

**Reconception de processus Cast3m basée sur des
composants « objet » en GIBIANE**

Club Cast3m 2017

TCHIENKOUA Rigobert

RTSolutions

Campus Tératec, 2 rue de la piquetterie
91680 Bruyères-le-Châtel

Reconception de processus Cast3m basée sur des composants « objet » en Gibiane

Plan de la présentation : ***

- ✦ Un modèle « objet » en Gibiane
- ✦ Gibi+ : Couche « objet » applicative
- ✦ Générateur automatique de code

Un modèle « objet » en Gibiane

- **Au commencement ...**

« *Gibiane* » par T. CHARRAS,
CEA, Série « Utiliser Cast3m » Edition 2011

L'opérateur DEBMETH crée un objet de type PROCEDUR qui contient une suite d'instructions élémentaires, dont la première est DEBMETH et la dernière est FINMETH.

...

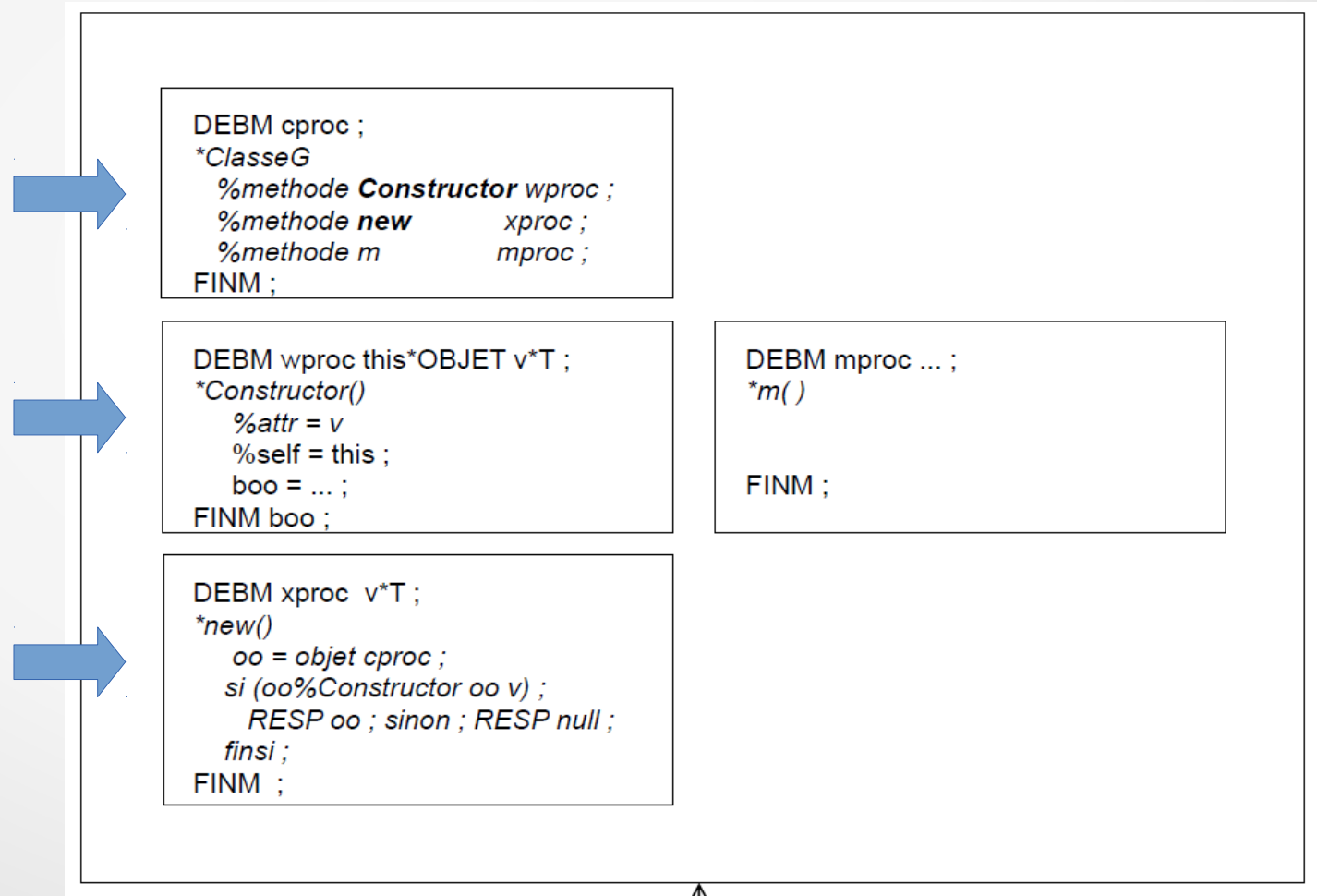
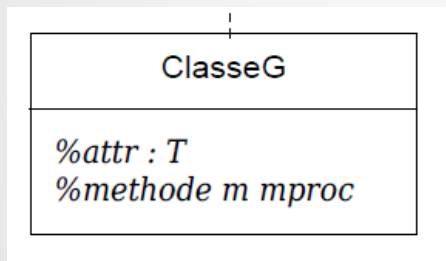
*Une procédure définie ainsi ne peut servir que comme une méthode sur un objet de type OBJET.
(Notice DEBM, Cast3m)*

```
*
* creation of object of class "complex number".
*
* define first the constructor
DEBMETH COMPLEX;
%REA=FAUX;
%IMA=FAUX;
%METHODE SET_IMA  IMAG;
%METHODE SET_REA  REAL;
%METHODE GET_IMA  GIMAG;
%METHODE GET_REA  GREAL;
FINMETH;

* define standard methods of the class
DEBMETH IMAG  I*'FLOTTANT';
%IMA = I;
'SI' ('EGA' ('TYPE' %REA ) 'FLOTTANT');
    %METHODE MODULE  MODUL;
'FINSI';
%OUBLIER SET_IMA;
FINMETH;
```

Un modèle « objet » en Gibiane

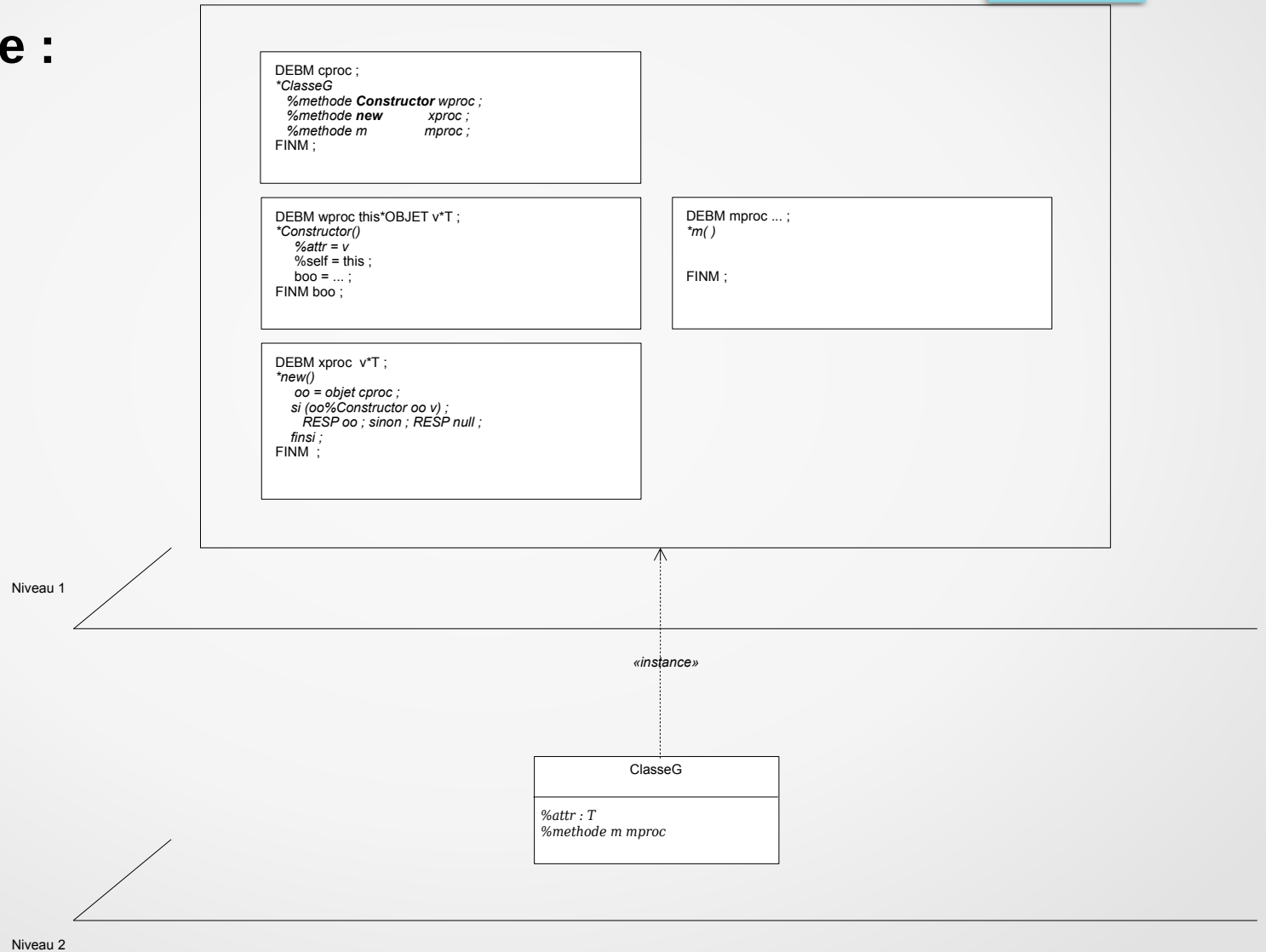
- ... modèle « objet » = 3 méthodes de base



Un modèle « objet » en Gibiane

- Méta-modèle :

Classe



Un modèle « objet » en Gibiane

- **Méta-modèle :**

Exemple simple : Classe Box

Classe Box

```
1 DEBM mei29oo ;
2 *Classe Box
3     %methode Constructor mei29om ;
4     %methode new         mei29op ;
5 FINM ;
```

Constructeur

```
1 DEBM mei29om this*OBJET w*FLOTTANT h*FLOTTANT d*FLOTTANT ;
2 *Constructor()
3 * - Variables d'instance
4     %width = w ;
5     %height = h ;
6     %depth = d ;
7 * - Référence de l'objet
8     %self = this ;
9 * - Indicateur de conformité (VRAI)
10     boo = VRAI ;
11 FINM boo ;
```

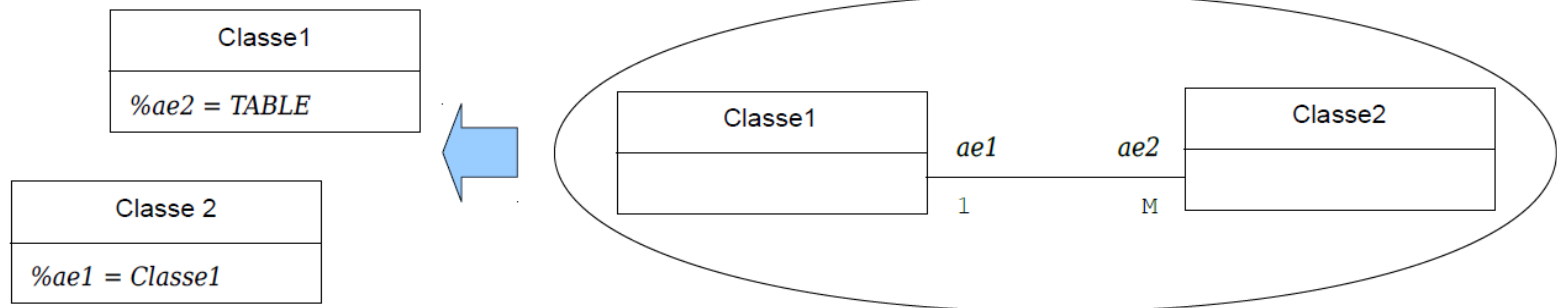
Opérateur new()

```
1 DEBM mei29op w*FLOTTANT h*FLOTTANT d*FLOTTANT ;
2 *new()
3     oo = objet mei29oo ;
4     si (oo%Constructor oo w h d) ;
5         RESP oo ; sinon ; RESP (getNULL) ;
6     finsi ;
7 FINM ;
```

Un modèle « objet » en Gibiane

- Méta-modèle :

Association de classes



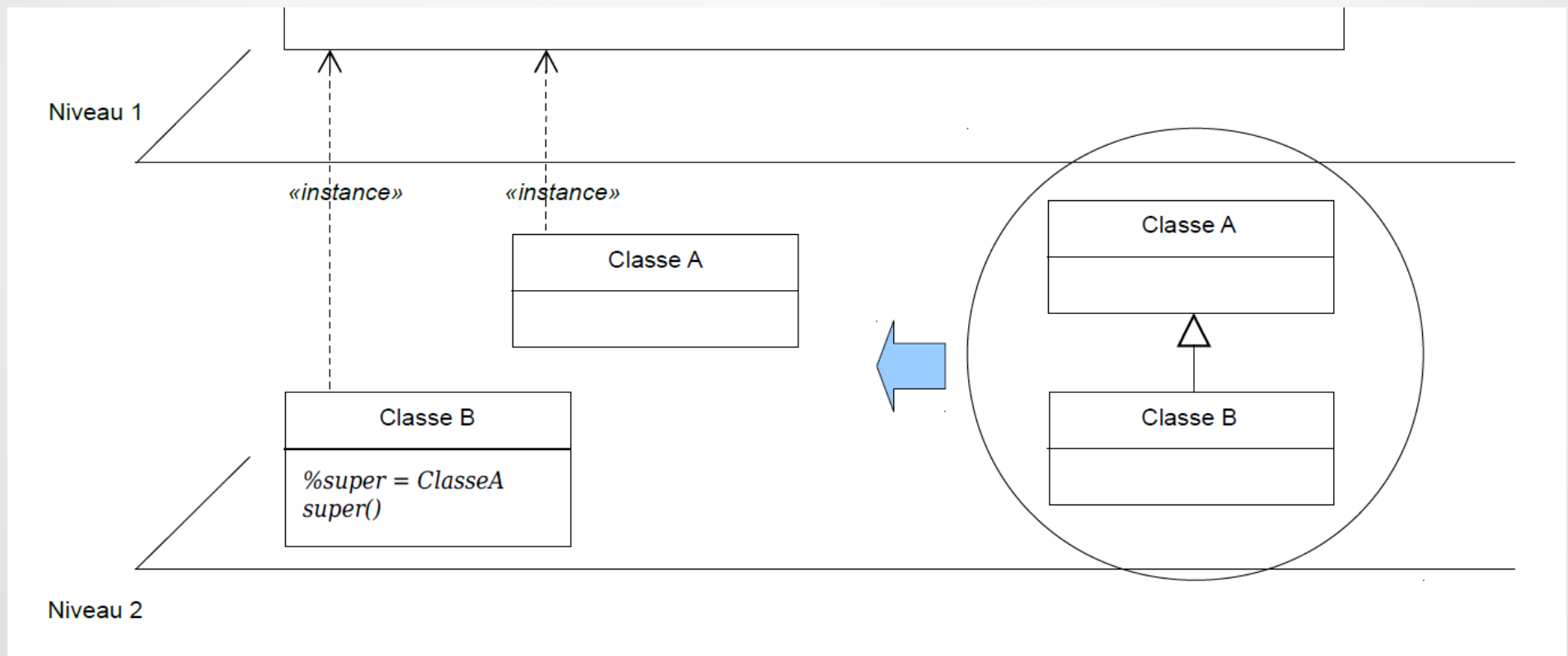
Niveau 2

```
M = DIME %.ae2 ;
(%.ae2).i = o2 ; (0 < i < M+1)
avec o2 = instanceof Classe2 ;
```

Un modèle « objet » en Gibiane

- Méta-modèle :

Héritage



Un modèle « objet » en Gibiane

- **Persistence des classes :**

- ... **nommage**

```
1  DEBM mei29oo ;
2  *Classe Box
3  %className = MOT 'Box' ;
4  *
5  %methode Constructor mei29om ;
6  %methode new          mei29op ;
7  FINM ;
```

- ... **unicité**

```
1  DEBP setClass oc*OBJET bd*TABLE ;
2  *Stocker un objet après avoir vérifié que son nom est unique. Ce nom est
3  *extrait de la classe elle-même.
4  *Règle d'écriture:
5  *   setClass oc tab ;
6  *| oc | Classe/Objet à stocker en un indice nommé
7  *| tab | Table de stockage à indice nommé (type MOT)
```

Gibi+ : couche « objet » applicative

- **GibianeObjet** : superclasse *cosmique*

La classe GibianeObjet

La classe *GibianeObjet* est la superclasse de toutes les autres classes.

```
1  DEBM meo18ii ;
2  *Classe GibianeObjet (Superclasse "cosmique")
3      %className = MOT 'GibianeObjet' ;
4      %methode Constructor meo18io ;
5  *
6      %methode instanceOf meo17jx ;
7  FINM ;
```

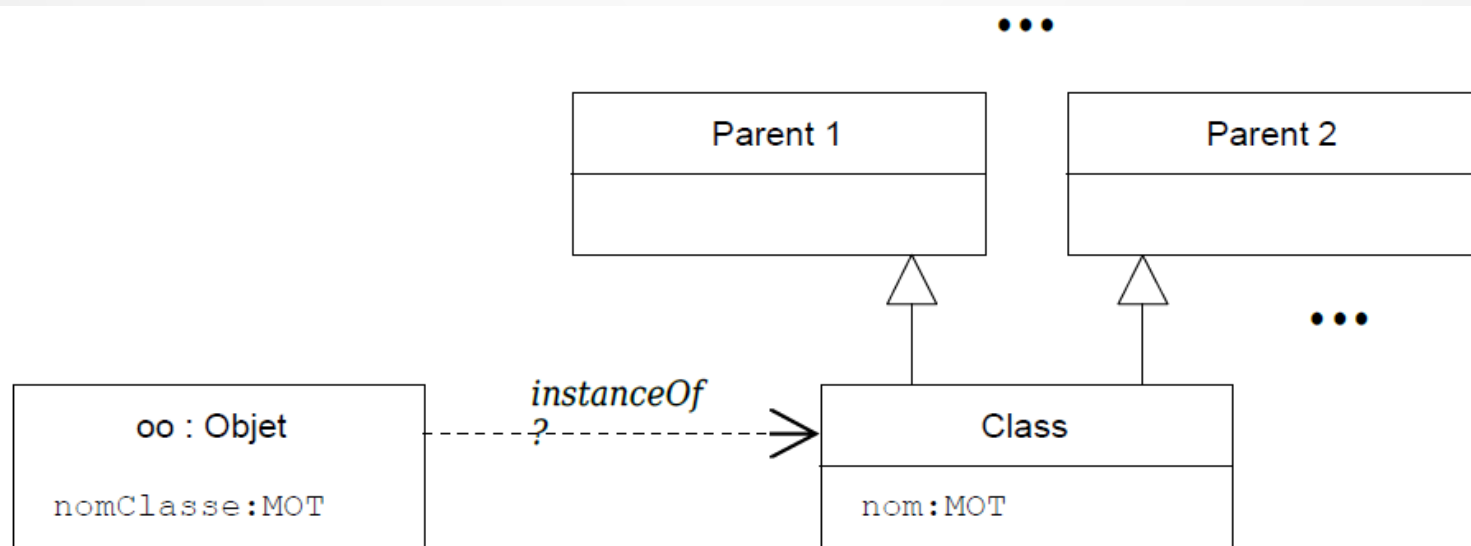
Gibi+ : couche « objet » applicative

- Méthode *instanceOf()*

La méthode *instanceOf()*

La méthode *instanceOf()* dit si l'objet courant est une instance de la classe fournie en argument. La règle d'écriture de l'appel à la méthode *instanceOf()* est la suivante :

```
boo = oo%instanceOf className ;
```



Gibi+ : couche « objet » applicative

- Exemples de classes de Gibi+ :

Classe « ObjetElastique »

```
1  DEBM mep04fy ;
2  *Classe ObjetElastique
3      %className = MOT 'ObjetElastique' ;
4      %super = gbClass.'GibianeObjet' ;
5
6      %methode Constructor    mep04fe ;
7      %methode new           mep04fr ;
8  *
9      %methode setMateriau   mep04ft ;
10     %methode bloquer       mep04fz ;
11     %methode resoudre      mep04fa ;
12     %methode addPression   mep04fq ;
13     %methode appliquerFxyz mep04ce ;
14     %methode deplaImpose   mep04xz ;
15  *
16     %methode voirDeformee  mep04fs ;
17     %methode voirReaction  mep04xx ;
18     %methode voirSigma     mep04xw ;
19     %methode voirVonMises  mep04xq ;
20  *
21     %methode trouverEF     mep10mm ;
22  FINM ;
```

Gibi+ : couche « objet » applicative

Maillage

```
1  *-----
2  * Etude de cas : Maillage d'une barre de 5 elements CU20
3  *-----
4  | DENS 10 ;
5  *---Géométrie
6  *   + Points
7  | p1 = (gbClass.'Point3D')%new 0. 0. 0. ;
8  | p2 = (gbClass.'Point3D')%new 10. 0. 0. ;
9  *   + Ligne droite reliant les 2 points
10 | li = (gbClass.'LigneDroite')%new p1 p2 ;
11 *   + Surface (créée par translation de la ligne)
12 | ov = (gbClass.'Point3D')%new 0. 10. 0. ;
13 | os = (gbClass.'SurfaceLigneTranslation')%new li ov ;
14
15 *   + Volume (créé par translation de la surface)
16 | ow = (gbClass.'Point3D')%new 0. 0. 50. ;
17 | vt = (gbClass.'VolumeSurfTranslation')%new os ow ;
18
19 *---Maillage : Avant de mailler le volume, il faut mailler la ligne et
20 *   la surface
21 *   + Ligne maillée contenant 1 élément:
22 | li%maillerUniforme 1 ;
23 *   + Surface maillée en se basant sur les densités par défaut
24 | os%maillerDensites ;
25 *   + Volume maillé en se basant sur les densités par défaut
26 | vt%maillerDensites ; total = vt ;
27 *   + 2e face du volume
28 | suh = vt%face 2 ;
29 *   + 2e côté de la face précédente
30 | lih = suh%cote 2 ;
```

- Exemple d'application

Gibi+ : couche « objet » applicative

Données mécaniques

```
1  * Etude de cas : Maillage d'une barre de 5 elements CU20
2  * -----
3  *Maillage et modèle
4  ome = (gbClass.'ObjetElastique')%new vt ;
5
6  *Matériau
7      oma = (gbClass.'Aluminium')%getInstance ;
8      ome%setMateriau oma ;
9
10 *Blocages
11     enc1 = ome%bloquer os   'UZ' ;
12     enc2 = ome%bloquer li   'UY' ;
13     enc3 = ome%bloquer p1   'UX' ;
14
15 *Force déf. par composantes
16     zf = ome%appliquerFxyz lih 0. 1000. 0. ;
17
18 *Déplacement imposé
19     zd = ome%deplaImpose suh 'UZ' 1.E-9 ;
20
21 *Résolution
22     zr = ome%resoudre ;
23
24 *Visu déformée
25     ome%voirDeformee ;
26     ome%voirReaction ;
27     ome%voirSigma ;
28     ome%voirVonMises ;
```

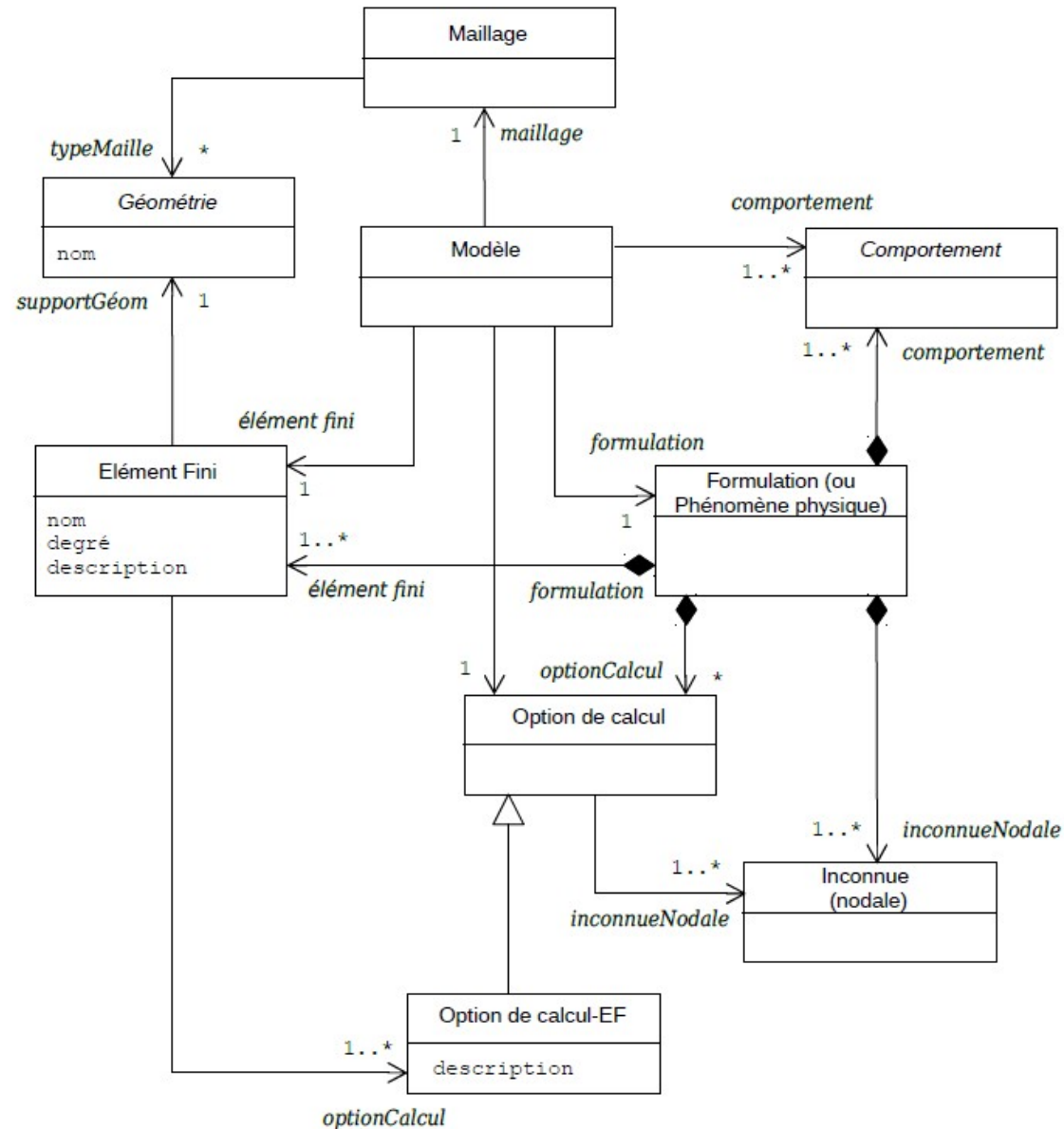

Gibi+ : couche « objet » applicative

Constructeur

```
1  DEBM mep04fe this*OBJET maya*OBJET ;
2  *Constructor()
3      this%methode super meil5ga ;
4      boo = this%super this ;
5      si boo ;
6          oef = %trouverEF fme maya ;
7          boo = unObjet oef ;
8      si boo ;
9          %mesh = maya ;
10         *
11             %efxx = fme%getElementFini 'CU20' ;
12             %efxx = oef ;
13             %cmxx = fme%getComportement 'ELASTIQUE' ;
14         *
15             %ocxx = fme%getOptionCalcul 'TRID' ;
16             tt = (%.efxx).optionCalcul ; ix = INDEX tt ;
17             %ocxx = tt.(ix.1) ;
18             boo = unObjet %.ocxx ;
19         si boo ;
20             si (EGA ix.1 'TRID') ;
21                 OPTI DIME 3 ; sinon ; OPTI DIME 2 ;
22             finsi ;
23         *
24             Modèle
25             %moxx = (gbClass.'Modele')%new maya fme
26                 (%.efxx) (%.cmxx) (%.ocxx) ;
27             %blocages = TABLE ;
28             %forces = TABLE ;
29         finsi ;
30     finsi ;
31     finsi ;
32     FINM boo ;
```

Gibi+ : couche « objet » applicative

Modèle

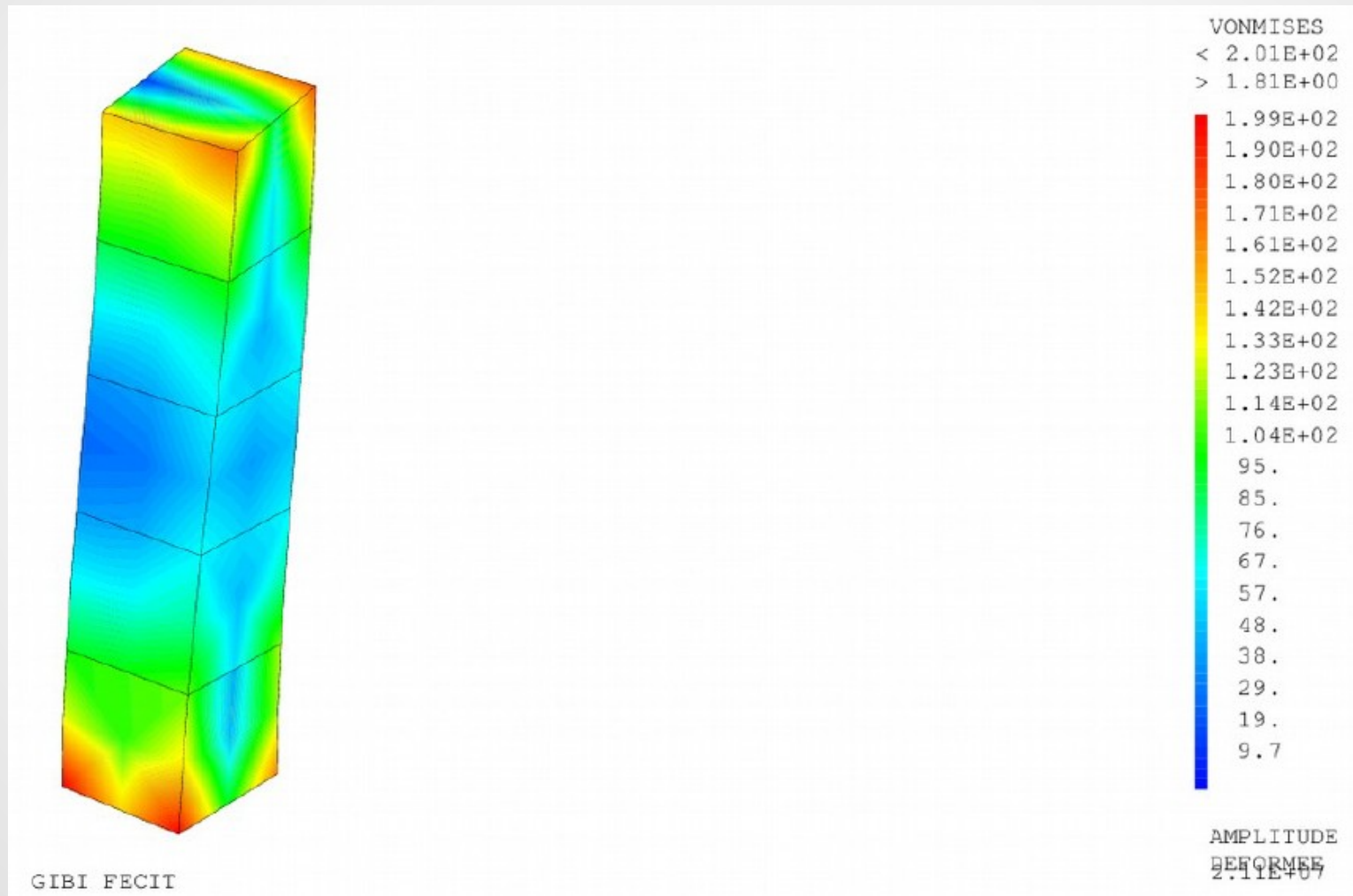


Gibi+ : couche « objet » applicative

Méthode resoudre()

```
1  DEBM mep04fa ;
2  *resoudre ()
3      n = DIME %.blocages ;
4      boo = n > 0 ;
5      si boo ;
6  *      - Rigidité totale
7          tt = TABLE ; tt.1 = %.matrRigidite ;
8  REPE cep04a n ; i = &cep04a ;
9          tt.(i+1) = (%.blocages).i ;
10 FIN cep04a ;
11      ku = (gbClass.'Rigidites')%new tt ;
12
13 *      - Force totale
14      f = (gbClass.'ForceT')%new (%.forces) ;
15
16 *      - resolution
17      %uxxx = (%.moxx)%resoudre ku f ;
18
19      %reaxx = (%.moxx)%creerReaction (%.uxxx) ;
20      %sigxx = (%.moxx)%calculerSigma (%.uxxx) ;
21      %vmxxx = (%.moxx)%calculerVonMises (%.uxxx) ;
22  finsi ;
23  FINM boo ;
```

Gibi+ : couche « objet » applicative



Gibi+ : couche « objet » applicative

- Exemples de classes de Gibi+ :

Conduction thermique (formulation)

```
1  DEBM mep17ue ;
2  *Classe ConductionThermique
3      %className = MOT 'ConductionThermique' ;
4      %super = gbClass.'FormulationThermique' ;
5
6      %methode Constructor      mep17ur ;
7      %methode getInstance      mep17uz ;
8  *  Méthodes concrètes obligatoires
9      %methode finalize         mep17ua ;
10     %methode getMateriau      mep18it ;
11 *  Méthodes de mise en données et de résolution spécifiques
12     %methode getObjectCdLim   mep19om ;
13     %methode creerCdLim       mep19op ;
14     %methode resoudre         mep19ob ;
15  FINM ;
```

Gibi+ : couche « objet » applicative

- Exemples

```
1  *Exemple inspiré de la formation pour débutant Cast3m
2  *(réf. http://www-cast3m.cea.fr/html/formations/Debuter\_avec\_Cast3M.pdf)
3  *
4  *1) Demi-plaque gauche = plqG
5
6  *   Géométrie: RectangleEntaille (au coin 2)
7  x = (gbClass.'RectangleEntaille')%new 1.2  1.  0.2 ;
8
9  *   Mailleur...
10 xm = x%getMailleur ;
11 *   ...données de maillage
12     xm%setDensiteIni 0.05 ;
13     xm%setDensiteFin 0.1 ;
14     (xm.'mailleurLigne1')%setNbElements 20 ;
15     (xm.'mailleurLigne2')%setNbElements 20 ;
16 *   xm%tracer ;
17 plqG = xm ;
18 *
19 *2) Demi-plaque droite = plqD (symétrique de plqG)
20
21     plqD = (gbClass.'MaillageSymetrique')%new plqG
22           (x%getPoint2) (x%getPoint3) ;
23 *3) Assemblage des deux morceaux:
24     plq = (gbClass.'Maillages')%new plqG plqD ;
25     plq%tracer ;
26
27 *4) Création des composants du maillage
28
29 * - Contour du maillage
30     csu = plq%getContour ;
31
```

(à suivre)

Gibi+ : couche « objet » applicative

● Exemple

```
1  *          --- Conduction thermique ---
2  *
3  * - Formulation
4  *   fm = (gbClass.'ConductionThermique')%getInstance ;
5  * - Élément fini
6  *   ef = fm%getElementFini 'QUA8' ;
7  * - Matériau (caractéristiques)
8  *   alu = (gbClass.'Aluminium')%getInstance ;
9  * - Hypothèse invariance matériau
10 *   hy = (gbClass.'Isotropie')%getInstance ;
11 * - Modele
12 *   mo = (gbClass.'Modele-v2')%new fm plq ef alu hy ;
13 * - Conditions aux limites
14 *   ok = mo%setCdLimite temperatureImposee
15 *       |       |       |       |
16 *       |       |       |       |
17 *       |       |       |       |
18 *       |       |       |       |
19 *   ok = mo%setCdLimite temperatureImposee
20 *       |       |       |       |
21 *       |       |       |       |
22 *       |       |       |       |
23 *   maillage (csu%getComposant 'arcCercle')
24 *       |       |       |       |
25 *       |       |       |       |
26 *       |       |       |       |
27 *       |       |       |       |
28 *   temperature 250. ;
29 *   ok = mo%setCdLimite temperatureImposee
30 *       |       |       |       |
31 *       |       |       |       |
32 *       |       |       |       |
33 *       |       |       |       |
34 *   maillage lgauc temperature 25. ;
35 * - Résolution
36 *   u = mo%resoudre ;
37 * - Visualisation des résultats (déplacement)
38 *   trac u (plq%getMaillage) ;
```


Gibi+ : couche « objet » applicative

- Exemples de classes de Gibi+ :

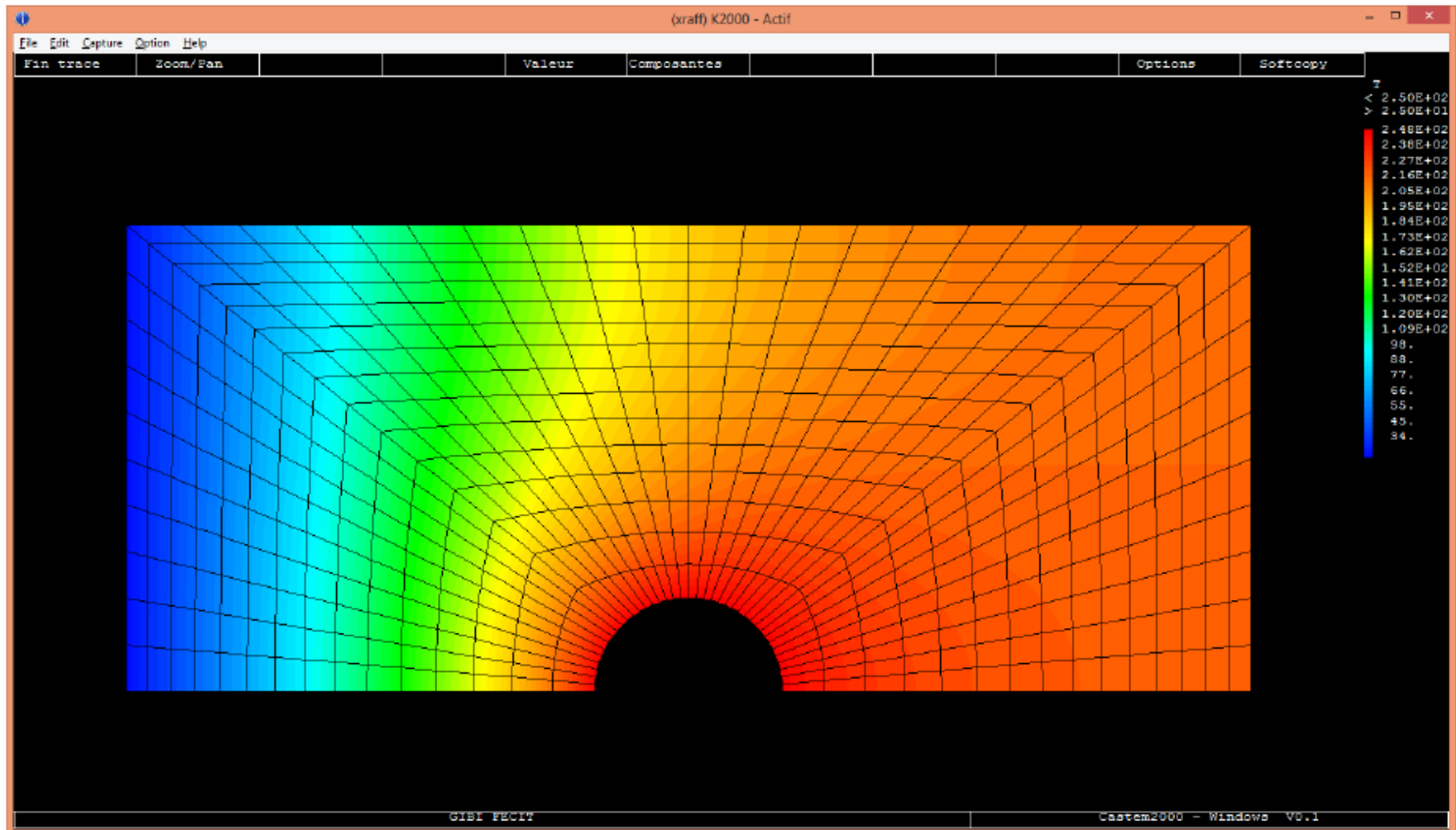
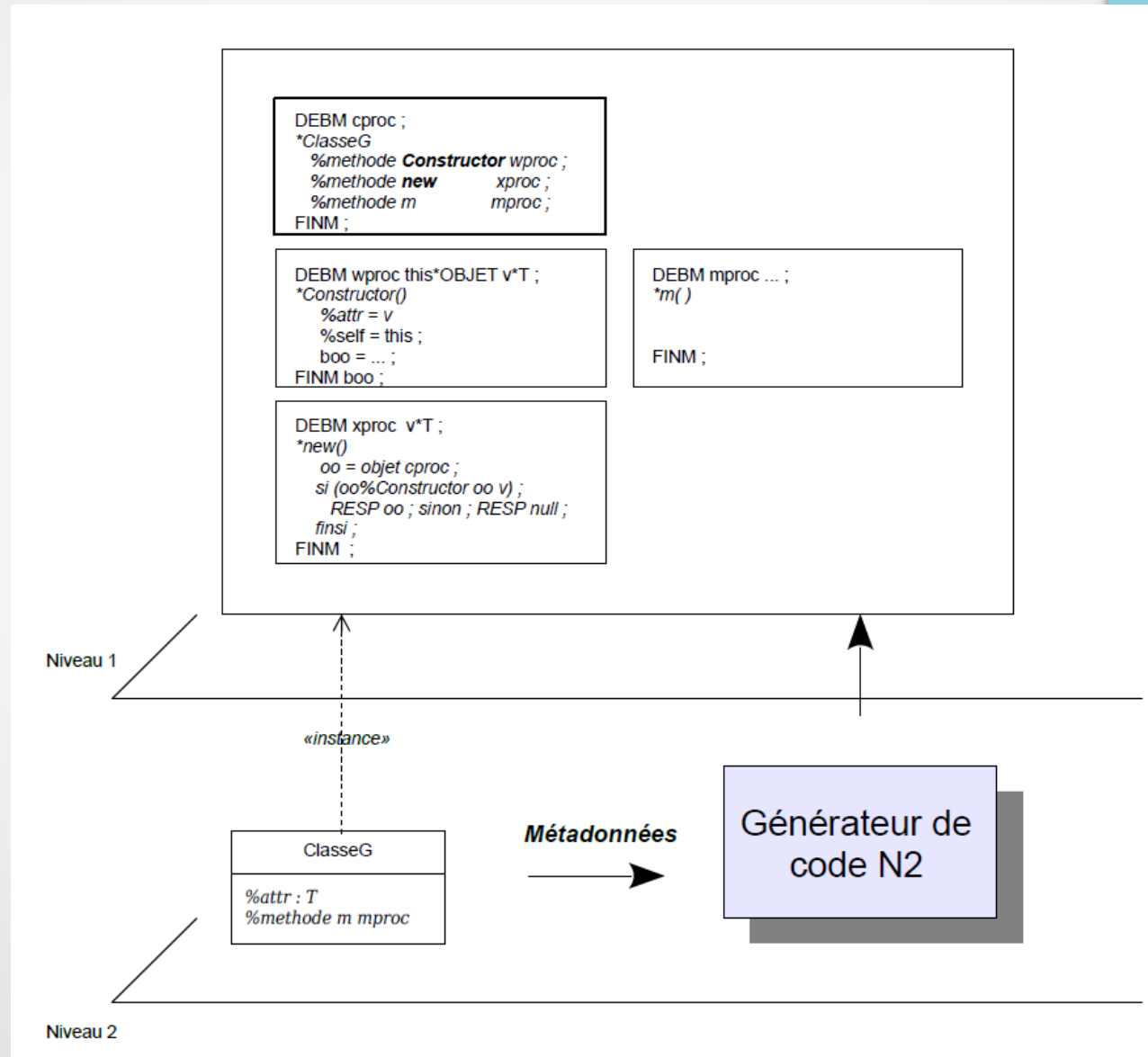


Figure 5: Visualisation des températures à l'état stationnaire

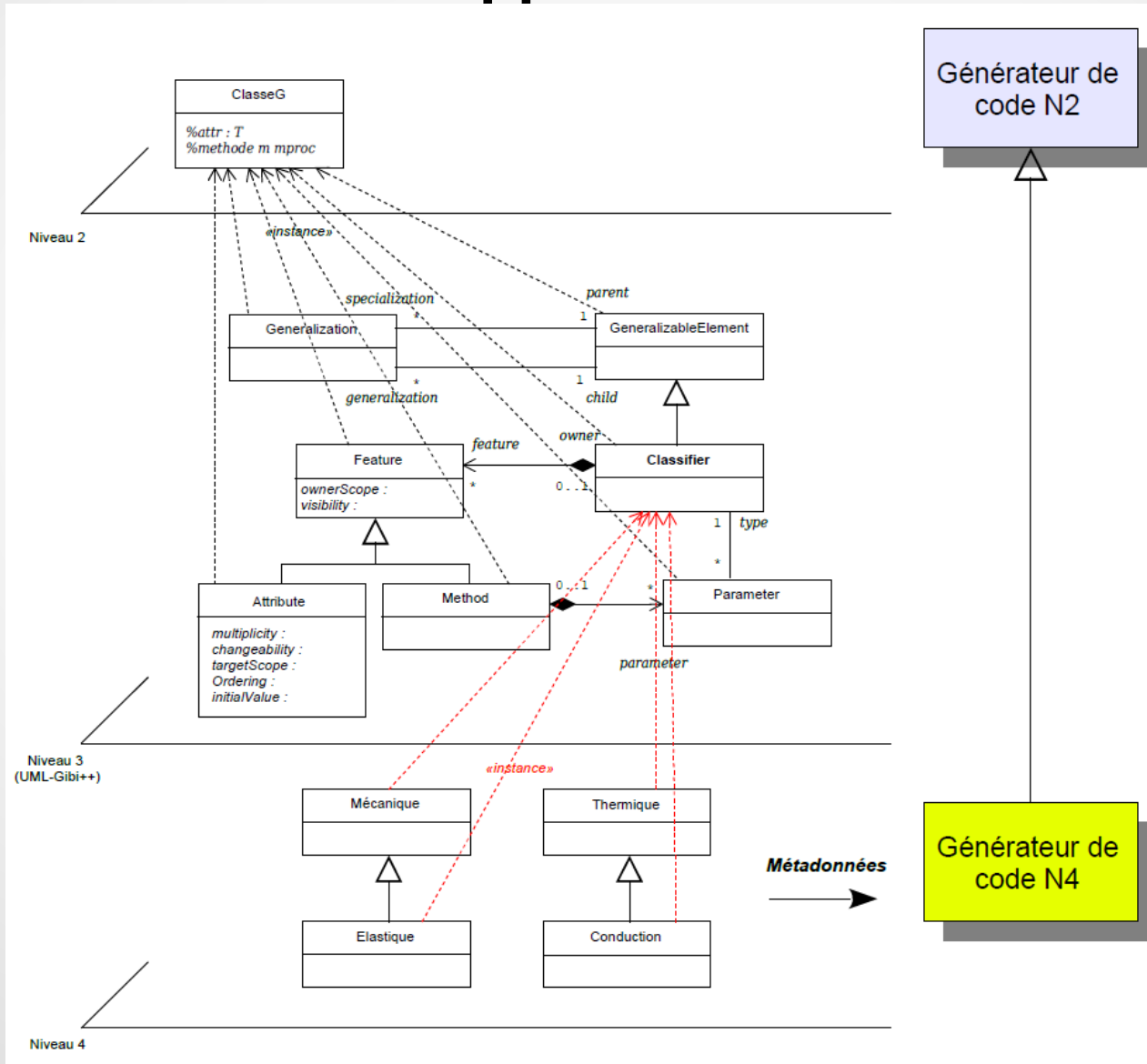
Gibi+ : Générateur automatique de code

- Générateur de classes Gibi+ :



Gibi+ : Générateur automatique de code

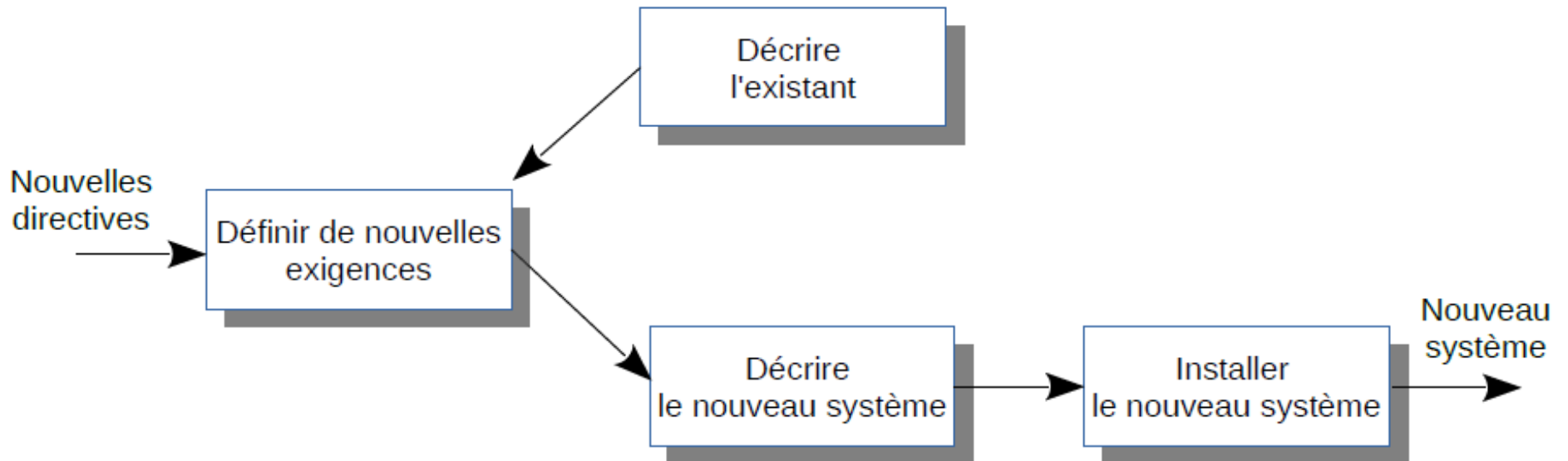
- Générateur de code applicatif



Gibi+ : Générateur automatique de code

- Conclusion

Processus formel de réingénierie



Gibi+ : Générateur automatique de code

- **Merci de votre attention**

A la mémoire de Thierry CHARRAS

Gibi+ : Générateur automatique de code

- **Des questions ?**