1

ConForM

1ère Journée Scientifique du Groupe ConForM

DOOSE
Distributed Object Oriented Software Engineering

ConForM Group
(Concurrency and Formal Methods)

Mathieu Buffo
Centre Universitaire d'Informatique
Université de Genève
Suisse

email : buffo@cui.unige.ch

2

ConForM

DOOSE
Généralités

DOOSE :
- O.O.S.E. ⇒ *modélisation*, expressivité, sécurité, ré-utilisation
- D. ⇒ *exécution* sur une plate-forme distribuée

Modélisation :
- objets
- coordination

⇒ sémantique par couches

Exécution :
- données
- opérations internes
- opérations externes

⇒ sémantique par couches

ConForM

Objets

Leur nature :
- services encapsulés
- unités de modélisation bien délimités — CO-OPN : avec encapsulation

Leur sorte :
- passifs
- réactifs
- actifs — CO-OPN : utilisation des méthodes

Leur concurrence :
- à l'intérieur de l'objet (parallélisme) — pb : gestion des synchronisations internes
- entre les objets (distribution) — se marie bien avec la notion d'objets actifs

Leur dynamique :
- statiques
- dynamiques — démiurge pour "create", "destroy" et "self"

Leurs relations :
- sous-typage
- héritage (sous-classage)
- clientèle

} ⇒ clientèle privée, délégation

ConForM

Coordination

Interactions :
- requête/réponse — - notion de client et de serveur; ex : acteurs
  - lié aux objets vus comme des services encapsulés
- échange d'information — interactions libres; ex : Linda

Synchronisations :
- faibles
- fortes — CO-OPN : naturel

Synthèse d'un système
- par opération de composition — objet, objet -> objet
- par opération de coordination — composante, composante -> composante

un objet est une composante particulière
une composante = un système

coordination = lien entre les méthodes d'entrée et de sortie ?

ConForM

Données

Caractéristiques
- statiques
- représentations différentes suivant les machines

Tolérance aux fautes lors des transmissions

En CO-OPN
- assimilation aux types abstraits algébriques

ConForM

Opérations internes

Sélection des opérations
- priorités — CO-OPN : utilisation des méthodes !
- synchronisations internes — CO-OPN : oui mais...

En CO-OPN
- prédicats pour l'expression des conditions; ex :
  xxx = yyy, *empty* place, valeur *into* place, exists objet, *sametype* xxx yyy
- variables
  - places pas pratiques dans certains cas
  - ajouter la notion de valeur constante

ConForM

Contrôle externe

Configurations statiques

- possibilité de description simple — ex : MIL, COIL
- pour les objets et pour les liens — ex de la fabrique de chocolat
- étude sur les objets

Configurations dynamiques

- reconfigurations dynamique
- pour les objets avec les appels au démiurge
- pour les liens par utilisation de références
- étude sur les classes

Migration des objets

- les configurations dynamiques peuvent migrer
- dans les configurations statiques
- notion d'appartenance v/s référence

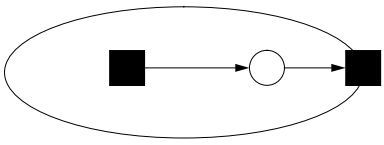
En CO-OPN

- tout est à faire ...

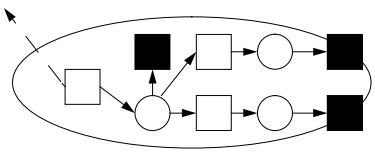
ConForM

Exemple : utilisations diverses des méthodes

Génération d'activité :



Mécanisme de priorités :



ConForM

Exemple : COIL

```
Generic Class tapisroulant(matériau);
Interface
        Type tapis roulant;
        Method délivre _ : matériau;
        Connector accepte _ : matériau;
Body
        Transition prend;
        Place récipient _ : matériau (1);
        Axioms
                prend With accepte x : -> récipient x;
                délivre x : récipient x -> ;
        Where x : matériau;
End;


Component ProductionDeTruffes;
Use
        ProducteurTruffes;
        TapisRoulants;
Objects
        Ptruffe : producteur de truffe;
        Ttruffe : tapis roulant de truffe;
Connections
        Ttruffe.accepte c With Ptruffe.délivre c;
Where
        c : chocolat;
End;
```


ConForM

Exemple : aspects statiques, aspects dynamiques

```
Class système;
Interface
   Use balise, relais;
   Types contrôleur; contrôleur de balises;
   Objects
        contrôleur : contrôleur;
        relais 1, relais 2, relais 3 : relais;
Body
        Method génère une nouvelle balise;
        Transition start;
        Place relais _ : relais;
        Initials
                relais relais 1, relais relais 2, relais relais 3;
        Axioms
                start With
                    relais 1.init pointeur droite relais 2 //
                    relais 2.init pointeur gauche relais 1 //
                    relais 2.init pointeur droite relais 3 //
                    relais 3.init pointeur gauche relais 2 : -> ;

                génère une nouvelle balise
                    With create b // r.nouvelle balise b :
                    relais r -> relais r;
   Where
        b : balise;
        r : relais;
End système;
```

ConForM

Conclusions
CO-OPN et DOOSE

Aspects de modélisation
- pas mal dans l'ensemble
- clientèle privée mal mise en valeur
- les méthodes servent à trop de choses !!!! A revoir !

Aspects de distributions
- Encore beaucoup à étudier :
  - COIL
  - migrations
  - tolérance aux fautes
- Choix à effectuer
- Exemples disponibles : fabrique de chocolats, balises mobiles

Réalisation d'une plate-forme distribuée pour CO-OPN/2
- Et Corba ?
- Utilisation de Java ?
- ET SI ON COMMENCAIT PAR UN SIMPLE CHECKER ...