

Université de Monastir  
Institut Supérieur d'Informatique et de Mathématiques

## Cours: Approche à base de composants et concurrence

Filière: MR-GL2

### Chapitre 2: Description de composants en UML2.0/PoSM

Réalisé par:  
Dr. Sakka Rouis Taoufik

1

#### Chapitre 2: Description de composants en UML2.0/PoSM

##### I. Introduction

- ❖ UML : Unified Modeling Language
- ❖ Langage utilisant une notation graphique
- ❖ Notation standard pour la modélisation d'applications à base d'**objets** (et de **composants** depuis la version 2)
- ❖ Utilisable dans de nombreux autres contextes de **conception** ou **spécification**
- ❖ Normalisé par l'OMG (Object Management Group)  
<http://www.omg.org/spec/UML/>
- ❖ Dernière version : 2.5.1 (Décembre 2017)

2

## Chapitre 2: Description de composants en UML2.0/PoSM

### I. Introduction

#### 1) Historique

- ❖ UML hérite principalement des méthodes objets de Booch (Booch), OMT (Rumbaugh) et OOSE (Jacobson)
  - Mais intègre également d'autres approches, comme les machines à états de Harel
- ❖ But initial : Définir un processus/méthode de développement complet (de l'analyse à l'implémentation) orienté objet
- ❖ Problème : Pas de notation, langage pour écrire les **modèles ou les artefacts** définis par ce processus
  - ➔ ceci devenu le but final d'UML
- ❖ UML n'est donc pas une méthode ou un processus
- ❖ UML propose un ensemble de modèles pour que chacun ait à sa disposition les éléments nécessaires à la conception d'une application.

3

## Chapitre 2: Description de composants en UML2.0/PoSM

### I. Introduction

#### 2) Modèle/Diagramme

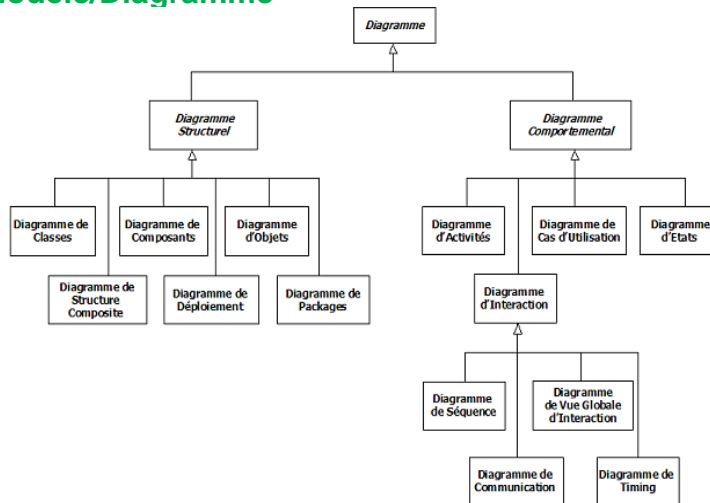
- ❖ Un modèle (ensemble de diagrammes) est une représentation partielle de la réalité:
  - Abstraction de ce qui est intéressant pour un contexte donné
  - Vue subjective et simplifiée d'un système
  - Avec UML, on va s'intéresser principalement aux modèles d'applications informatiques
  - Faciliter la compréhension d'un système
  - Permettre également la communication avec le client

4

## Chapitre 2: Description de composants en UML2.0/PoSM

### I. Introduction

#### 2) Modèle/Diagramme



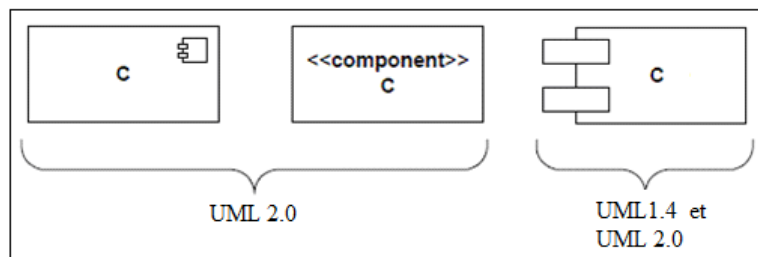
5

## Chapitre 2: Description de composants en UML2.0/PoSM

### II. Diagramme de composants

#### ❖ 1) Composant

- ❖ Un composant UML2.0 représente une partie modulaire d'un système qui encapsule son contenu et qui est remplaçable au sein de son environnement.
- ❖ 3 représentations graphiques proposées par UML2.0



6

## Chapitre 2: Description de composants en UML2.0/PoSM

### II. Diagramme de composants

#### 1) Composant

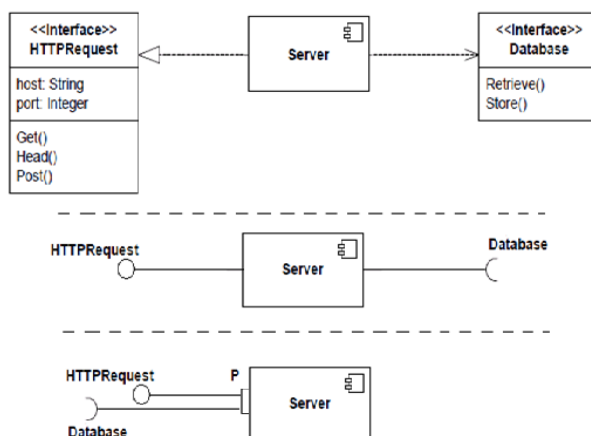
- ❖ Un composant UML2.0 interagit avec son environnement par l'intermédiaire des **points d'interactions appelés ports ou interfaces**.
- ❖ Un port peut rassembler des interfaces offertes et/ou requises.
- ❖ Une interface définit un ensemble d'**opérations**, d'**attributs** et de **contraintes**.
- ❖ **Interface offerte (ou fournie)** → caractérise le comportement fourni par le composant.
- ❖ **Interface requise** → caractérise le comportement attendu

7

## Chapitre 2: Description de composants en UML2.0/PoSM

### II. Diagramme de composants

#### 1) Composant



8

## Chapitre 2: Description de composants en UML2.0/PoSM

### II. Diagramme de composants

#### 1) Composant

##### Remarques:

- ❖ Le comportement d'une interface est décrit par une machine à états
- ❖ Le comportement d'un port est issu de la composition des comportements de ses interfaces.
- ❖ La seule contrainte pour pouvoir substituer un composant par un autre est de **respecter les interfaces offertes/requises**

9

## Chapitre 2: Description de composants en UML2.0/PoSM

### II. Diagramme de composants

#### 2) Structure composite

- ❖ UML2.0 étend les diagrammes de composants d'UML1.4 d'une simple représentation physique vers la prise en compte des composants logiques.
- ❖ Ainsi, deux vues complémentaires sont possibles pour modéliser les composants UML2.0 :
  - **Une vue externe (boîte noire)** : définit le composant comme un élément exécutable du système
  - **Une vue interne (boîte blanche)**: définit un composant comme **une structure composite**, elle étend la première en définissant le composant comme un ensemble cohérent des parties appelées parts. Chaque partie représente une instance d'un autre composant.

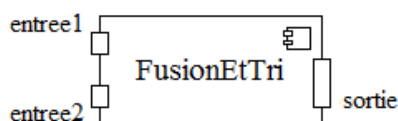
10

## Chapitre 2: Description de composants en UML2.0/PoSM

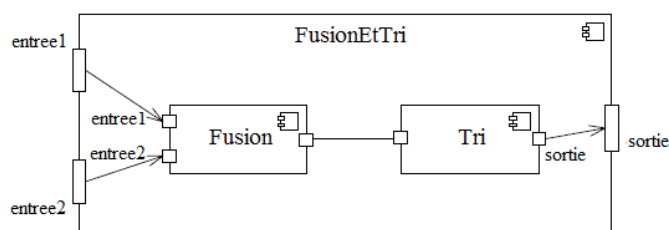
### II. Diagramme de composants

#### 2) Structure composite

Vue externe d'un composant FusionEtTri



Vue interne d'un composant FusionEtTri



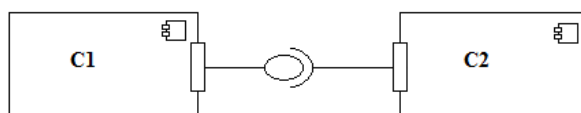
11

## Chapitre 2: Description de composants en UML2.0/PoSM

### II. Diagramme de composants

#### 3) Connecteur

- ❖ La connexion entre les ports ou les interfaces des composants se fait au moyen de **connecteurs**.
- ❖ On distingue deux types de connecteurs:
  - connecteurs d'assemblage
  - connecteurs de délégation
- ❖ **Un connecteur d'assemblage** relie deux composants.
- ❖ Ce connecteur joint **une interface requise** (un port requis) à **une interface offerte** (un port fourni)



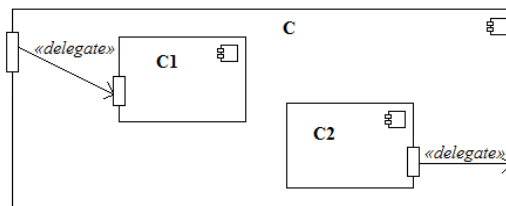
12

## Chapitre 2: Description de composants en UML2.0/PoSM

### II. Diagramme de composants

#### 3) Connecteur

- **Les connecteurs de délégation** relient les ports d'un composant aux éléments qu'il contient.
- La délégation peut se faire sur plusieurs niveaux (composition imbriquée).
- **RQ.** Deux interfaces (ports) reliées par un connecteur de délégation doivent être de types compatibles (les deux sont requis ou les deux sont offerts).

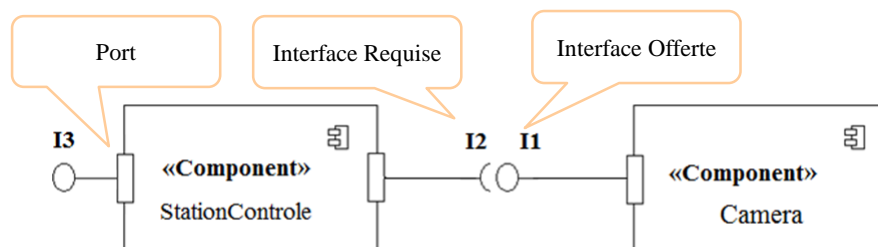


13

## Chapitre 2: Description de composants en UML2.0/PoSM

### II. Diagramme de composants

#### 4) Exemple: Niveau 1

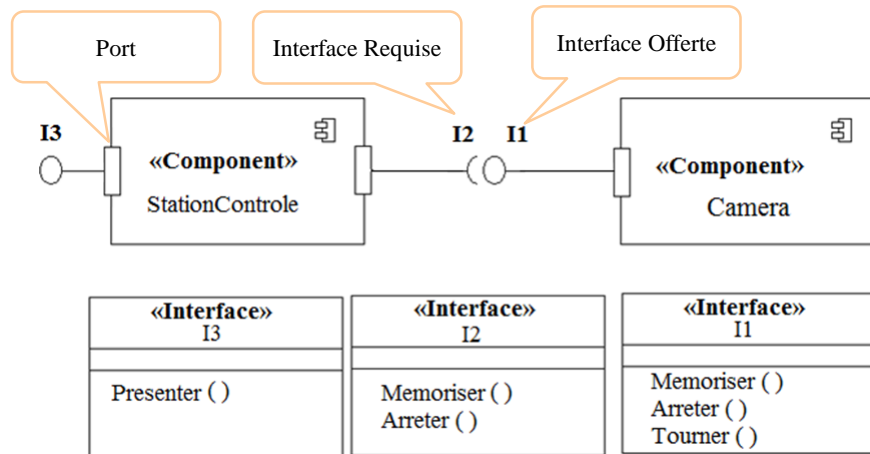


14

## Chapitre 2: Description de composants en UML2.0/PoSM

### II. Diagramme de composants

#### 4) Exemple: Niveau 2



15

## Chapitre 2: Description de composants en UML2.0/PoSM

### II. Diagramme de composants

#### Exercice 1: Le système GAB

Le système GAB est composé de deux composants dont le premier appelé Client est doté d'un port P1 qui propose une interface qui exige deux services (solde et retirer) alors que le second appelé Serveur est doté d'un port P2 qui propose une interface offerte qui regroupant trois services (solde, retirer et déposer).

Pour cette spécification informelle, proposer une conception architecturale (niveau 2) en UML2.0

16



## Chapitre 2: Description de composants en UML2.0/PoSM

### II. Diagramme de composants

#### **Exercice 2: Le système CaméraVidéo**

Le système CaméraVidéo numérique est composé essentiellement par trois éléments : Une Caméra qui capture des séquences vidéo, une Mémoire qui lui permet d'enregistrer ces séquences et d'un VideoPlayer permettant la présentation des séquences déjà enregistrées.

Pour mieux comprendre le fonctionnement de ce système, nous allons essayer de détailler les trois composants qui lui constituent :

1- Le premier composant appelé Camera, propose une interface appelée Memorisation. Cette interface comporte deux services (Memoriser et Arrêter).

2- Le second composant appelé Memory, propose à son tour deux interfaces. La première exige les services proposés par l'interface Memorisation du composant Caméra. La seconde, appelée VideoStream, fournit les quatre services (Lire, Avancer, Rebobiner et Arrêter).

3- Le troisième composant appelé VideoPlayer exige une interface VideoStream et fournit une autre interface appelée VideoPresented. Cette interface définit le service Presenter.

Proposer une conception architecturale (Niveau 2) pour cette application en utilisant le diagramme de composants UML2.0.

17

## Chapitre 2: Description de composants en UML2.0/PoSM

### III. Diagramme d'états transition: SM, PSM, PoSM

❖ UML2.0 distingue deux modèles pour la représentation des machines à états:

- State Machine: (SM)
- Protocol State Machine: (PSM)

18

## Chapitre 2: Description de composants en UML2.0/PoSM

### III. Diagramme d'états transition: SM, PSM, PoSM

#### 1) Diagramme d'états : SM

##### ❖ Comportement interne d'un objet:

- La définition de tous les états possibles d'un objet
- La définition de tous les changement d'états via des transitions

##### ❖ Associé à un objet ou à une opération

##### ❖ Syntaxe d'une transition: événement [garde] /action

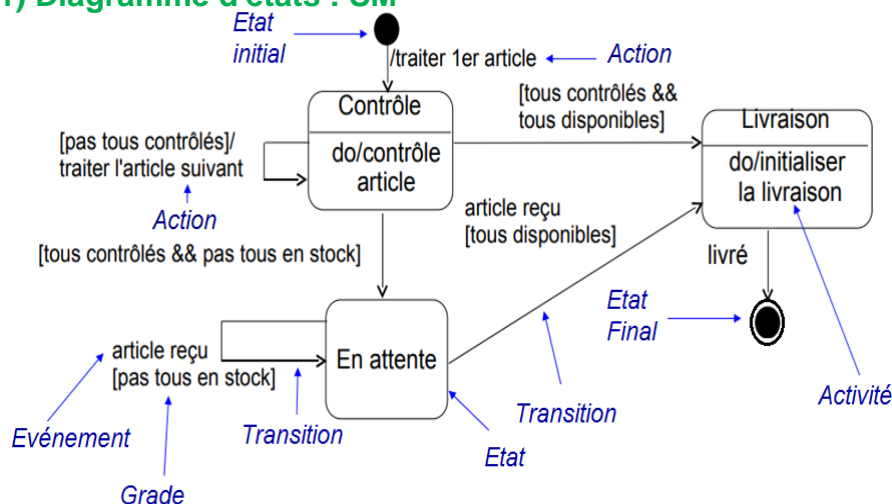
- Chaque partie est optionnelle
- La transition est suivie si l'événement a été généré **et** que la garde est valide → Exécute alors l'action **avant** de rentrer dans l'état ciblé par la transition

19

## Chapitre 2: Description de composants en UML2.0/PoSM

### III. Diagramme d'états transition: SM, PSM, PoSM

#### 1) Diagramme d'états : SM



20

## Chapitre 2: Description de composants en UML2.0/PoSM

### III. Diagramme d'états transition: SM, PSM, PoSM

#### 2) Diagramme d'états : PSM

- ❖ UML2.0 hérite d'UML1.x les SMs et introduit les PSMs pour modéliser le comportement d'un système.
- ❖ Contrairement à une machine à états de type **SM** qui **décrit les réactions d'un objet en réponse à des événements**, une machine à états de type **PSM** **spécifie les séquences légales des événements** qui peuvent se produire dans le contexte d'une classe ou d'un composant
- ❖ Les SMs permettent de décrire **le comportement** d'un classifieur (class, composant, etc.) dans le cadre de son implémentation. Alors que, les PSMs définissent **le protocole** des interfaces, qu'on peut considérer comme une spécialisation des UML1.x SM, **sans action, ni activité**.

21

## Chapitre 2: Description de composants en UML2.0/PoSM

### III. Diagramme d'états transition: SM, PSM, PoSM

#### 3) Diagramme d'états : PoSM

- ❖ **Prob.** Les PSMs ne peuvent décrire qu'un seul sens de la communication.
- ➔ Cela signifie également qu'un PSM ne peut pas décrire **les relations de communication entre les interfaces requises et fournies**.
- ❖ **Sol.** Nous avons retenus le profil PoSM (Port State Machine) (MencI, 2004).
  - Celui-ci est considéré comme une extension aux PSMs.
  - Les principales contributions de ce profil sont:
    - A) la possibilité de définir **des comportements requis ou fournis**
    - B) la possibilité de **distinguer une requête d'une réponse**.





22

## Chapitre 2: Description de composants en UML2.0/PoSM

### III. Diagramme d'états transition: SM, PSM, PoSM

#### 3) Diagramme d'états: PoSM

##### Transition PoSM

- Événement est préfixé par l'un des opérateurs (? ou !).
- Événement est suffixé par l'un des opérateurs (↑ ou ↓).
- ❖ L'opérateur de préfixage noté !  
 Envoi d'un événement initialisé.
- ❖ L'opérateur de préfixage noté ?  
 Réception d'un événement observé.
- ❖ L'opérateur de suffixage noté ↑  
 Demande d'exécution d'une méthode.
- ❖ L'opérateur de suffixage noté ↓  
 Réponse venant d'une méthode.

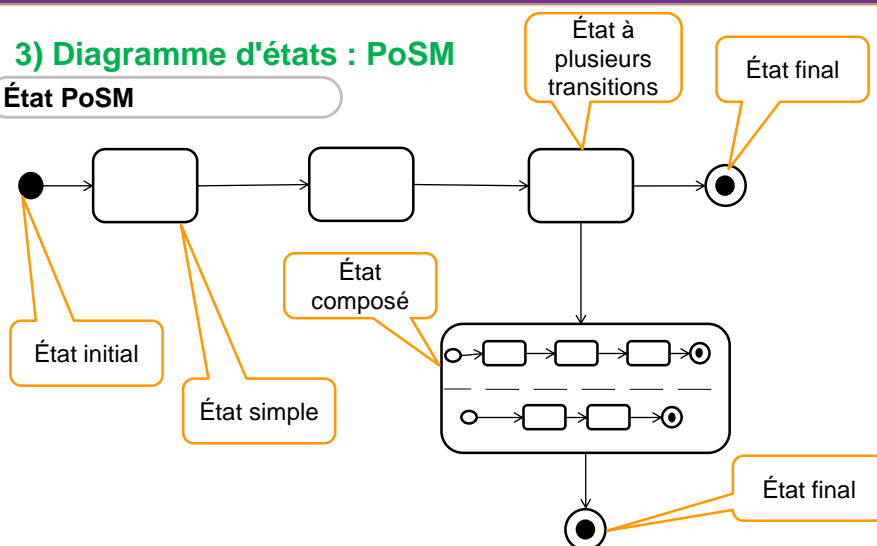
23

## Chapitre 2: Description de composants en UML2.0/PoSM

### III. Diagramme d'états transition: SM, PSM, PoSM

#### 3) Diagramme d'états : PoSM

##### État PoSM

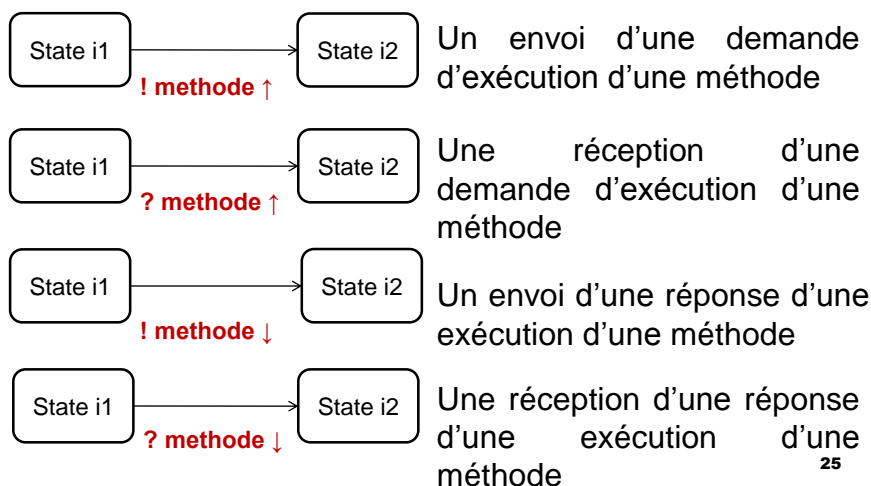


24

## Chapitre 2: Description de composants en UML2.0/PoSM

### III. Diagramme d'états transition: SM, PSM, PoSM

#### 3) Diagramme d'états : PoSM



## Chapitre 2: Description de composants en UML2.0/PoSM

### IV. Exercices d'application

#### Exercice 1:

Our system consists essentially of two components: an ATM that represents an automated silver teller machine, and a Bank component representing a subsystem of a bank.

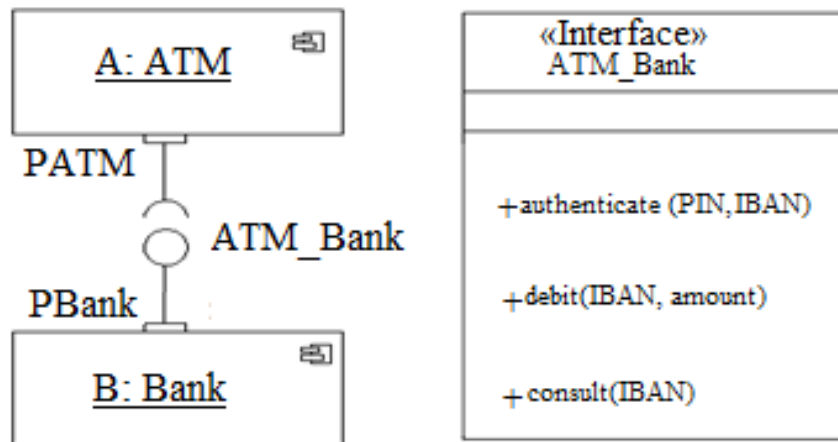
The operating system of these components is as follows: When inserting a bank card into an ATM, it must authenticate itself by sending a PIN code and the IBAN code of the card to the appropriate bank. At this time, the bank checks the validity of this information. If the codes are correct, the Bank component allows the ATM component to consult and/or debit from this account. To simplify this system, we ignore the part of ATM communication with the user.

Using the UML2.0 components model and the PoSM profile, propose a structural and behavioral description of the ATM system.

26

## Chapitre 2: Description de composants en UML2.0/PoSM

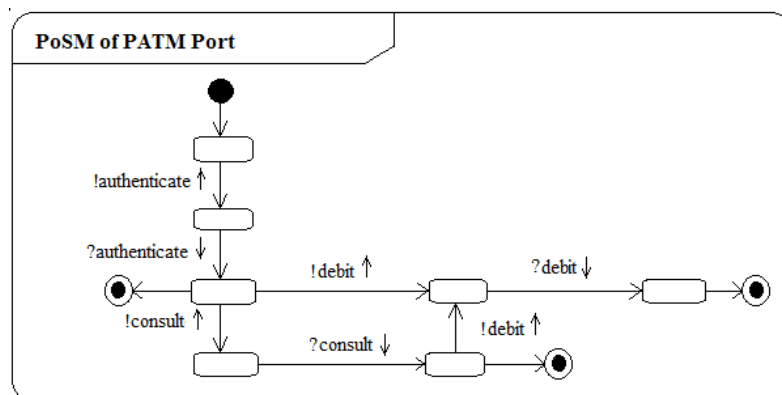
### IV. Exercices d'application



27

## Chapitre 2: Description de composants en UML2.0/PoSM

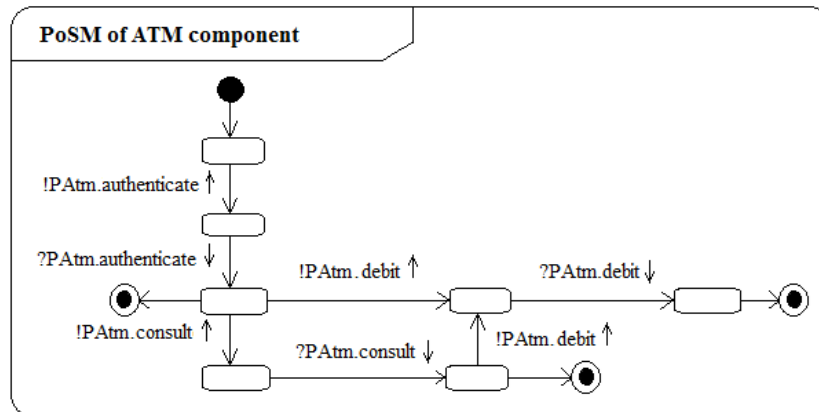
### IV. Exercices d'application



28

## Chapitre 2: Description de composants en UML2.0/PoSM

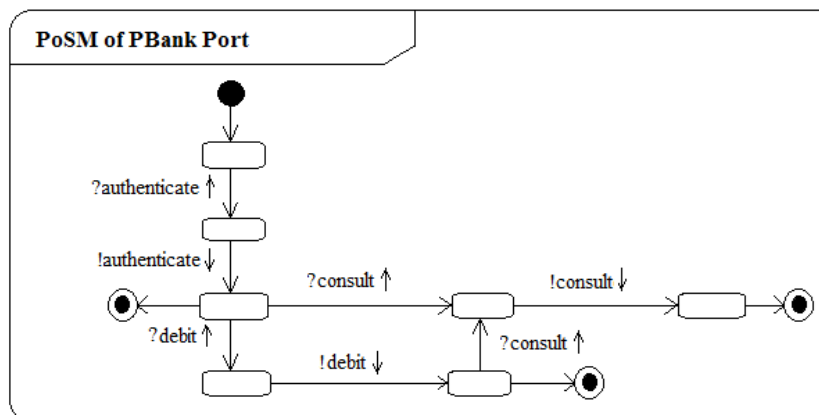
### IV. Exercices d'application



29

## Chapitre 2: Description de composants en UML2.0/PoSM

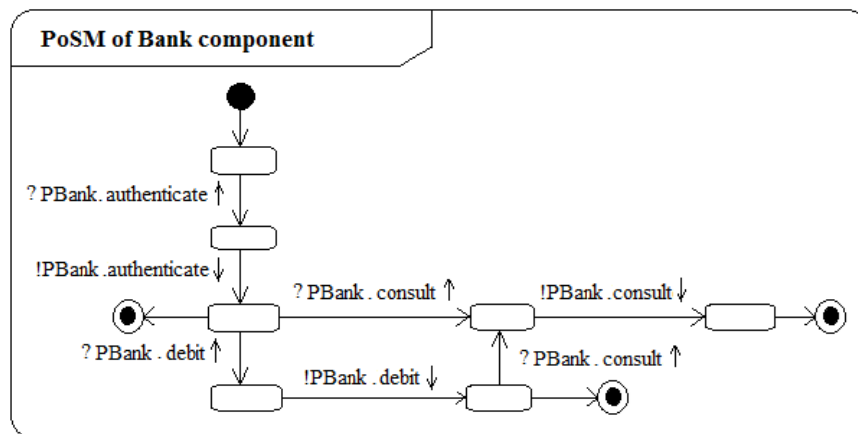
### IV. Exercices d'application



30

## Chapitre 2: Description de composants en UML2.0/PoSM

### IV. Exercices d'application



31

## Chapitre 2: Description de composants en UML2.0/PoSM

### IV. Exercices d'application

#### Exercice 2:

La description structurelle du système de compression de proxy (compressing proxy system) est décrite dans (Bernardo et al., 2002) par un composite composé de trois composants comme suit: Adopter, GifCompressing et TxtCompressing.

Le composant Adopter propose une interface offerte (Data) et deux interfaces requises (GifData et TxtData). Chaque interface est dotée d'une seule méthode appelée Compressing. Ce premier composant reçoit les données non compressées, délègue les données textuelles au composant TxtCompressing et les données graphiques au composant GifCompressing. Le composant TxtCompressing permet la compression des données textuelles vers des données zippées. Cependant le composant GifCompressing permet la compression des données graphique par conversion du GIF vers JPG. Enfin, le résultat de la compression sera ensuite envoyé aux clients.

En utilisant le diagramme de composants UML2.0 et le profile PoSM:

- 1) proposer une conception architecturale pour ce système.
- 2) proposer le protocole de l'interface GifData du composant Adopter
- 3) proposer le protocole de l'interface GifData du composant GifCompressing

32