

# Modélisation de systèmes complexes

Vincent Augusto

École Nationale Supérieure des Mines de Saint-Étienne

2012-2013

Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

Outils de  
modélisation

SADT/IDEF0  
IDEF3  
ARIS

UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

Réseaux de  
Petri

Conclusion

- 1 Introduction
- 2 Outils de modélisation
- 3 UML
- 4 Réseaux de Petri
- 5 Conclusion

## Introduction

### Présentation

L'approche POA  
Architectures de  
référence  
Vues

### Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

### UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

### Réseaux de Petri

### Conclusion

## La modélisation en entreprise...

- consiste à décrire l'organisation et les processus opérationnels d'une entreprise ;
- permet la simulation de ses processus pour la comparaison de différents scénarios ;
- permet l'analyse et la restructuration de l'entreprise pour l'amélioration des performances.

## Comprendre le fonctionnement :

- pour aider la prise de décision ;
- pour améliorer son fonctionnement (reengineering).

## Introduction

### Présentation

L'approche POA  
Architectures de  
référence  
Vues

### Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

### UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

### Réseaux de Petri

### Conclusion

## Motivations justifiant une étude de modélisation :

- Comprendre et analyser la structure et le fonctionnement de l'entreprise.
- Prévoir le comportement et les performances des processus opérationnels avant leur implantation.
- Choisir la meilleure alternative d'implantation.
- Identifier les risques d'implantation à gérer.
- Justifier les choix d'implantation sur des critères liés aux ressources et aux coûts.
- Bâtir une vision commune du fonctionnement de l'entreprise et la **communiquer** facilement au plus grand ensemble possible du personnel.

# Terminologie

## Introduction

### Présentation

L'approche POA  
Architectures de  
référence  
Vues

### Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

### UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

### Réseaux de Petri

### Conclusion

- **Entreprise** : industrie, banque, compagnie d'assurance, hôpitaux.
- **Modèle** : représentation d'une abstraction d'une partie du monde réel, exprimée dans un langage de représentation.
- **Activité** : accomplissement d'une tâche (séquence d'opérations).
- **Processus opérationnel** : succession de tâches qui contribue à la réalisation des objectifs de l'entreprise.

# L'approche processus-opération-acteur

## Introduction

### Présentation

### L'approche POA

### Architectures de référence

### Vues

## Outils de modélisation

### SADT/IDEF0

### IDEF3

### ARIS

## UML

### Diagrammes de classes et d'objets

### Diagrammes de cas d'utilisation

### Diagrammes de séquence

### Diagrammes d'activité

### Diagrammes d'état

### Applications

## Réseaux de Petri

## Conclusion

## Point de vue fonctionnel :

- Niveau macroscopique : ensemble de processus concurrents.
- Niveau microscopique : ensemble d'actions élémentaires réparties dans le temps et l'espace.
- Ces **opérations fonctionnelles** peuvent être regroupées en ensembles d'actions formant des activités pour réaliser des tâches.
- Trois concepts fondamentaux (général → atomique) : le processus, l'activité, l'opération.

# L'approche processus-opération-acteur

## Point de vue organisationnel :

- Ensemble de **ressources** à utiliser au mieux.
- Les ressources sont nécessaires à l'exécution des processus opérationnels.
- Les ressources peuvent être :
  - des moyens inertes (machines, outils) ;
  - des moyens financiers ;
  - des **acteurs**.

## Interface entre points de vue fonctionnel et organisationnel :

- Opérations fonctionnelles associées aux compétences.
- **Paradigme POA** : concilier le point de vue du monde de la gestion et celui du monde de la technique.

# Architectures de référence : deux concepts fondamentaux

## Introduction

Présentation  
L'approche POA

Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion

## La systémique :

- L'entreprise est considérée comme un système dynamique.
- Trois sous-systèmes :
  - **système opérant** : machines et hommes ;
  - **système de décision** : stratégies, objectifs, choix ;
  - **système d'information** : gestion des données, capitalisation des connaissances.

## Le génie logiciel :

- Décomposition fonctionnelle.
- Formalisation du concept de cycle de vie d'un système.
- Techniques structurées d'analyse et de conception.
- Outils de modélisation.

# Exemple d'architecture de référence : CIMOSA (1/4)

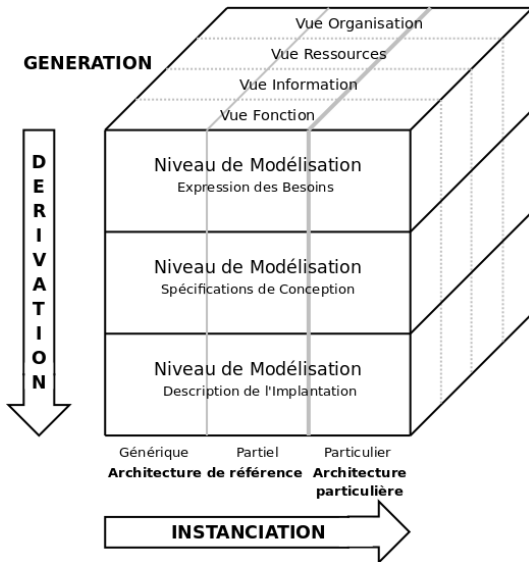
CIMOSA (CIM Open System Architecture) comprend :

- un cadre de modélisation ;
- une plate-forme d'intégration ;
- une méthodologie d'intervention.

Cadre de modélisation organisé autour de 3 axes :

- axe de généricité (3 niveaux d'instanciation) ;
- axe de dérivation (relatif au cycle de développement) ;
- axe de génération.

## Exemple d'architecture de référence : CIMOSA (2/4)



### Introduction

Présentation

L'approche POA

Architectures de  
référence

Vues

### Outils de modélisation

SADT/IDEF0

IDEF3

ARIS

### UML

Diagrammes de  
classes et  
d'objets

Diagrammes de  
cas d'utilisation

Diagrammes de  
séquence

Diagrammes  
d'activité

Diagrammes  
d'état

Applications

### Réseaux de Petri

### Conclusion

# Exemple d'architecture de référence : CIMOSA (3/4)

## Introduction

### Présentation

### L'approche POA

### Architectures de référence

### Vues

## Outils de modélisation

### SADT/IDEF0

### IDEF3

### ARIS

## UML

### Diagrammes de classes et d'objets

### Diagrammes de cas d'utilisation

### Diagrammes de séquence

### Diagrammes d'activité

### Diagrammes d'état

### Applications

## Réseaux de Petri

## Conclusion

## Axe de généricité :

- **niveau générique** (primitives de base) ;
- **niveau partiel** (structures réutilisables) ;
- **niveau particulier** (modèles spécifiques).

## Axe de dérivation (objet de l'étude) :

- **niveau de définition des besoins** (cahier des charges) ;
- **niveau des spécifications de conception** (solutions au besoin) ;
- **niveau de description de l'implantation** (implémentation).

# Exemple d'architecture de référence : CIMOSA (4/4)

## Introduction

### Présentation

#### L'approche POA

#### Architectures de référence

#### Vues

## Outils de modélisation

### SADT/IDEF0

### IDEF3

### ARIS

## UML

### Diagrammes de classes et d'objets

### Diagrammes de cas d'utilisation

### Diagrammes de séquence

### Diagrammes d'activité

### Diagrammes d'état

### Applications

## Réseaux de Petri

## Conclusion

## Axe de génération :

- **vue fonction** (description des processus opératoires) ;
- **vue information** (description des objets, de leurs relations et de leurs états) ;
- **vue ressources** (description des moyens nécessaires pour réaliser les fonctions) ;
- **vue organisation** (description de la distribution des responsabilités et des autorités dans les prises de décision).

## But de la modélisation CIMOSA :

- fournir un support à l'ingénierie systématique d'un système intégré de production ;
- participer à l'intégration du système en utilisant le modèle d'entreprise pour piloter les opérations de l'entreprise.

# Vue fonction

## Analyse des aspects fonctionnels :

- **Fonctionnalité** : description des tâches à exécuter.
- **Comportement** : ordre d'exécution des tâches de l'entreprise.
- Puis, prise en compte des **aspects dynamiques** du comportement de l'entreprise (temps et disponibilité des ressources).

## Activités et processus opérationnels :

- **Activité** : réalisation d'une tâche en transformant un état d'entrée en un état de sortie et ce, en utilisant du temps et des ressources.
- **Processus opérationnel** : ensemble partiellement ordonné d'activités dont l'exécution a pour but de contribuer à la réalisation d'un des objectifs de l'entreprise.

# Vue information

## Analyse des aspects informationnels :

- Représentation des objets modélisés précédemment et de leurs relations.
- Utilisation d'un langage de description des données.
- Objectif final : conception des bases de données servant à gérer les données du système.

## Système d'information :

- Ensemble des données et informations utilisées, stockées ou traitées pour les besoins d'utilisateurs ou d'applications d'une entreprise.
- Existence de plusieurs SI communicants selon le domaine.
- Système d'information  $\neq$  Base de données.

Introduction

Présentation

L'approche POA

Architectures de  
référence

Vues

Outils de  
modélisation

SADT/IDEF0

IDEF3

ARIS

UML

Diagrammes de  
classes et  
d'objets

Diagrammes de  
cas d'utilisation

Diagrammes de  
séquence

Diagrammes  
d'activité

Diagrammes  
d'état

Applications

Réseaux de  
Petri

Conclusion

## Introduction

### Présentation

### L'approche POA

### Architectures de référence

### Vues

## Outils de

### modélisation

### SADT/IDEF0

### IDEF3

### ARIS

## UML

### Diagrammes de classes et d'objets

### Diagrammes de cas d'utilisation

### Diagrammes de séquence

### Diagrammes d'activité

### Diagrammes d'état

### Applications

## Réseaux de Petri

## Conclusion

## Analyse des aspects liés aux ressources :

- Constituent le sous-système physique : moyens de production, de transport, opérateurs humains, etc.
- Motivation : évolution de l'automatisation et spécialisation des tâches des opérateurs humains.
- Structures de données décrivant aspects statistiques et fonctionnels/dynamiques.

## Définition et types de ressources :

- Les ressources matérielles (machines).
- Les ressources humaines (opérateurs).
- Les applications informatiques (logiciels).

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion

## Analyse des aspects organisationnels :

- Somme des moyens employés pour diviser le travail en tâches distinctes et pour ensuite assurer la coordination nécessaire entre ces tâches.
- **Structure organisationnelle** : réalisation d'un organigramme.
- **Coordination des tâches** : mécanismes permettant aux entités de s'informer et de se consulter :
  - ajustement mutuel ;
  - supervision directe/hiéarchique ;
  - par standardisation des objectifs.

Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

Outils de  
modélisation

SADT/IDEF0  
IDEF3  
ARIS

UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

Réseaux de  
Petri

Conclusion

1 Introduction

2 Outils de modélisation

3 UML

4 Réseaux de Petri

5 Conclusion

## Analyse des aspects fonctionnels :

- **Fonctionnalité** : description des tâches à exécuter.
- **Comportement** : ordre d'exécution des tâches de l'entreprise.
- Puis, prise en compte des **aspects dynamiques** du comportement de l'entreprise (temps et disponibilité des ressources).

## Activités et processus opérationnels :

- **Activité** : réalisation d'une tâche en transformant un état d'entrée en un état de sortie et ce, en utilisant du temps et des ressources.
- **Processus opérationnel** : ensemble partiellement ordonné d'activités dont l'exécution a pour but de contribuer à la réalisation d'un des objectifs de l'entreprise.

# Modélisation de l'activité : SADT/IDEF0

## SADT (Structured Analysis and Design Technique) :

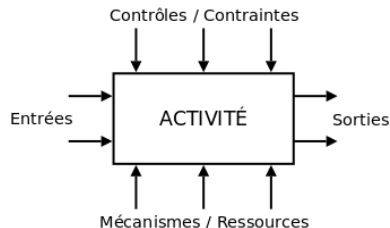
- Méthode d'analyse et de conception de système proposée par D.T. Ross en 1977.
- Approche systémique (système complexe = ensemble de systèmes simples en interaction).
- Largement utilisée dans l'industrie, excellent outil de communication.

## SADT/IDEF0 cherchent à répondre aux questions suivantes :

- Quelles fonctions sont mises en œuvre par le système ?
- Quels objets sont traités par les fonctions ?
- Quels mécanismes ou ressources sont nécessaires à l'exécution des fonctions ?

# L'activité dans SADT/IDEF0

Une activité peut être vue  
comme une fonction qui  
transforme des objets d'entrée  
en objets de sortie  
(décomposition possible).



## Sémantique associées aux entrées/sorties :

- **Entrées** : objets à traiter.
- **Entrées de contrôle** : informations contraignant l'exécution de l'activité.
- **Sorties** : objets traités.
- **Mécanismes** : moyens nécessaires à l'exécution de l'activité.

# Interconnectivité et décomposition fonctionnelle

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

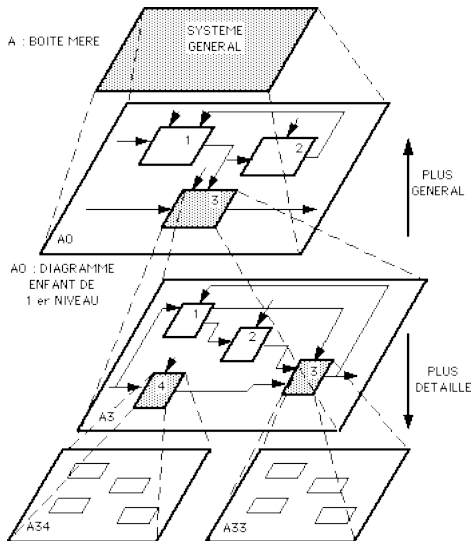
SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion



- **Interconnectivité :** flèches représentant flux de données, de matières, etc.
- Adoption d'une représentation **hiérarchique** et **modulaire**.
- Chaque activité décomposée contient au moins 3 et au plus 6 activités.

# Approche descendante/ascendante

## Introduction

### Présentation

L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion

## Approche descendante :

Décomposition à partir du niveau A0 du système en sous-systèmes jusqu'au niveau le plus bas.

## Approche ascendante :

Décrire les activités observées et les organiser en diagrammes SADT au niveau de détail le plus bas, puis aggrégation des activités en diagrammes jusqu'au niveau A0.

## Deux approches opposées :

- globalement équivalentes ;
- choix de l'approche en fonction du cas à résoudre :
  - peu d'informations disponibles : descendante ;
  - système dont les bases sont connues : ascendante.

# Construction d'un modèle SADT/IDEF0

## Étapes de construction :

- ❶ **Identifier les activités** : représenter les boîtes à placer sur la diagonale du diagramme.
- ❷ **Ajouter les entrées/sorties** : pour chaque activité, indiquer les entrées/sorties consommées ou produites.
- ❸ **Ajouter des contrôles, mécanismes et connectivités** : ajouter les ressources et les mécanismes pour connecter les interfaces d'activité entre elles.

# Exemple : processus d'intervention chirurgicale (1/3)

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

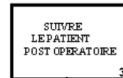
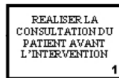
SADT/IDEFO  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion



# Exemple : processus d'intervention chirurgicale (2/3)

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

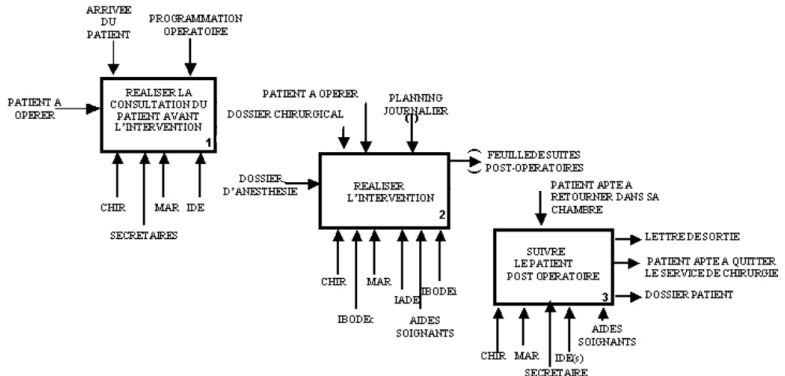
Outils de  
modélisation  
SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

Réseaux de  
Petri

Conclusion



# Exemple : processus d'intervention chirurgicale (3/3)

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

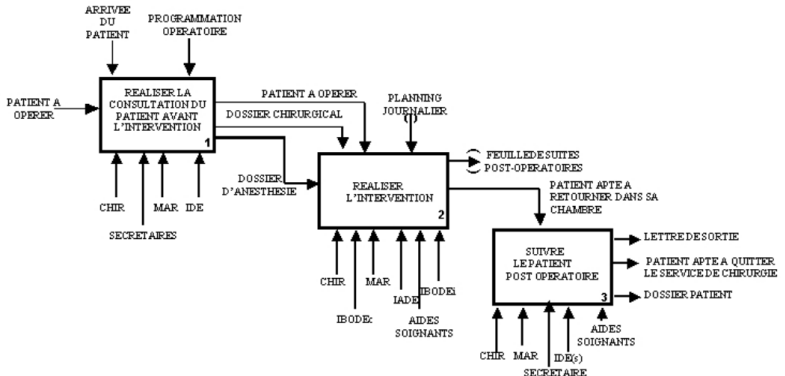
SADT/IDEF0  
IDEF3  
ARIS

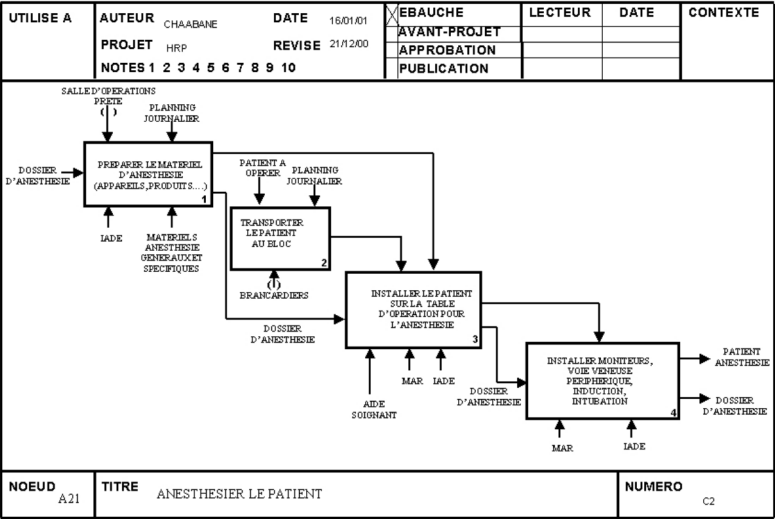
## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

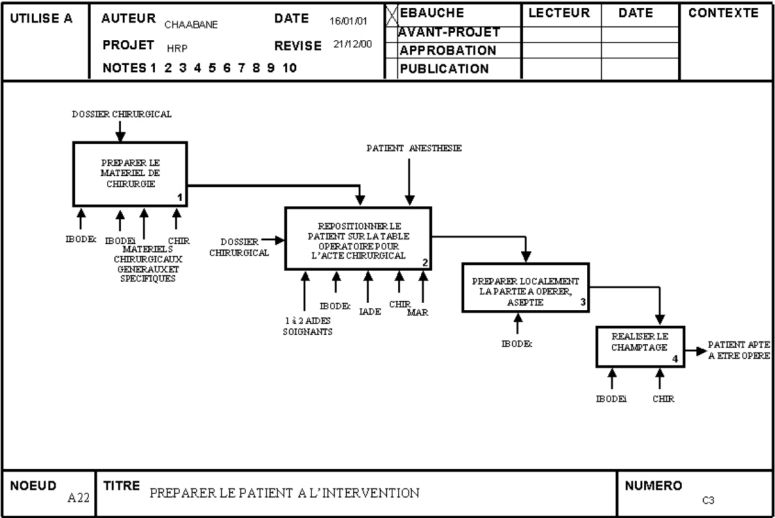
## Réseaux de Petri

## Conclusion

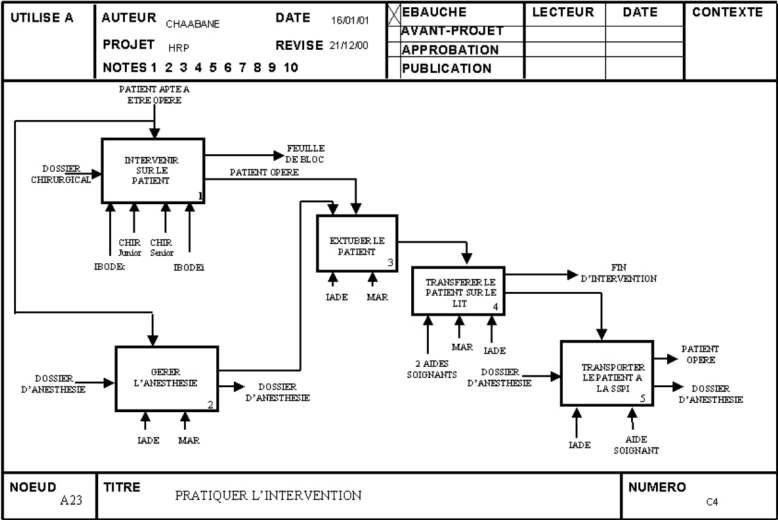




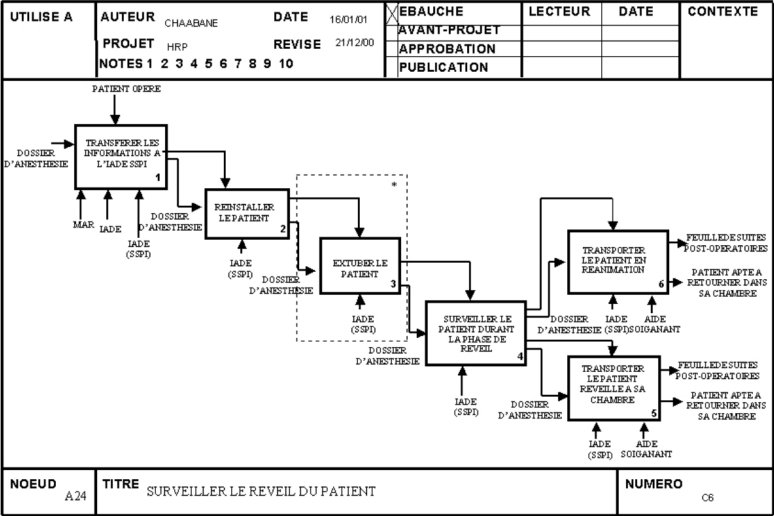
Source : Projet de recherche régional HRP2.



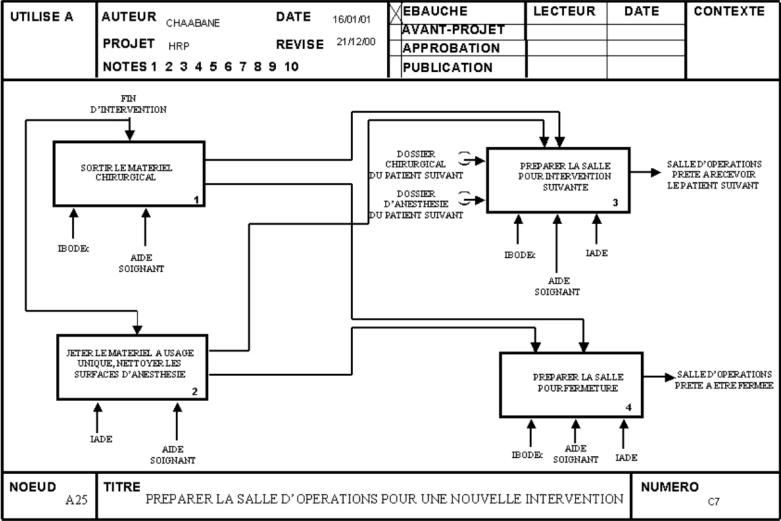
Source : Projet de recherche régional HRP2.



Source : Projet de recherche régional HRP2.



Source : Projet de recherche régional HRP2.



Source : Projet de recherche régional HRP2.

# Modélisation des processus opérationnels : IDEF3

## Apports d'IDEF3 par rapport à IDEF0 :

- Fonctionnalités supplémentaires en matière de modélisation du comportement de l'entreprise (flux de contrôle).
- Saisie et description des processus opérationnels d'entreprise.
- Notation graphique simple pour favoriser le dialogue entre les utilisateurs.
- Modélisation des processus sous forme d'un enchaînement d'étapes, appelées **unités de comportement** ⇒ **description de flux de contrôle du processus.**

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

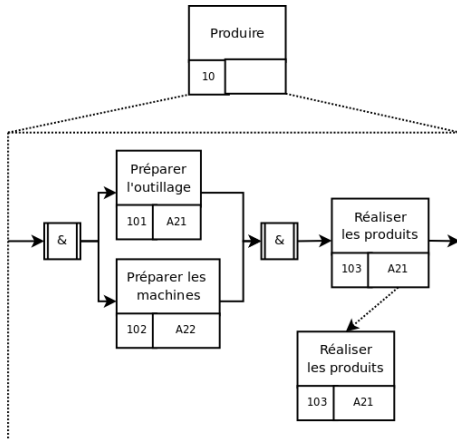
## Réseaux de Petri

## Conclusion

# Exemple

## Deux processus communicants :

- fabrication des produits ;
- maintenance.



⇒ Comportement asynchrone explicite dans IDEF3.

# Application de SADT aux systèmes hospitaliers

## Avantages/Inconvénients :

- + SADT propose une structure hiérarchisée par niveau permettant une représentation claire d'un système, aussi complexe soit-il.
- + Les diagrammes sont intemporels.
  - Absence d'opérations de logique booléenne (ET, OU, etc) dans IDEF0.
  - Les flux ne peuvent être différenciés en fonction de leur nature.
  - Aucune représentation dynamique n'est proposée.
  - Conversion en modèle de simulation difficile.

## Introduction

Présentation

L'approche POA

Architectures de  
référence

Vues

## Outils de modélisation

SADT/IDEF0

IDEF3

ARIS

## UML

Diagrammes de  
classes et  
d'objets

Diagrammes de  
cas d'utilisation

Diagrammes de  
séquence

Diagrammes  
d'activité

Diagrammes  
d'état

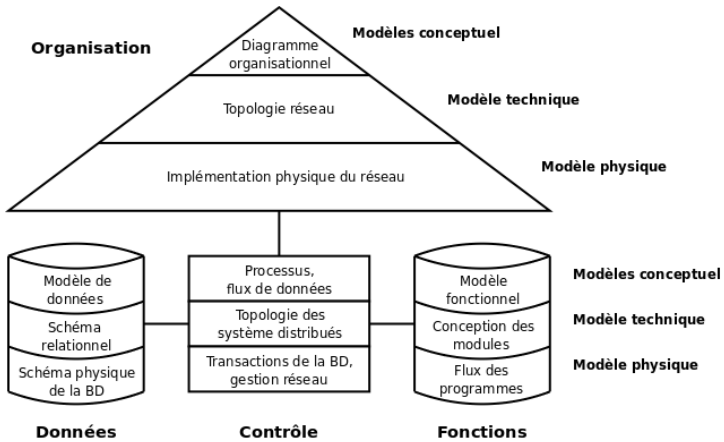
Applications

## Réseaux de Petri

## Conclusion

## ARIS (Architecture of Integrated Information Systems) :

- Cadre de modélisation.
- Approche multi-niveaux (conceptuel, technique, implémentation).
- Approche multi-vues (fonction, information, organisation, contrôle).
- Accent sur l'analyse et la définition des besoins durant la phase de conception de systèmes d'information et de gestion.



# ARIS : Event-driven process chain

## Les EPC font partie des outils de BPM :

- Les EPC permettent l'implémentation d'un ERP (Enterprise Resource Planning).
- Un EPC est un graphe orienté d'événements et de fonctions, permettant l'exécution alternative et parallèle de processus.
- Concept comparable aux diagrammes d'activité UML.

## Quelques éléments d'un EPC :

- **Événement** : pré-requis ou conséquence d'une fonction (passif).
- **Fonction** : tâche ou activité dans l'entreprise (actif).
- **Unité d'organisation** : responsable d'une fonction spécifique.

### Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

### Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

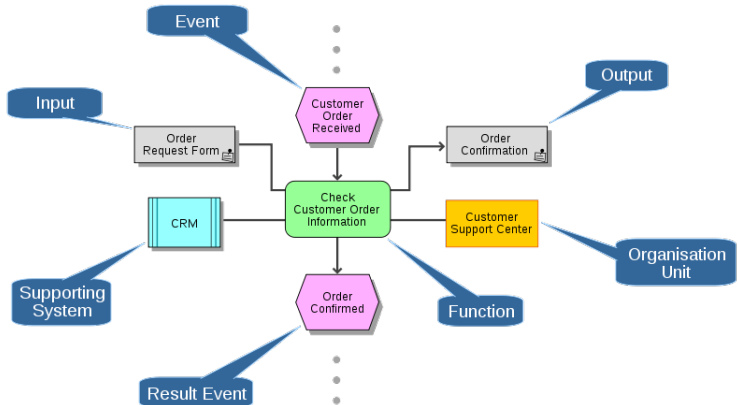
### UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

### Réseaux de Petri

### Conclusion

# ARIS : Event-driven process chain



## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion

# Exemple : modélisation d'une entreprise

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

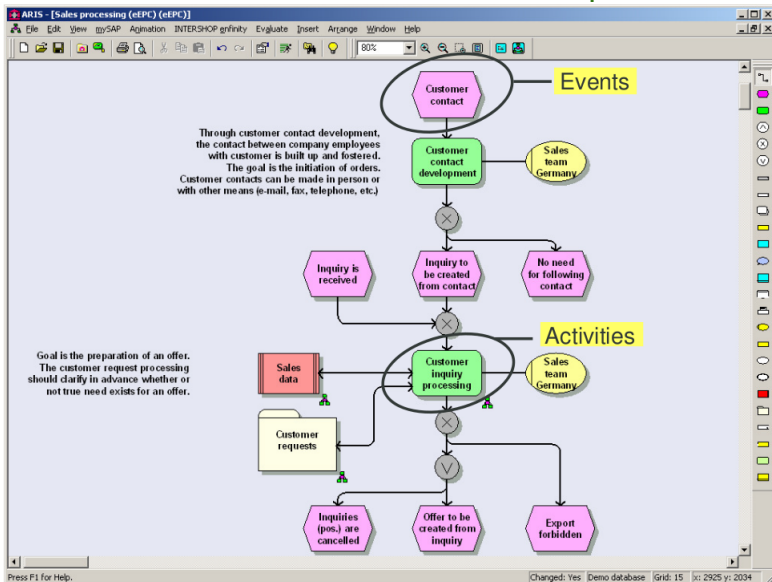
SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion



# Exemple : modélisation des urgences (extrait)

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

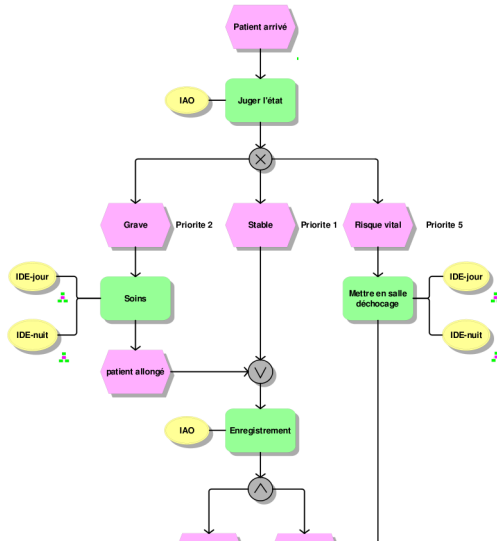
SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion



# Application d'ARIS aux systèmes hospitaliers

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion

## Avantages/Inconvénients :

- + ARIS offre une représentation graphique attrayante et intuitive.
- + La vue organisation structure le modèle et assure la cohérence entre les vues.
- + La représentation distincte d'activités, de données et de ressources dans un même modèle est possible.
- + L'intégration d'informations dynamiques est possible (durées opératoires par exemple).
- ARIS est un cadre de modélisation, aucune méthodologie n'est fournie.
- Les fonctionnalités de simulation sont restreintes.

Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

Outils de  
modélisation

SADT/IDEF0  
IDEF3  
ARIS

UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

Réseaux de  
Petri

Conclusion

1 Introduction

2 Outils de modélisation

3 UML

4 Réseaux de Petri

5 Conclusion

## Introduction

### Présentation

L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion

## UML (Uniform Modeling Language) :

- Langage de modélisation et de spécification orienté objet.
- Ensemble de notations graphiques standardisées regroupées en treize types de diagrammes.
- UML n'est pas une méthode : utilisation personnalisée.

## Décomposition en sous-ensembles :

- Les **vues** : organisationnel, dynamique, temporel, architectural, etc.
- Les **diagrammes** : description graphique du contenu des vues (notions abstraites).
- Les **modèles d'élément** : briques de base d'UML utilisées dans plusieurs types de diagramme.

## Introduction

Présentation

L'approche POA

Architectures de  
référence

Vues

## Outils de modélisation

SADT/IDEF0

IDEF3

ARIS

## UML

Diagrammes de  
classes et  
d'objets

Diagrammes de  
cas d'utilisation

Diagrammes de  
séquence

Diagrammes  
d'activité

Diagrammes  
d'état

Applications

## Réseaux de Petri

## Conclusion

## 9 diagrammes pour présenter le système :

- **Diagramme de classe** : les classes et leurs relations.
- **Diagramme d'états** : comportement d'une classe en terme d'états.
- **Diagramme d'activité** : comportement d'une opération en terme d'actions.
- **Diagramme de séquence** : représentation temporelle des objets et de leurs interactions.
- **Diagramme de cas** : fonctions du système du point de vue de l'utilisateur.

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion

## 9 diagrammes pour présenter le système :

- **Diagramme objet** : les objets et leurs relations.
- **Diagramme de composant** : composant physique de l'application.
- **Diagramme de déploiement** : déploiement des composants sur des dispositifs matériels.
- **Diagramme de collaboration** : représentation spatiale des objets, des liens et des interactions.

Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

Outils de  
modélisation

SADT/IDEF0  
IDEF3  
ARIS

UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

Réseaux de  
Petri

Conclusion

# Bases d'UML

## Fonctionnel

Diagrammes de use-case :

- activité
- séquence

## Statique

Diagrammes de classe :

- objet
- composant

## Dynamique

Diagrammes d'état :

- activité
- séquence

# Diagrammes de classes (1/4)

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets

Diagrammes de  
cas d'utilisation

Diagrammes de  
séquence

Diagrammes  
d'activité

Diagrammes  
d'état

Applications

## Réseaux de Petri

## Conclusion

## Une classe

- est un groupe d'objets possédant des états et un comportement communs.
- est un type de *classificateur*.
- peut représenter un concept concret (facture, automobile) ou abstrait (stratégie).

## Une classe possède

- des attributs : données primitives simples ou relations avec d'autres objets.
- des opérations : fonctionnalité offerte par une classe.

# Diagrammes de classes (2/4)

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets

Diagrammes de  
cas d'utilisation

Diagrammes de  
séquence

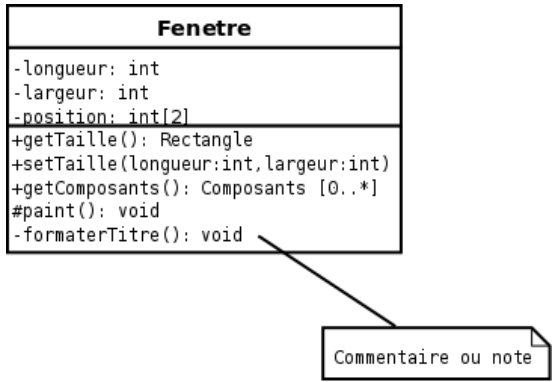
Diagrammes  
d'activité

Diagrammes  
d'état

Applications

Réseaux de  
Petri

Conclusion



# Diagrammes de classes (3/4)

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion

## Relations

- **Dépendance** : la classe *A* est utilisée brièvement par la classe *B*.
- **Association** : la classe *A* est en relation avec la classe *B* pour un certain temps.
- **Agrégation** : la classe *A* appartient à la classe *B*.
- **Composition** : la classe *A* fait partie de la classe *B*.
- **Généralisation (ou Héritage)** : la classe *A* est une classe *B*.

Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

Outils de  
modélisation

SADT/IDEF0  
IDEF3  
ARIS

UML

Diagrammes de  
classes et  
d'objets

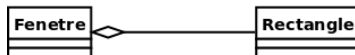
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

Réseaux de  
Petri

Conclusion

# Diagrammes de classes (4/4)

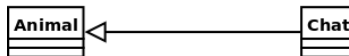
Fenetre « est propriétaire d'un » Rectangle



BarreDeTitre « fait partie d'une » Fenetre



La classe Chat hérite de la classe Animal



## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets

Diagrammes de  
cas d'utilisation

Diagrammes de  
séquence

Diagrammes  
d'activité

Diagrammes  
d'état

Applications

## Réseaux de Petri

## Conclusion

## Dessiner les diagrammes de classes correspondant aux situations suivantes :

- ① Un polygone est constitué de points. Un point possède une abscisse et une ordonnée.
- ② Un joueur de rugby est un avant, un demi ou un arrière.
- ③ Une voiture possède 4 roues et 1 moteur, qui peut être diesel ou essence.
- ④ Une galerie expose des œuvres, faites par des créateurs, et représentant des thèmes. Des clients, accueillis par la galerie, achètent des œuvres.
- ⑤ L'option Ingénierie et Santé possède 3 profils. Chaque profil compte entre 1 et 10 élève. Chaque élève peut être un AST.

# Diagrammes objet

## Les diagrammes objet :

- permettent d'obtenir une structure statique du système ;
- sont utiles pour préciser un contexte (avant ou après une interaction) ;
- sont utiles pour préciser la structure des classes (i.e. illustrer la récursivité).

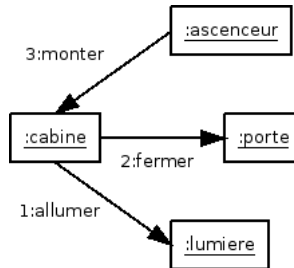
nom de l'objet

nom de l'objet : classe

# Diagramme de collaboration

## Les diagrammes de collaboration :

- permettent de décrire les interactions entre objets ;
- sont une extension du diagramme d'objets en insistant sur leurs interactions et en explicitant l'ordre d'envoi des messages.



# Diagrammes de cas d'utilisation

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
**Diagrammes de  
cas d'utilisation**  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion

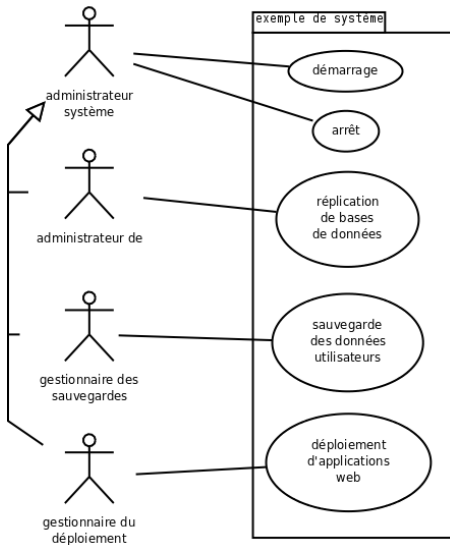
## Un cas d'utilisation

- représente un élément fonctionnel bien identifié dans un système.
- permet d'identifier les conditions nécessaires à leur bon fonctionnement.
- Par exemple : « Visualisation d'un compte-rendu ».

## Un acteur

- « déclenche » un cas d'utilisation.
- est un humain ou un système externe au cas d'utilisation.
- Par exemple : horloge système qui déclenche un cas d'utilisation à un instant donné.

## Exemple de cas d'utilisation



### Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

### Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

### UML

Diagrammes de  
classes et  
d'objets

### Diagrammes de cas d'utilisation

Diagrammes de  
séquence

Diagrammes  
d'activité

Diagrammes  
d'état

Applications

### Réseaux de Petri

### Conclusion

# Liens entre cas d'utilisation

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

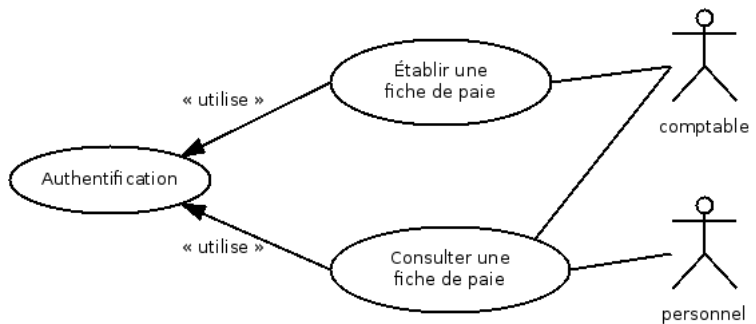
## UML

Diagrammes de  
classes et  
d'objets  
**Diagrammes de  
cas d'utilisation**  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

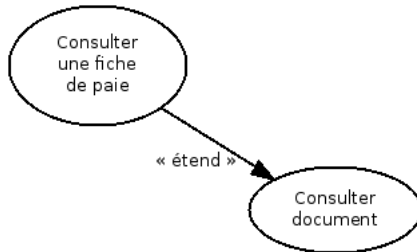
## Conclusion

Utilisation : implique une relation entre cas.

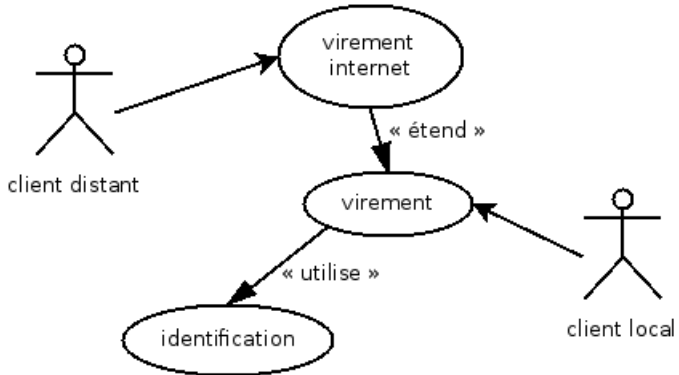


# Liens entre cas d'utilisation

Extension : implique un héritage entre cas.



## Exemple de cas d'utilisation



# Multiplicité

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets

### Diagrammes de cas d'utilisation

Diagrammes de  
séquence

Diagrammes  
d'activité

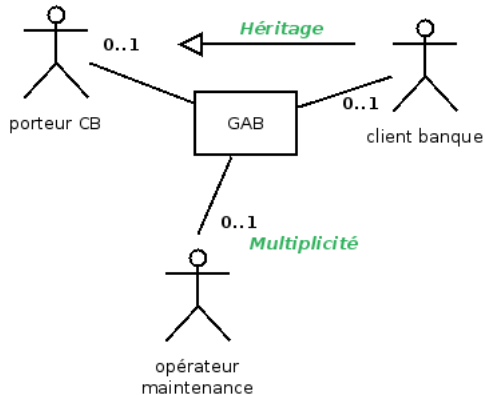
Diagrammes  
d'état

Applications

Réseaux de  
Petri

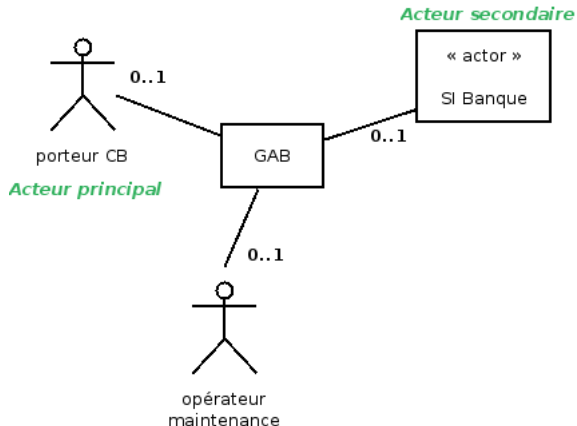
Conclusion

- Spécification du nombre d'utilisateur concernés.
- Possibilité de spécifier les acteurs.



## Acteurs principal/secondaire

- L'acteur principal produit un résultat **observable**.
- L'acteur secondaire est sollicité par le système pour des informations secondaires.



# Cas d'utilisation

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
**Diagrammes de  
cas d'utilisation**  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion

Une fois les cas identifiés, il faut les décrire :

- Description textuelle de toutes les interactions entre les acteurs.
- Début et fin clairement identifiés.
- Variantes possibles :
  - cas nominaux ;
  - alternatives ;
  - cas d'erreur...
- Spécification des préconditions, enchaînements :
  - besoins interfaces homme-machine ;
  - contraintes non fonctionnelles (fréquence, disponibilité, fiabilité, performance)...

# Démarche générale

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

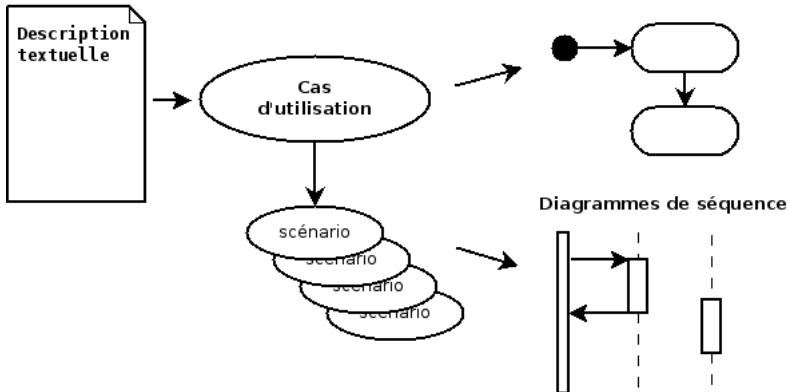
SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion



## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion

## Réservation de matériel

Dans un établissement scolaire, on désire gérer la réservation des salles de cours ainsi que du matériel pédagogique (ordinateur portable ou/et vidéo-projecteur). Seuls les enseignants sont habilités à effectuer des réservations (sous réserve de disponibilité de la salle ou du matériel). Le planning des salles peut quant à lui être consulté par tout le monde (enseignants et étudiants). Par contre, le récapitulatif horaire par enseignant (calculé à partir du planning des salles) ne peut être consulté que par les enseignants. Enfin, il existe pour chaque formation un enseignant responsable qui seul peut éditer le récapitulatif horaire pour l'ensemble de la formation.

Modéliser cette situation par un diagramme de cas d'utilisation.  
Quelles sont les classes à prévoir ?

# Diagrammes de séquence

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
**Diagrammes de  
séquence**  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion

## Objectif :

Illustrer les données échangées entre utilisateurs et composants dans un exemple particulier d'utilisation du système.

## Corollaire :

- Un diagramme de séquence est un scénario.
- Ce n'est pas une description exhaustive du comportement du système.

## Conséquences :

- Toujours donner un titre à diagramme de séquence.
- Toujours rattacher un diagramme de séquence à un cas d'utilisation.

# Composants d'un diagramme de séquence

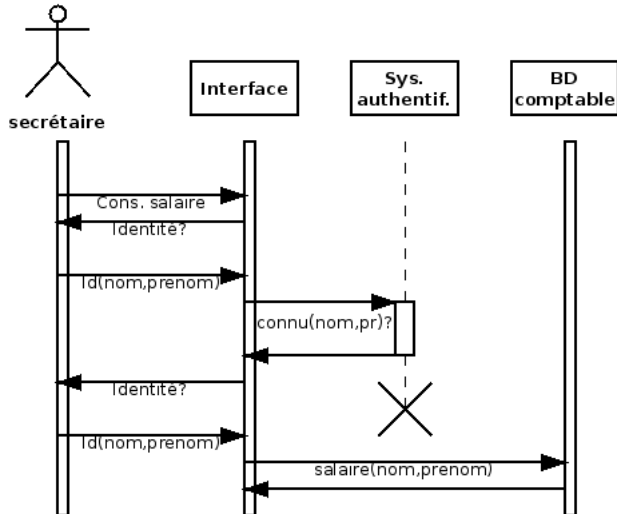
Un diagramme de séquence est composé :

- d'un axe vertical non gradué (implicite), le temps ;
- des objets et utilisateurs, un par axe ;
- des messages (flèches entre objets/utilisateurs) ;
- des étiquettes.

Un axe objet/utilisateur est composé :

- d'un début : quoi, existence.
- d'une fin : une croix si l'objet meurt.
- d'un tracé : épais si actif, pointillé si inactif.

# Exemple de séquence



# Messages dans un diagramme de séquence

## Un acteur/objet actif :

- effectue une opération ;
- attend le retour d'un voie de message en mode synchrone.

## Un message modélise le transit de données entre objets/acteurs :

- Un message est représenté par une flèche surmontée d'un nom.
- Plusieurs types de messages sont identifiés :
  - signaux : pas de paramètres ;
  - envoi de données : paramètres, pas de résultat ;
  - demande de données : résultat, éventuellement paramètre.

# Messages dans un diagramme de séquence

## Messages synchrones/asynchrones :

- Synchrone : l'émetteur est bloqué et attend que l'appelé ait fini de traiter le message. Dans ce cas le retour peut être implicite. →
- Asynchrone : l'émetteur n'est pas bloqué et peut continuer son exécution. ↘

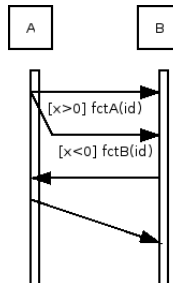
## Messages instantanés/lents :

- Instantané : →
- Lent : ↘

# Messages dans un diagramme de séquence

## Représentation des messages :

- Les paramètres d'entrée/sortie sont indiqués : nom(p1,p2).
- Messages conditionnels : la condition entre crochets doit être vraie pour que le message soit envoyé.



# Diagrammes d'activité

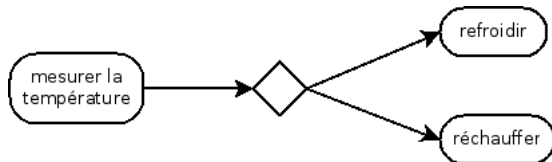
## Un diagramme d'activité

- décrit l'exécution d'un système et permet d'identifier ses comportements successifs sans s'intéresser à sa structure :
  - déterminer tous les branchements conditionnels possibles ;
  - déterminer toutes les boucles possibles.
- Une **activité** modélise la réalisation d'une étable complexe et décomposable, elle peut être interrompue par un événement.
- Une **action** est une étape simple, atomique, qui ne peut être interrompue.

# Diagrammes d'activité

## Représentation de l'aspect dynamique :

- organisé par rapport aux activités ;
- destiné à représenter le comportement interne ;
- représente l'état de l'exécution d'un mécanisme sous la forme d'un déroulement d'étapes regroupées séquentiellement dans les branches parallèles d'un flot de contrôle.



# Exemple de diagrammes d'activité

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

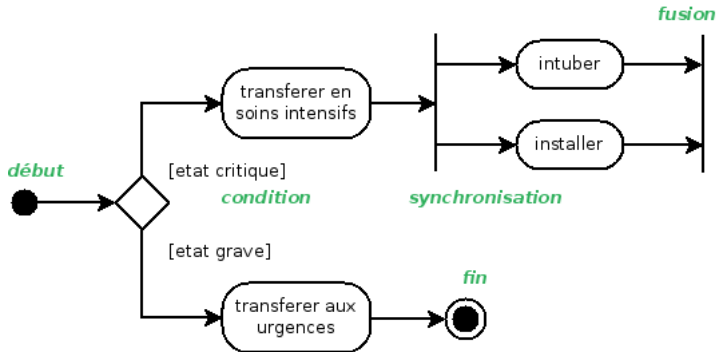
SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
**Diagrammes  
d'activité**  
Diagrammes  
d'état  
Applications

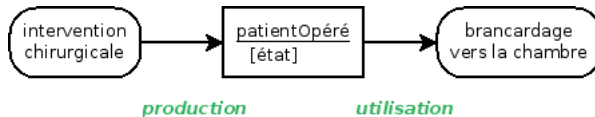
## Réseaux de Petri

## Conclusion



# Activités et objets

Représentation des objets utilisés en entrée ou produits en sortie.



# Diagrammes d'état

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion

Un diagramme d'état ou une machine d'état permet de décrire l'aspect dynamique des objets.

## Un état d'un objet

- se définit à un instant donné par les valeurs de ses propriétés ;
- modélise un moment spécifique du comportement du système.
- Seuls certains états caractéristiques sont étudiés (disjoints ou impriqués).
- Exemple pour l'objet « employé » : recruté, en activité, en congé, malade.

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
**Diagrammes  
d'état**  
Applications

## Réseaux de Petri

## Conclusion

## Une transition ou un événement

- une relation, ou chemin, entre deux états ;
- est un fait survenu qui fait passer un objet d'un état à un autre ;
- est considéré comme instantané.
- Deux événements peuvent être liés par des relations de causalité.
- Deux événements sans lien de causalité sont dits concurrents.
- Un événement est aussi une voie de transmission d'information d'un objet vers un autre.

# Exemple de diagramme d'état

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

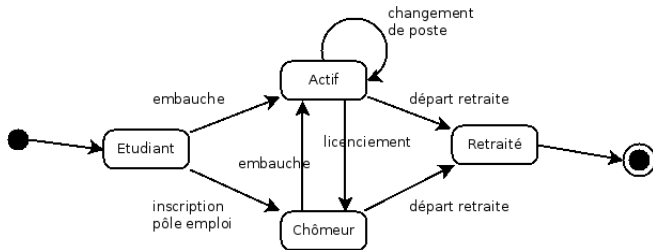
Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité

## Diagrammes d'état

Applications

## Réseaux de Petri

## Conclusion



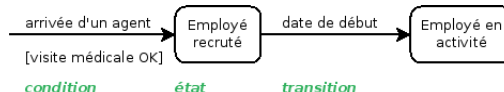
# Concepts du modèle dynamique

## Action

- Une action est une opération instantanée qui ne peut être interrompue, elle est associée à un événement.
- Certaines actions peuvent être associées à un état : elles surviennent avec un événement qui ne fait pas changer l'objet d'état.

## Activité

- Une activité est une opération d'une certaine durée qui peut être interrompue, elle est associée à l'état d'un objet.
- Une activité est interrompue si un événement provoque le changement d'état d'un objet.



## Introduction

Présentation

L'approche POA

Architectures de

référence

Vues

## Outils de modélisation

SADT/IDEF0

IDEF3

ARIS

## UML

Diagrammes de  
classes et  
d'objets

Diagrammes de  
cas d'utilisation

Diagrammes de  
séquence

Diagrammes  
d'activité

**Diagrammes  
d'état**

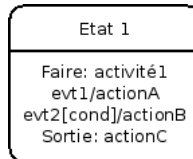
Applications

## Réseaux de Petri

## Conclusion

## Modélisation d'événements à l'intérieur d'un état :

- entry (entrée) : dès que l'on rentre dans l'état.
- exit (sortie) : dès que l'on quitte l'état.
- do (faire) : tant que l'on est dans l'état.



## Exercice

### Représenter le comportement du Tamaguchi :

Un Tamaguchi en état normal n'a pas faim pendant un certain temps (temps d'autonomie). Au bout de ce temps, le Tamaguchi a faim et il pleure. Pour lui donner à manger, l'utilisateur du Tamaguchi le met à table et le Tamaguchi s'arrête de pleurer. Un Tamaguchi mange pendant un certain temps (temps de restauration). Au bout de ce temps, il se remet à pleurer. Il pleure jusqu'à ce que l'utilisateur le sorte de table. Quand il sort de table, le Tamaguchi revient dans l'état normal... et ainsi de suite tant que le Tamaguchi ne meurt pas. Si le Tamaguchi pleure plus de 5 minutes d'affilée, il meurt.

On suppose que les événements émis par le Tamaguchi vers l'utilisateur sont : « avoir faim », « ne plus avoir faim », « mourir » et que les événements émis par l'utilisateur et reçus par le Tamaguchi sont « être mis à table » et « sortir de table ».

# Application d'UML aux systèmes hospitaliers (1/5)

## Staccini et al. (2001)

- Création d'un modèle de données dédié à plusieurs systèmes hospitaliers.
- Proposition d'une méthodologie pour la structuration des besoins des patients.
- Application au processus de transfusion sanguine.

## Vasilakis et Kuramoto (2005)

- Utilisation de diagrammes d'états pour modéliser l'activité de trois chirurgiens.
- Modélisation d'activités en parallèle.
- Déduction d'un modèle de simulation.
- **Presentation du problème.**

# Application d'UML aux systèmes hospitaliers (2/5)

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état

## Applications

Réseaux de  
Petri

## Conclusion

## Modélisation multi-vues :

- Utilisation de plusieurs types de diagrammes pour la représentation de plusieurs vues.
- Dualité vue processus/vue ressource.
- Mise en forme de modèles plus clairs et personnalisés.

## Application à la distribution des médicaments :

- Modèle centré sur le médicament :
  - l'armoire à pharmacie est amenée dans la pharmacie ;
  - contrôle et réapprovisionnement ;
  - l'armoire à pharmacie est ramenée dans le service.
- Modèle centré sur le transporteur : mise en place de tournées selon le type de véhicule.

# Application d'UML aux systèmes hospitaliers (3/5)

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

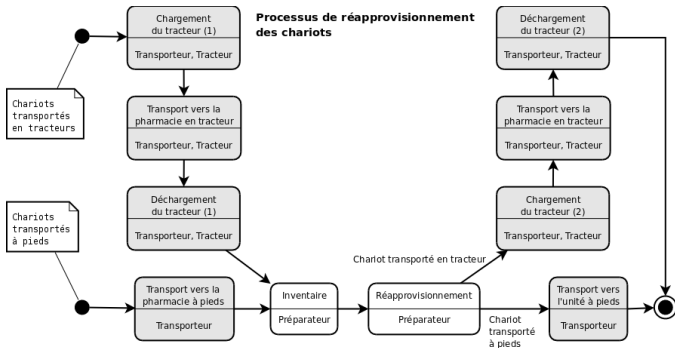
## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état

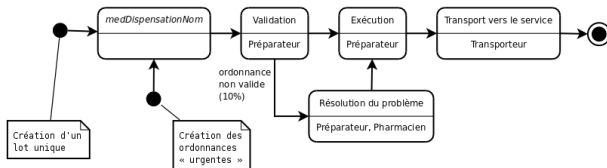
## Applications

Réseaux de  
Petri

Conclusion



## Processus de dispensation nominative



# Application d'UML aux systèmes hospitaliers (4/5)

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

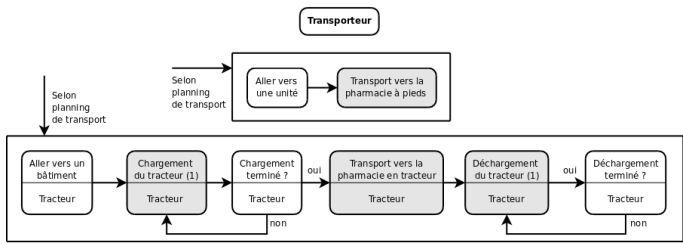
## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état

## Applications

Réseaux de  
Petri

## Conclusion



# Application d'UML aux systèmes hospitaliers (5/5)

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état

## Applications

## Réseaux de Petri

## Conclusion

## Avantages/Inconvénients :

- + UML est un langage formel et normalisé, offrant précision et stabilité.
- + UML est un support de communication performant, permettant de cadrer l'analyse ; la compréhension de représentations abstraites complexes est facilitée et son caractère polyvalent et sa souplesse en font un langage universel.
- La mise en pratique d'UML est dangereuse car la modélisation est totalement libre et les outils sont variés.
- UML doit être spécifié pour être utilisé dans un cadre précis sans risque d'erreur.

Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

Outils de  
modélisation

SADT/IDEF0  
IDEF3  
ARIS

UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

Réseaux de  
Petri

Conclusion

1 Introduction

2 Outils de modélisation

3 UML

4 Réseaux de Petri

5 Conclusion

# Réseaux de Petri

## Un réseau de Petri est :

- Un modèle mathématique permettant la représentation de systèmes distribués discrets.
- Un langage de modélisation représenté sous forme d'un graphe biparti orienté.
- Introduit par Petri en 1962.

## Objectif des réseaux de Petri :

- Étude de systèmes discrets dont les champs d'application sont les systèmes de production et informatique.
- Exemple : prévoir la production d'une industrie manufacturière qui fabrique de nombreux produits différents en petites séries (variation des besoins, flexibilité).

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion

## Systèmes manufacturiers flexibles (FMS) :

- Ensemble de machines flexibles (multi-produits).
- Système de transfert automatique (multi-ressources).
- Système de prise de décision (ordonnanceur).

## Un système complexe à mettre en œuvre :

- Valider les spécifications.
- Prévoir les performances.
- Optimiser le fonctionnement.

Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

Outils de  
modélisation

SADT/IDEF0  
IDEF3  
ARIS

UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

Réseaux de  
Petri

Conclusion

1 Introduction

2 Outils de modélisation

3 UML

4 Réseaux de Petri

5 Conclusion

# Conclusion

## Conclusion sur l'analyse des aspects fonctionnels

- Décomposition fonctionnelle : principe fondamental en modélisation fonctionnelle.
- Approche d'analyse verticale descendante.
- Modélisation processus opérationnels : approche horizontale (analyse des flux de contrôle/d'objets).

## Quel outil pour quelle application ?

- SADT : modélisation de systèmes du point de vue de ses fonctions sans activités concurrentes.
- ARIS : cadre de modélisation bien défini pour la modélisation d'activités, d'informations et de ressources.
- UML : potentiel important, mais utilisation risquée.

# Conclusion

## Quatre aspects fondamentaux :

- Vue fonction.
- Vue information.
- Vue ressource.
- Vue organisation.

## Modélise et analyse de ces aspects essentiels pour :

- La conception de nouveaux systèmes d'entreprise.
- La réorganisation ou l'amélioration du fonctionnement de tout ou partie d'une entreprise.
- La mise en œuvre d'un programme d'intégration d'entreprise.

# Et les systèmes hospitaliers ?

Un hôpital n'est pas une entreprise comme les autres :

- Observation et modélisation de flux de patients, et non de produits (comportement des entités).
- Influence du dossier médical sur le parcours patient (soins, pathologie).
- Intervention d'un grand nombre de ressources humaines très variées (du manutentionnaire au chirurgien).
- Un service de soin = Une organisation spécifique.
- Coordination précise des ressources humaines et matérielles.
- Les services de soins d'un même hôpital sont hautement cloisonnés.
- Notion de pilotage du système essentielle (réaction aux aléas).

## Introduction

Présentation  
L'approche POA  
Architectures de  
référence  
Vues

## Outils de modélisation

SADT/IDEF0  
IDEF3  
ARIS

## UML

Diagrammes de  
classes et  
d'objets  
Diagrammes de  
cas d'utilisation  
Diagrammes de  
séquence  
Diagrammes  
d'activité  
Diagrammes  
d'état  
Applications

## Réseaux de Petri

## Conclusion

# Et les systèmes hospitaliers ?

## Quelques conseils et remarques :

- La phase de collecte de données sera plus longue que dans n'importe quelle entreprise.
- Les outils de modélisation doivent être simples à comprendre et personnalisés.
- Les concepts doivent être aménagés pour la modélisation de mécanismes particuliers.
- L'intégration d'un système de pilotage est inévitable pour obtenir un modèle fiable.

# Références

- F. Vernadat, *Techniques de Modélisation en Entreprise : Applications aux Processus Opérationnels*, Economica, Paris, 1999.
- D. Pilone et N. Pitman, *UML 2 en concentré*, O'Reilly, Paris, 2006.
- L. Trilling, B. Besombes, S. Chaabane et A. Guinet, *Investigation et comparaison des méthodes et outils d'analyse pour l'étude des systèmes hospitaliers*, Rapport de recherche sur le projet HRP2, 2004.
- C. Vasilakis et L. Kuramoto, *Comparing two methods of scheduling outpatient clinic appointments using simulation experiments*, Clinical & Investigative Medicine, 28 :368-370, 2005.
- P. Staccini, M. Joubert, J. M. Quaranta, D. Fieschi et M. Fieschi, *Modeling health care processes for eliciting user requirements : a way to link quality paradigm and clinical information system design.*, International Journal of Medical Informatics, 64 :129-142, 2001.
- H. Hess, *From Event-driven modeling to Process monitoring*, Event Processing Symposium, 14-16 March 2006.