquoi mesurer?

- **Pour normaliser les mesures de performance.**  
**Exemples:**
  - productivité = taille/effort
  - densité des défauts = nbre de défauts/taille
- **Pour l'estimation – la taille est probablement le facteur de coût le plus déterminant.**
- **Comment mesurer?**
  - Lignes de code (valeur limitée)
  - Mesures fonctionnelles. Par exemple, la version 4.1 des points de fonction, Mark II, FFP

# **L'industrie a besoin de meilleures méthodes de mesure de taille des logiciels**

**Les méthodes courantes pour mesurer les demandes (ex.: points de fonction) sont largement acceptées dans le domaine de l'informatique de gestion, mais ne sont pas satisfaisantes à long terme.**

- Elles sont peu acceptées en dehors du domaine de l'informatique de gestion**
- Ne composent pas de façon satisfaisante avec les demandes “non fonctionnelles” (technique et de qualité)**
- Les règles ne sont pas toujours compatibles avec les demandes et les méthodes de développement actuels, tout en étant parfois subjectives dans leur interprétation.**

# Buts du projet COSMIC

**Le projet COSMIC vise à développer, tester, apporter sur le marché et faire accepter comme norme industrielle une nouvelle génération de méthodes pratiques de mesure de la taille des logiciels**

- **pour la mesure de performance**
- **en tant qu'élément des méthodes d'estimation en amont**
- **pour plusieurs domaines, la priorité étant accordée à l'informatique de gestion et au temps réel (ex.: processus de contrôle, téléphonie, embarqué)**

# Méthode COSMIC : caractéristiques désirables

La méthode COSMIC doit:

- dériver la taille de chaque élément d'un logiciel tout au long de son cycle de vie.
- être académiquement juste et compatible avec les vues modernes de détermination des demandes, mais indépendante de toute méthode spécifique
- être compatible avec la norme 14143 d'ISO
- utiliser les meilleures idées de IFPUG 4.1, NESMA, MarkII et la méthode FFP 1.0, plus de nouvelles idées
- produire une taille avec un niveau de confiance mesurable
- ne pas être subjective, être répétable (donc automatisable), mais facile à appliquer manuellement.

# Statut et structure du projet COSMIC

- **Équipe provenant de 5 nations**
- **Début le premier novembre 1998**
- **Initiatives privées, largement auto-financée**
- **Propose des principes de base maintenant largement reconnus**
- **Production par l'UQAM (LRGL à Montréal) de la version 2 de FFP en adoptant les principes de COSMIC; le guide de mesure doit être produit en octobre 1999.**
- **Collecte de données auprès d'un grand nombre d'organisations pour tester les principes de base prévue au quatrième semestre de 1999.**

# Groupe COSMIC

Un groupe avec une vaste expérience tant académique que pratique

		<u>Experience</u>	
Alain Abran	Canada	IFPUG, FFP, ISO WG12	] Chefs de Projet
Charles Symons	UK	MkII FP, ISO WG12	
Jean-Marc Desharnais, Serge Oigny, Denis St Pierre	Canada	IFPUG, FFP	
Peter Fagg, Paul Goodman, Grant Rule	UK	IFPUG, MkII FP, ISO WG12	
Pam Morris	Australia	IFPUG, FFP, ISO WG12	
Jolijn Onvlee	Netherlands	IFPUG, NESMA	
Risto Nevalainen	Finland	IFPUG, Laturi, ISO WG12	
(en croissance)			

## **Buts de cette présentation**

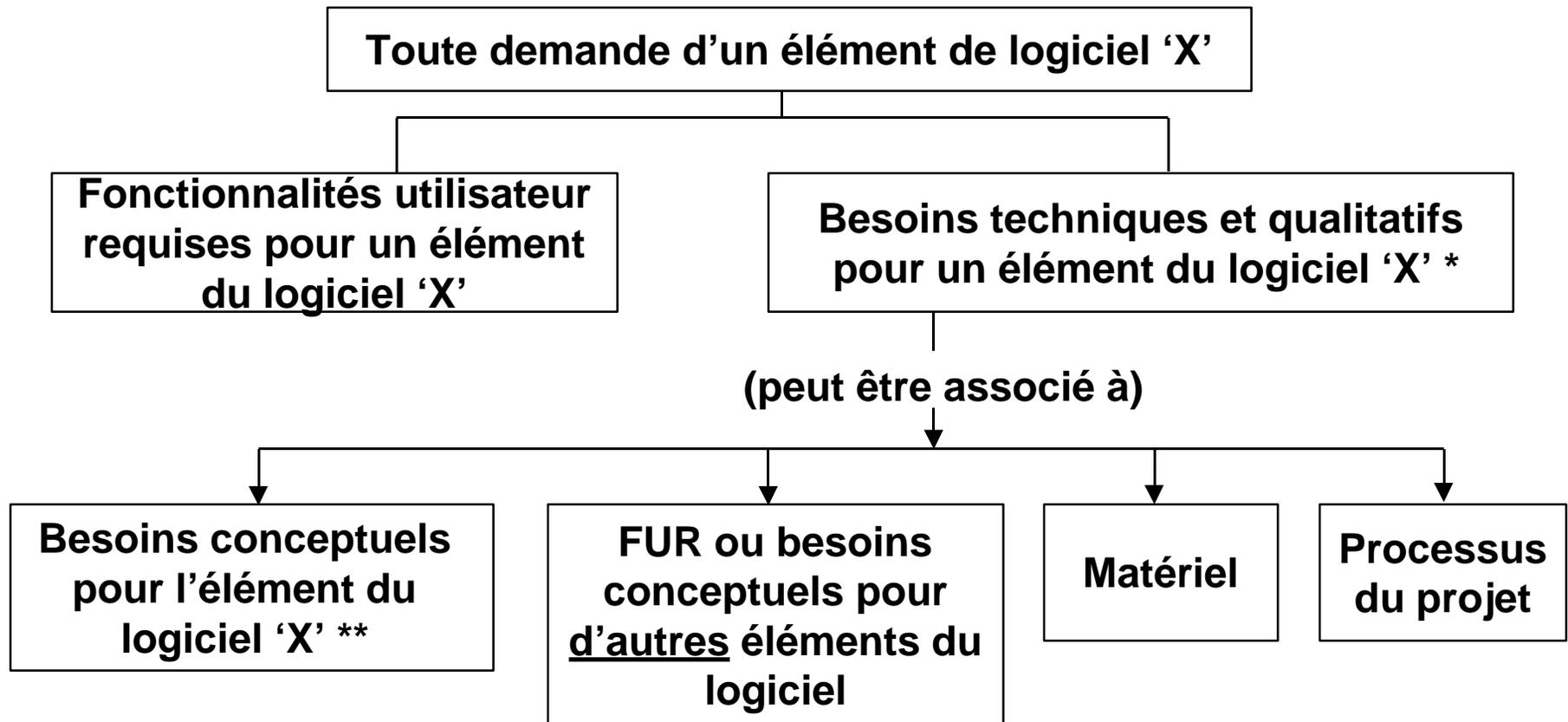
- **Introduire le projet COSMIC, ses buts, sa structure et son statut**
- **Décrire les principaux principes de la mesure de la taille fonctionnelle des logiciels, pour lesquels il y a consensus.**
- **Inviter les organisations à participer à l'étape d'expérimentation des principes COSMIC**

# Principes de base de COSMIC

## Thèmes:

- **Le modèle général structurant toutes les demandes et les Fonctionnalités utilisateur requises ( ' FUR ' ) pour un 'Élément du logiciel'**
- **Les couches et la frontière du logiciel devant être mesuré**
- **Les tailles possibles du logiciel sur la base des demandes**
- **La détermination de la taille d'un Composant de Base Fonctionnel (CBF)**
- **Le méta modèle résumant les concepts de COSMIC FFP**

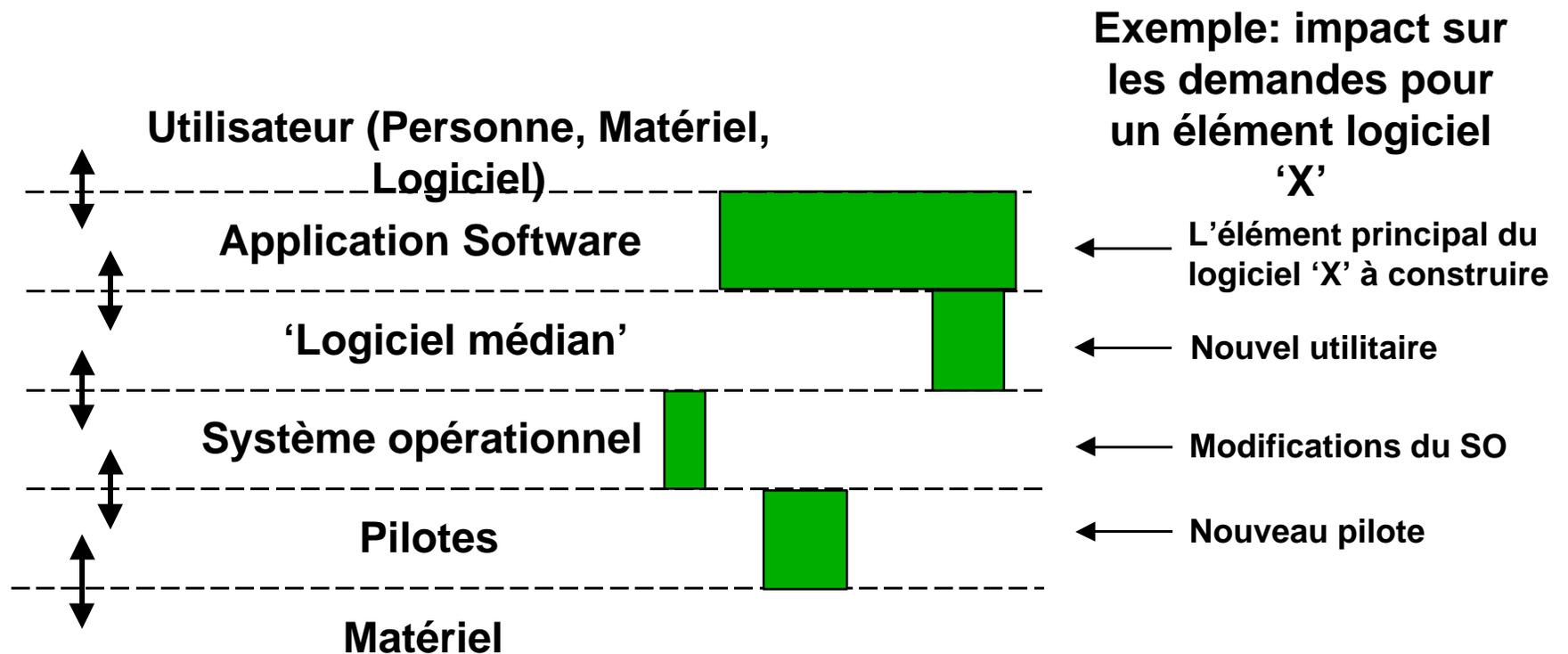
# Les demandes pour un logiciel peuvent être satisfaites de plusieurs façons



•Ex.: demande technique tel un processus en différé ou compatible UNIX, ou une demande de qualité tel la « criticalité » de l'élément du logiciel

•\*\* Ex.: demande de haute performance

# La demande peut affecter plusieurs couches



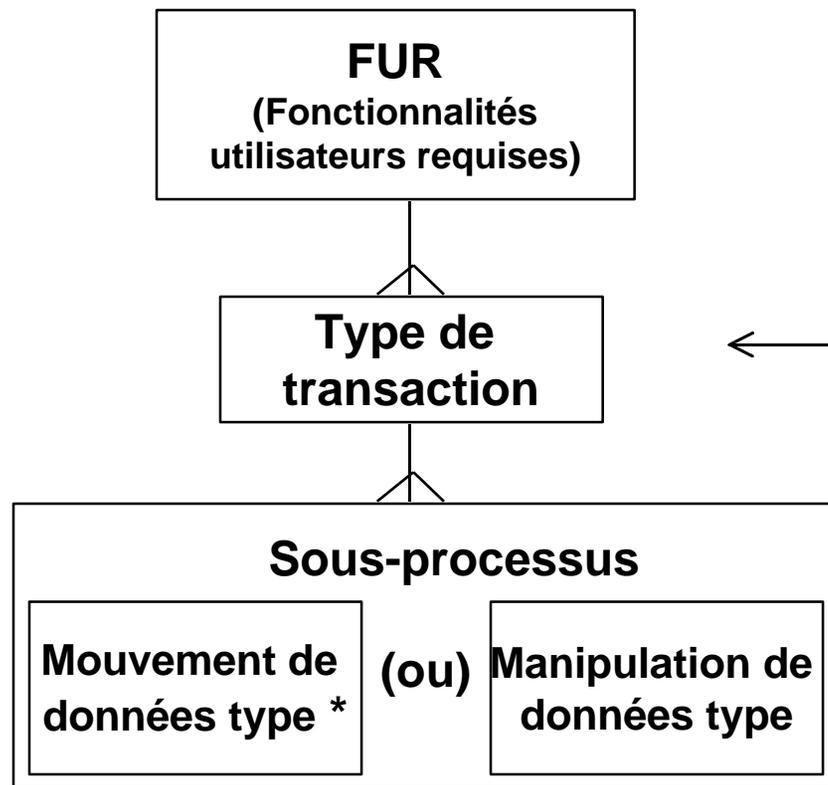
**COSMIC vise à être capable de mesurer la taille des demandes pour le logiciel pour toutes les couches et donne des orientations pour reconnaître les couches.**

## **Reconnaître la taille des pièces de logiciel dans les autres couches doit permettre à COSMIC de simplifier l'approche de la mesure de taille**

- **Les méthodes existantes évaluent les demandes techniques et qualitatives via des facteurs d'ajustements (VAF) ou l'équivalent. On sait que ces facteurs sont peu satisfaisants.**
- **Si COSMIC-FFP peut mesurer la taille des composants, ou les changements des composants, pour toutes les couches, en donnant comme résultat le total des demandes, alors les facteurs tels les VAF sont moins importants.**

**Cependant notre priorité est de mesurer les fonctionnalités utilisateur requises (FUR)**

# Nous proposons un modèle général pour la structure des **FUR** pour tout élément d'un logiciel pour n'importe laquelle des couches

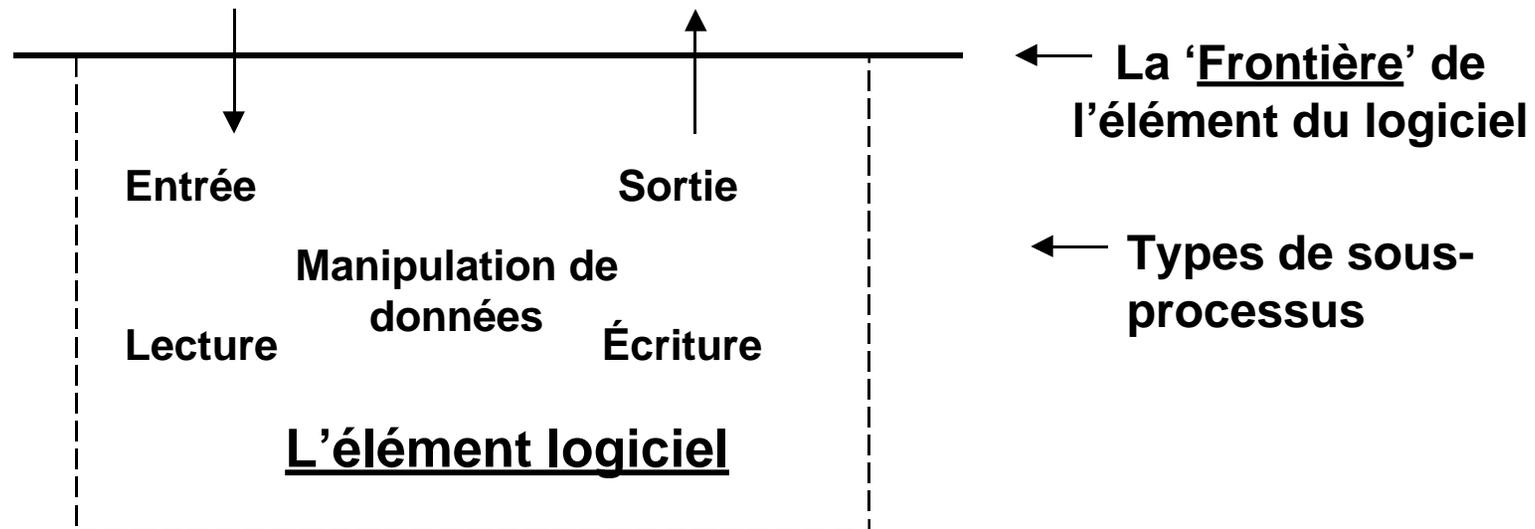


« Processus fonctionnel » selon la terminologie temps réel

\* Peut être brisé en types d'attributs

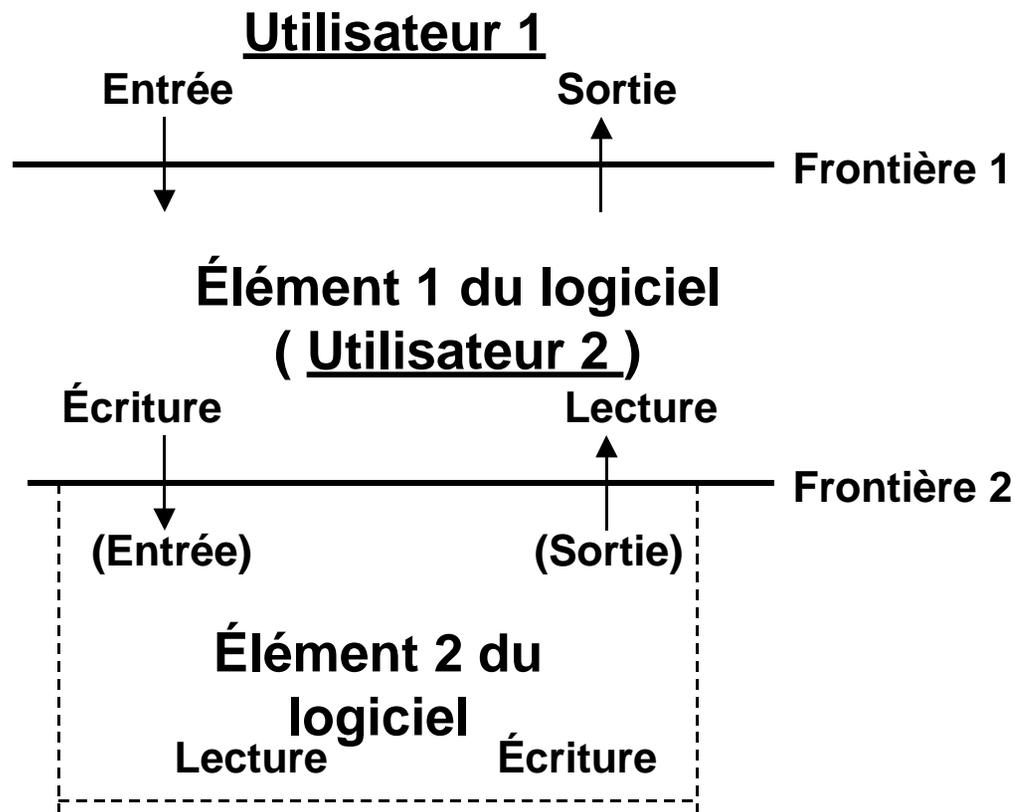
# Le modèle général montre aussi les interactions entre les éléments du logiciel avec la frontière de sa couche

L'utilisateur (Personne ou tout objet incluant un autre logiciel)



Les entrées et sorties déplacent les données à travers la frontière utilisateur/logiciel  
Les lectures et écritures conservent et retrouvent les données persistantes dans l'élément du logiciel, en autant que l'utilisateur est concerné.

# De multiples frontières, et aussi des “couches” sont créées lorsque des tâches requises sont déléguées à des éléments d’autres logiciels



L'architecte décide de déléguer l'écriture et la lecture de l'élément 1 du logiciel à un élément 2 d'une couche inférieure d'un autre logiciel

La frontière 2 est alors créée

L'élément 1 du logiciel devient l'utilisateur de l'élément 2 du logiciel.

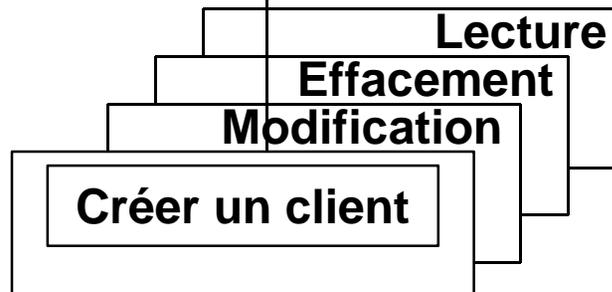
L'écriture telle que perçue par l'utilisateur 1 devient une entrée pour l'utilisateur 2 et l'élément 2 du logiciel

Ainsi le même modèle de sous-processus peut être utilisé à chaque niveau.

# Un exemple simple de la structure générale

**Maintenir les informations d'un client**

Les FUR (Fonctionnalités Utilisateur Requises)

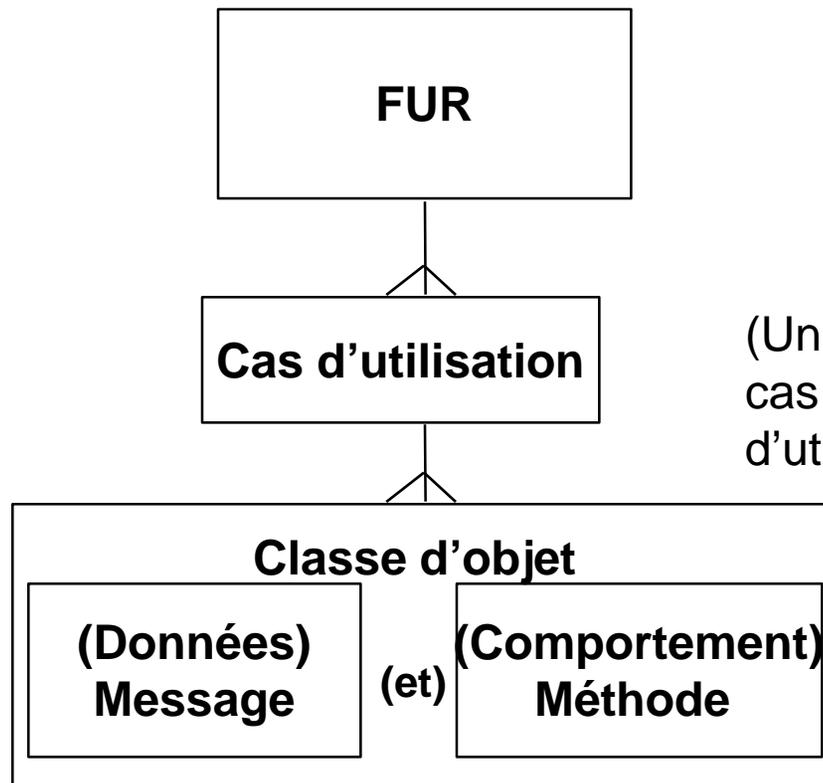


4 types de transactions, chacune déclenchée par un événement unique provenant du monde réel

<b>Entrée:</b>	<b>Entrer les informations d'un client</b>
<b>Manipulation:</b>	<b>Valider les informations du client</b>
<b>Écriture:</b>	<b>Créer un nouveau client</b>
<b>Sortie:</b>	<b>Confirmer la transaction</b>

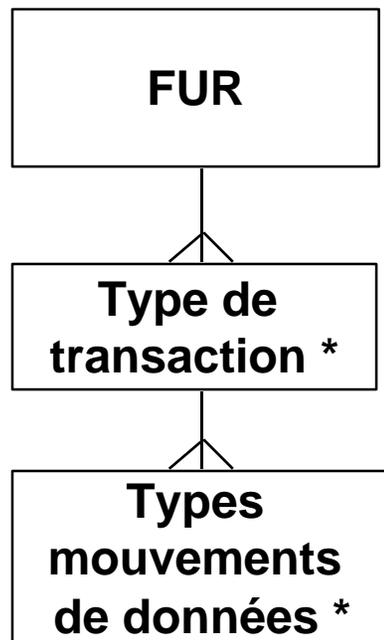
Les sous-processus de 'Créer un client'

# (Observation: ce modèle traduit bien le paradigme Orienté-Objet)



(Un "type de transaction" est un cas spécifique de 'Cas d'utilisation)

## En pratique, cependant, nous avons à simplifier le modèle général, du moins au départ



Les types de mouvement de données (Entrée, Sortie, Lecture ou Ecriture) sont définis comme:

“un sous processus déplaçant un bloc logique de données regroupées

ET la manipulation de données associée aux données du bloc logique”

\* Pour COSMIC “Composants de Base Fonctionnel types” ou ‘Types CBF’

**(Observation: les types CBF COSMIC sont généralement en correspondance avec les CBF de la version 4.1 des points de fonction)**

<b>COSMIC</b>	<b>IFPUG</b>	<b>MkII FPA</b>	<b>FFP V1.0</b>
<b>Type de transaction</b>	<b>Composant (ENT, SOR, INT)</b>	<b>Transaction logique</b>	<b>Composant de contrôle</b>
<b>Types de mouvement de données</b>			
<b>Entrée/Sortie</b>	<b>E/S parties de ENT, SOR et INT</b>	<b>Input/Output</b>	<b>ECE/SCE</b>
<b>Lecture/Écriture</b>	<b>Références à GDI, GDE</b>	<b>Entité Référence</b>	<b>LCI/ECI</b>

**COSMIC FFP a des règles différentes pour établir la taille des CBF**

# Un ensemble de demandes pour un logiciel a plusieurs tailles possibles

## COSMIC en a exploré quatre

Type de taille	Signification de l'unité de taille pour un type de CBF	Utilisations possibles
<b>Taille fonctionnelle</b>	Vue fonctionnelle de l'expert	('Fonctionnel') Mesure de productivité Contrôle des demandes fonctionnelles
<b>Taille totale des demandes</b>	Vue de l'expert de toutes les demandes	('Total') Mesure de productivité Contrôle des demandes totales
<b>Effort normalisé (1)</b>	'Effort normalisé', lorsque la norme de la mesure est une moyenne qui ne tient pas compte des différentes technologies, de l'environnement de développement, etc. et est, par ce fait, indépendante de toute technologie spécifique, etc.	Mesure relative de la productivité Contrôle de la taille du projet
<b>Effort normalisé (2)</b>	'Effort normalisé' dont la mesure est liée à une technologie et un environnement de développement spécifique	Estimation pour une technologie et un environnement de développement spécifique. Contrôle de la taille du projet.

# La convention proposée est d'allouer une unité de taille fonctionnelle pour chaque TMD\*

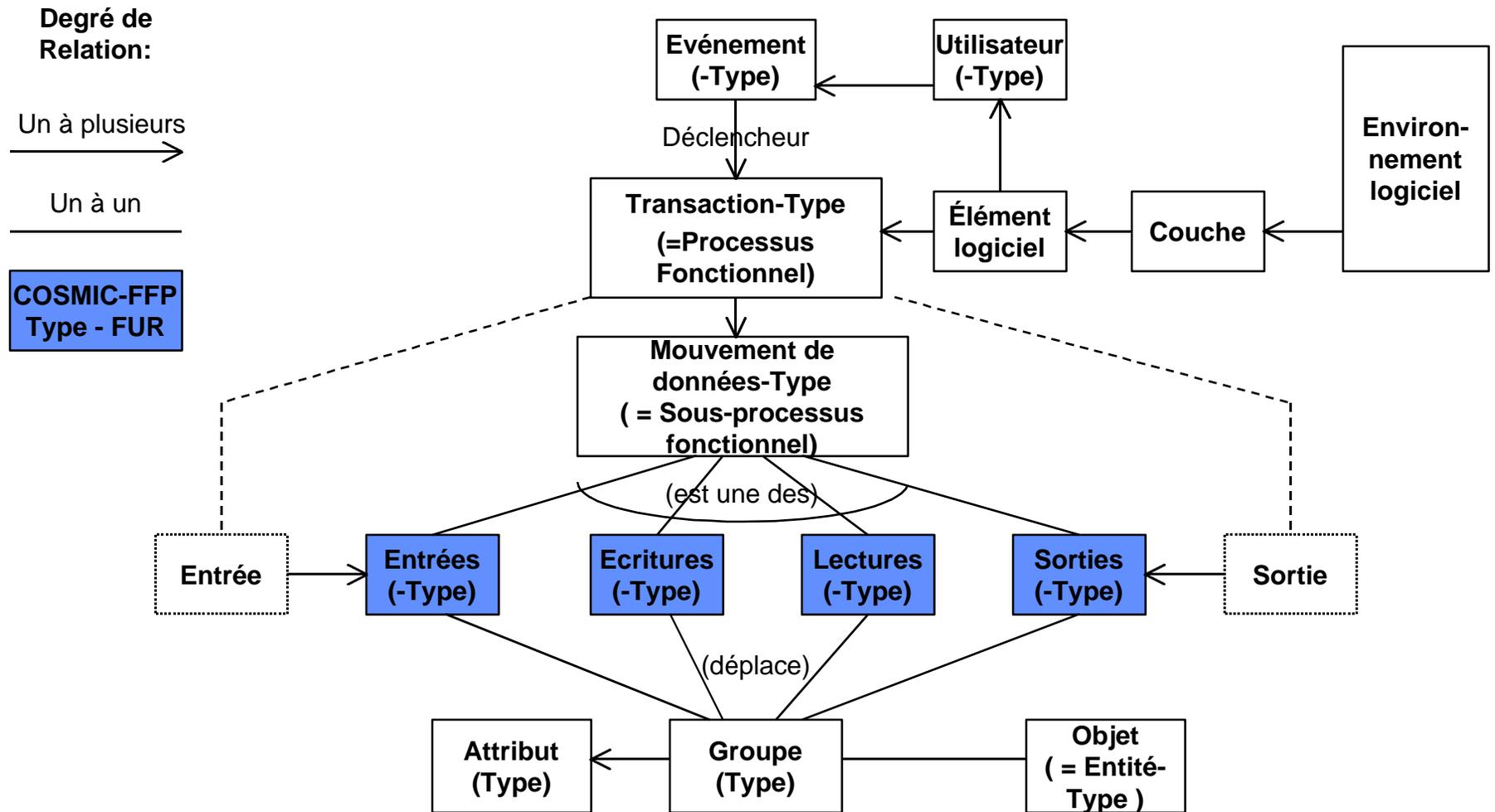
Cependant, la collecte sur le terrain permettra la collecte du nombre d'attributs par TMD afin de vérifier si la granularité proposée est assez fine

## Compter les TMD ou les attributs? Facteurs à considérer:

- Conversion aux méthodes des points de fonction actuelles
- La précision et la granularité requise peut différer par domaine, et la phase du cycle de vie du logiciel
- L'effort pour mesurer la taille fonctionnelle manuellement

TMD: Type de mouvement de données

# Sommaire: Méta-modèle COSMIC FFP



# Principes conceptuels de COSMIC: sommaire

- **Les principes conceptuels de base de COSMIC ont été travaillés et approuvés par un groupe de travail**
- **La méthode COSMIC réalisera un certain nombre de premières. Ce sera la première méthode de taille fonctionnelle qui:**
  - **Aura été conçu par un groupe international d'experts sur des bases théoriques fiables**
  - **Aura capitalisé sur l'expérience de toutes les méthodes de mesures fonctionnelles existantes**
  - **Sera conçu en conformité avec ISO 14143 Partie 1**
  - **Sera conçue pour fonctionner tant pour le domaine de l'informatique de gestion que pour le temps réel, et ce pour toutes les couches du logiciel.**

## **Buts de cette présentation**

- **Introduire le projet COSMIC, ses buts, sa structure et son statut**
- **Décrire les principaux principes de la mesure de la taille fonctionnelle des logiciels, pour lesquels il y a consensus.**
- **Inviter les organisations à participer à l'étape d'expérimentation des principes COSMIC**

## **La prochaine phase implique des essais sur le terrains des principes conceptuels et aussi des règles de calculs**

**Le groupe COSMIC demande aux producteurs de logiciel ou aux utilisateurs tant du domaine de l'informatique de gestion que du temps réel:**

- **S'ils peuvent désigner un expert en mesure du logiciel pour réviser l'applicabilité de l'approche COSMIC dans leur organisation**
- **S'ils peuvent fournir des exemples de demandes pour un logiciel et des données d'effort de développement qui aideraient à réaliser des essais et calibrer la méthode COSMIC avec le support d'un membre du groupe COSMIC**
- **S'ils peuvent contribuer financièrement aux coûts du projet**

# **Chaque participant aux essais sur le terrain peut obtenir des bénéfices significatifs**

- **Primauté de l'accès aux nouvelles idées et une opportunité de les influencer**
- **Mesurer ses projets pilotes avec la nouvelle méthode:**
  - **Données de mesure**
  - **Rapport corporatif confidentiel**
  - **Assistance pendant les projets pilotes et résultats des analyses**
- **Transfert technologique en primeur tout au long du projet pilote.**

# Points de contact pour la participation au projet COSMIC

Voici les principales adresses:

Web-site: [www.cosmicon.com](http://www.cosmicon.com)

Alain Abran: [abran.alain@uqam.ca](mailto:abran.alain@uqam.ca)

Charles Symons: [charles.symons@compuserve.com](mailto:charles.symons@compuserve.com)

Jean-Marc Desharnais: [desharnais.jean-marc@uqam.ca](mailto:desharnais.jean-marc@uqam.ca)

# Principales abréviations

Anglais	Français	Description
<b>COSMIC</b>	<b>COSMIC</b>	<b>COmmon Software Measurement International Consortium</b>
<b>FFP</b>	<b>PFE</b>	<b>Points de fonction étendus</b>
<b>BFC</b>	<b>CBF</b>	<b>Composant de Base Fonctionnel</b>
<b>FUR</b>	<b>FUR</b>	<b>Fonctionnalités utilisateurs requises</b>
<b>Entry</b>	<b>Entrée</b>	
<b>Exit</b>	<b>Sortie</b>	
<b>Read</b>	<b>Lecture</b>	
<b>Write</b>	<b>Ecriture</b>	
	<b>ECE</b>	<b>Equivalent de l'entrée</b>
	<b>SCE</b>	<b>Equivalent de la sortie</b>
	<b>LCI</b>	<b>Equivalent de la lecture</b>
	<b>ECI</b>	<b>Equivalent de l'écriture</b>
<b>DMT</b>	<b>TMD</b>	<b>Type de mouvement de données</b>