UPEC
UNIVERSITÉ PARIS-EST CRÉTEIL
EPISEN

Introduction

Ingéniérie des systèmes d'information

Paul Brunet

Introduction
Plan

1. Présentation du cours

2. Systèmes d'information et modèles
   - 2.1. Systèmes d'information
   - 2.2. Modèles
   - 2.3. Exemple

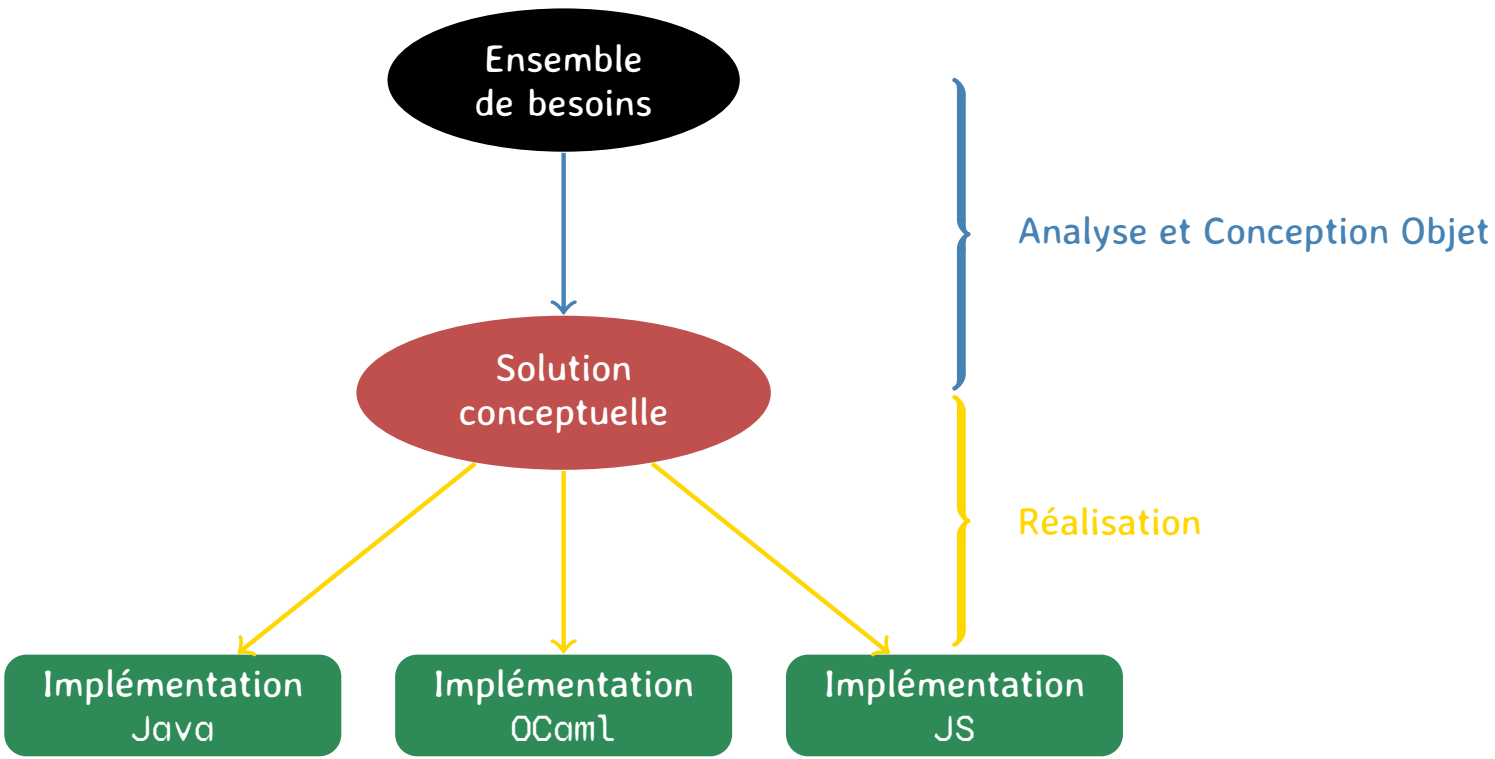
3. Analyse et Conception Orientées Objet
   - 3.1. Origine
   - 3.2. Principes
   - 3.3. Difficultés

4. Unified Modeling Language

Le sujet

définition

L'Ingénierie de Système d'Information vise à transformer les besoins des utilisateurs en spécifications formalisées d'une future application.



En bref

UPEC UNIVERSITÉ PARIS-EST CRÉTEIL | EPISEN

☞ Pré-requis
- Connaissances générales en S.I.
- Savoir-faire en analyse et conception
- Savoir-faire en programmation orientée objet

☞ Objectifs
- Construire des modèles d'analyse et de conception selon une approche objet
- Utiliser ces modèles méthodiquement pour réaliser une application orientée objet

☞ Contenu
1) S.I. & Ingénierie de S.I.
2) Modélisation d'un système d'information
   - 2.1 La vue fonctionnelle
   - 2.2 La vue structurelle
   - 2.3 La vue dynamique
3) Intégration des trois vues
4) Patrons de conception

La pédagogie

- Exposé des connaissances de base
- Exercices d'application
- Projet logiciel
- Contrôle continu (1/3 de la note) :
  - DM analyse fonctionnelle
  - Réalisation de modèles
  - Production logicielle
- Examen final (2/3 de la note)

Bibliographie

- Le guide de l'utilisateur UML
  Grady Booch, Ivar Jacobson, James Rumbaugh, Eyrolles,
- UML 2 par la pratique
  Pascal Roques, Eyrolles
- De UML à SQL
  Christian Soutou, Eyrolles
- UML2 : Modéliser une application WEB
  Pascal Roques, Eyrolles
- UML2 et les design patterns
  Craig Larman, Pearson Education
- UML2 : Principe de modélisation
  B. Charroux, A. Osmani, Y. Thierry-Mieg, Eyrolles

Introduction
Plan

1. Présentation du cours

2. Systèmes d'information et modèles
   - 2.1. Systèmes d'information
   - 2.2. Modèles
   - 2.3. Exemple

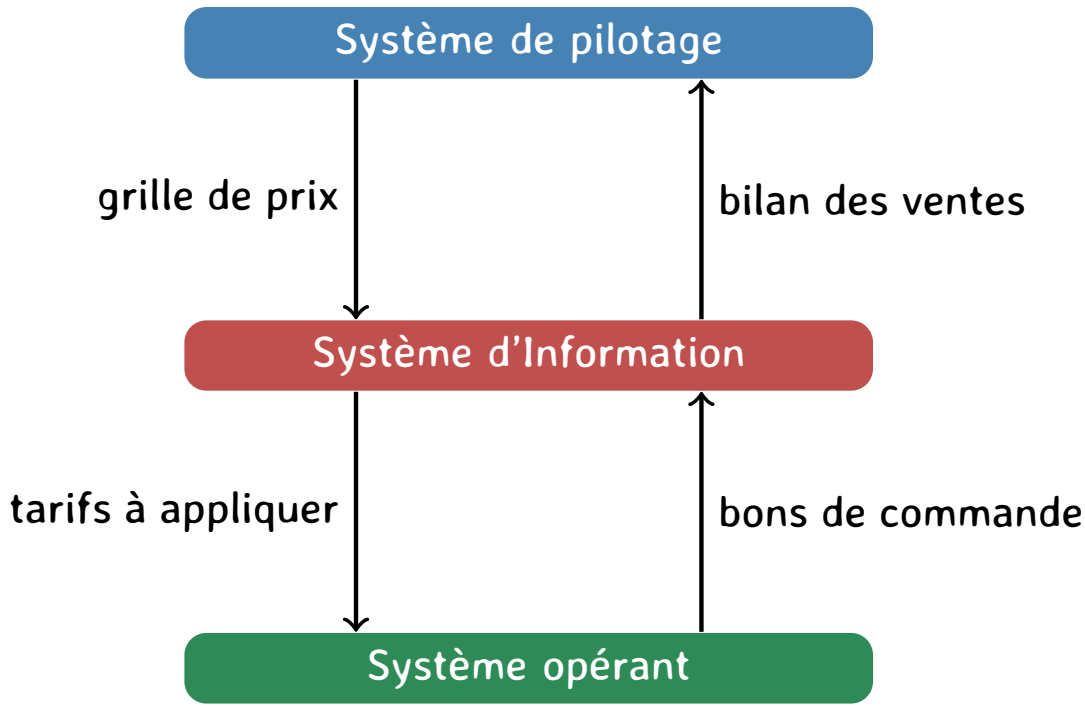
3. Analyse et Conception Orientées Objet
   - 3.1. Origine
   - 3.2. Principes
   - 3.3. Difficultés

4. Unified Modeling Language

Les Systèmes d'Informations (S.I.)

définition

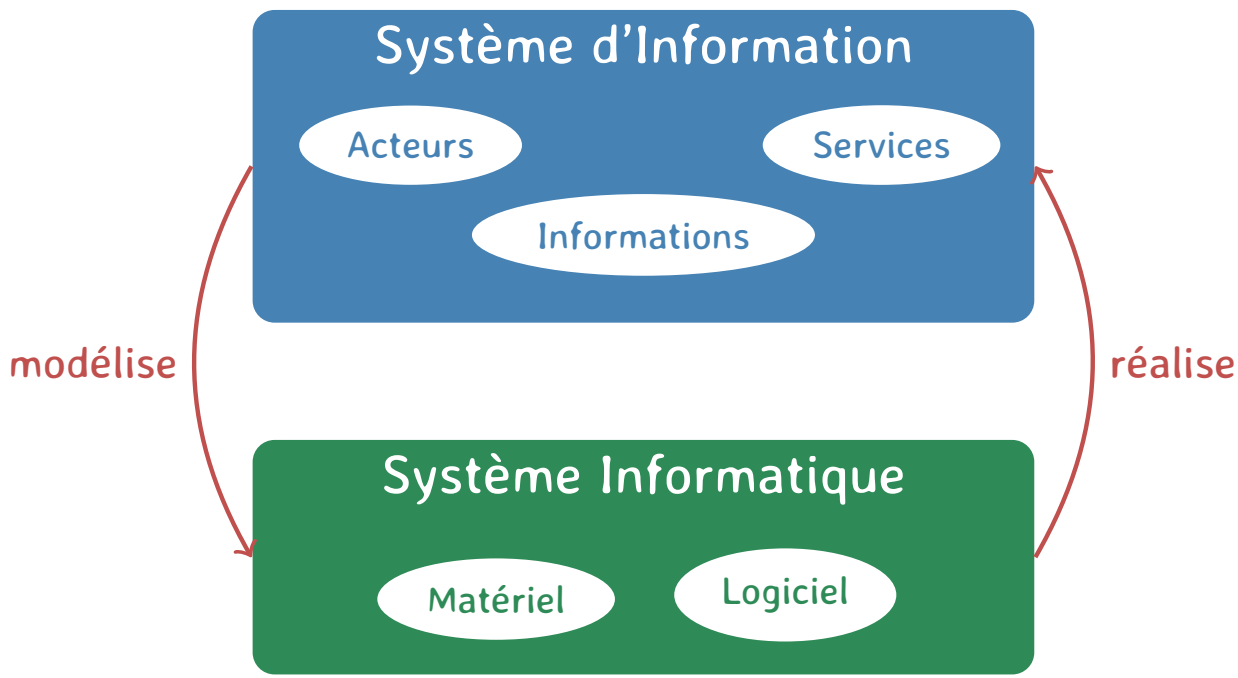
Un S.I. est un ensemble de ressources *(humaines, organisationnelles, matérielles et logicielles)* permettant de gérer *(saisir, stocker, traiter, restituer)* les informations utiles aux décideurs et opérationnels d'une organisation.



S.I. versus S.I.A.

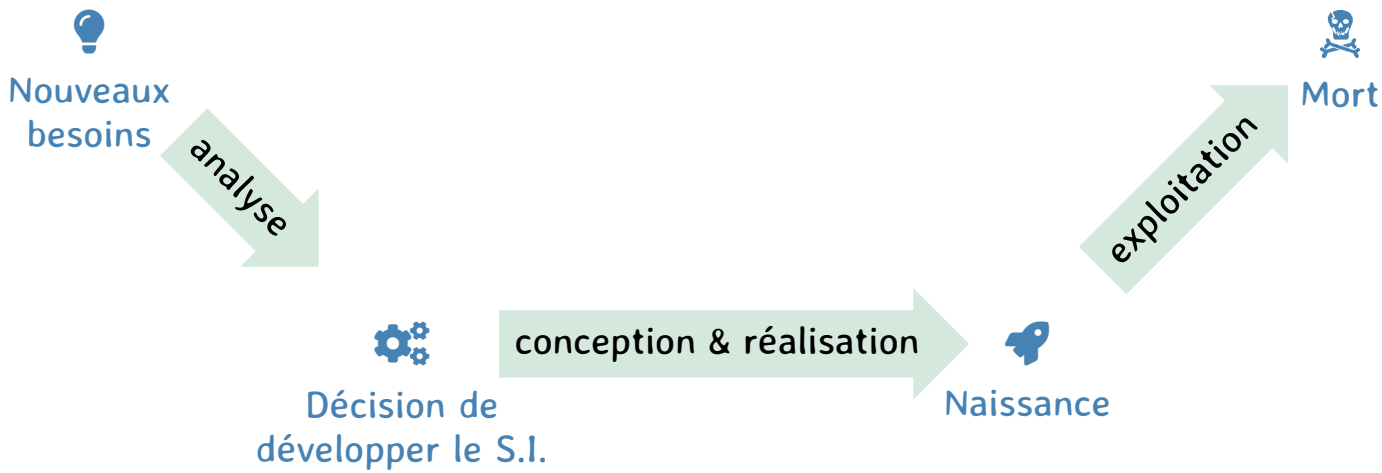
Ne pas confondre

- Système d'information : modèle recouvrant tous les moyens (logiciels, matériels, humains, organisationnels) permettant de répondre aux besoins
- Système d'information automatisé (S.I.A., aussi appelé Système informatique) : solution technique répondant aux besoins



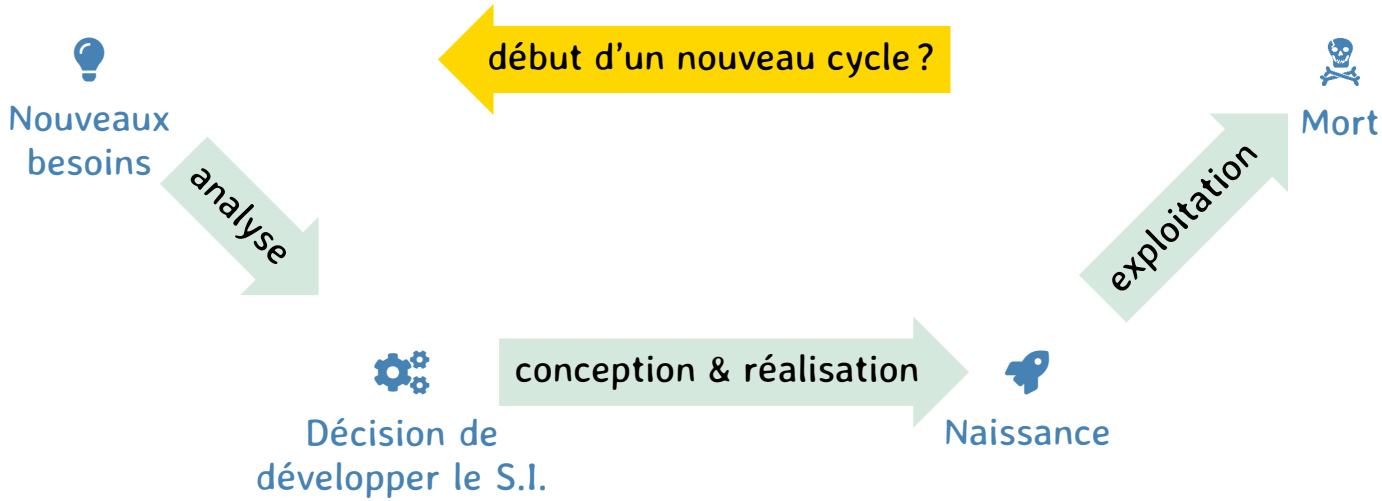
Le système d'information automatisé

Au sein de l'organisation, le S.I.A. a son propre cycle de vie, qui commence par l'apparition de besoins.



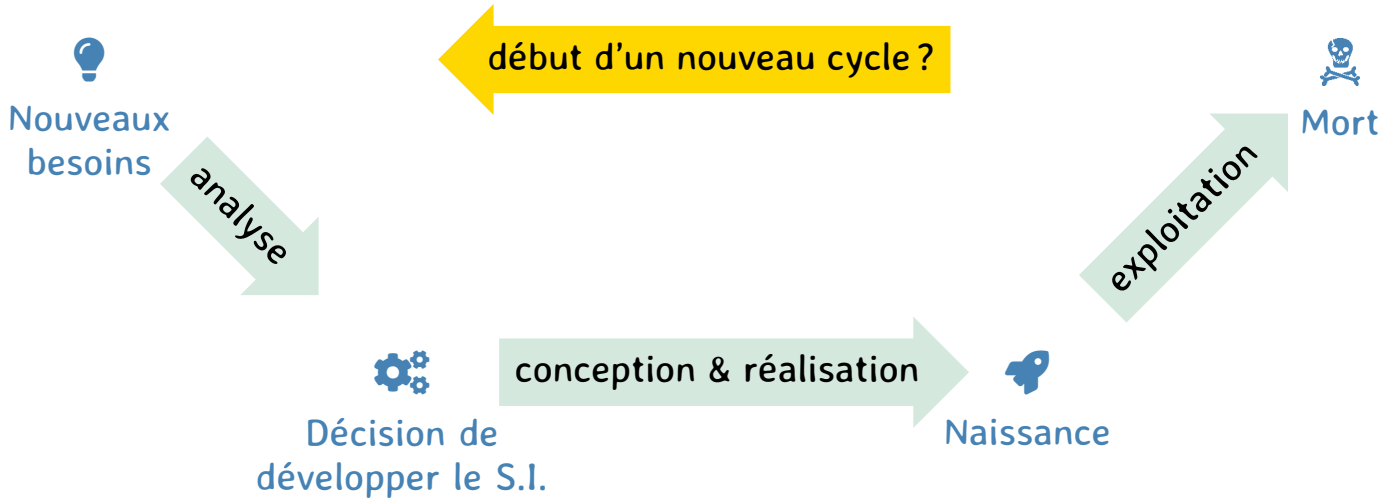
Le système d'information automatisé

Au sein de l'organisation, le S.I.A. a son propre cycle de vie, qui commence par l'apparition de besoins.



Le système d'information automatisé

Au sein de l'organisation, le S.I.A. a son propre cycle de vie, qui commence par l'apparition de besoins.



- Les besoins correspondent à de nouveaux objectifs.
- Les besoins résultent de problèmes, de disfonctionnements.
- Les besoins peuvent venir d'exemples, de démonstrations de produits.
- Les besoins sont associés à des innovations techniques.
- Les besoins proviennent de changements de règlementations, de lois...

- Le S.I. d'une entreprise en est un composant clé : son bon fonctionnement est souvent critique aux opérations de celle-ci.
- Ses responsabilités sont multiples, et il est parfois dépoyé sur des appareils **nombreux** et **hétérogènes**.
- En conséquences, les S.I. modernes sont

  gros, sensibles, et compliqués.

- Le S.I. d'une entreprise en est un composant clé : son bon fonctionnement est souvent critique aux opérations de celle-ci.
- Ses responsabilités sont multiples, et il est parfois dépoyé sur des appareils nombreux et hétérogènes.
- En conséquences, les S.I. modernes sont gros, sensibles, et compliqués.

beaucoup de code,
beaucoup de matériel,
beaucoup d'usagers,...

- Le S.I. d'une entreprise en est un composant clé : son bon fonctionnement est souvent critique aux opérations de celle-ci.
- Ses responsabilités sont multiples, et il est parfois dépoyé sur des appareils **nombreux** et **hétérogènes**.
- En conséquences, les S.I. modernes sont

gros, sensibles, et compliqués.

beaucoup de code, beaucoup de matériel, beaucoup d'usagers,...

opérations, décisions, sécurité,...

- Le S.I. d'une entreprise en est un composant clé : son bon fonctionnement est souvent critique aux opérations de celle-ci.
- Ses responsabilités sont multiples, et il est parfois dépoyé sur des appareils **nombreux** et **hétérogènes**.
- En conséquences, les S.I. modernes sont

gros, sensibles, et compliqués.

beaucoup de code, beaucoup de matériel, beaucoup d'usagers,...

opérations, décisions, sécurité,...

langages de programmation multiples, technologies multiples, enjeux métiers multiples,...

Méthodes d'analyse et conception de S.I.

- Le S.I. d'une entreprise en est un composant clé : son bon fonctionnement est souvent critique aux opérations de celle-ci.
- Ses responsabilités sont multiples, et il est parfois dépoyé sur des appareils **nombreux** et **hétérogènes**.
- En conséquences, les S.I. modernes sont

gros, sensibles, et compliqués.

beaucoup de code, beaucoup de matériel, beaucoup d'usagers,...

opérations, décisions, sécurité,...

langages de programmation multiples, technologies multiples, enjeux métiers multiples,...

- Pour les développer, il faut donc procéder méthodiquement.
- Il existe pour cela de nombreuses méthodes.
- Ces méthodes fournissent :
  - des **concepts et langages** pour représenter et décrire le S.I.
  - une **démarche pour progresser** dans l'analyse et la conception
- Des outils logiciels peuvent assister cette démarche.

Méthodes d'analyse et conception de S.I.

- Le S.I. d'une entreprise en est un composant clé : son bon fonctionnement est souvent critique aux opérations de celle-ci.
- Ses responsabilités sont multiples, et il est parfois dépoyé sur des appareils nombreux et hétérogènes.
- En conséquences, les S.I. modernes sont

gros, sensibles, et compliqués.

beaucoup de code, beaucoup de matériel, beaucoup d'usagers,...

opérations, décisions, sécurité,...

langages de programmation multiples, technologies multiples, enjeux métiers multiples,...

- Pour les développer, il faut donc procéder méthodiquement.
- Il existe pour cela de nombreuses méthodes.
- Ces méthodes fournissent :
  - des concepts et langages pour représenter et décrire le S.I.
  - une démarche pour progresser dans l'analyse et la conception

modéliser

- Des outils logiciels peuvent assister cette démarche.

Modèle

To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A.

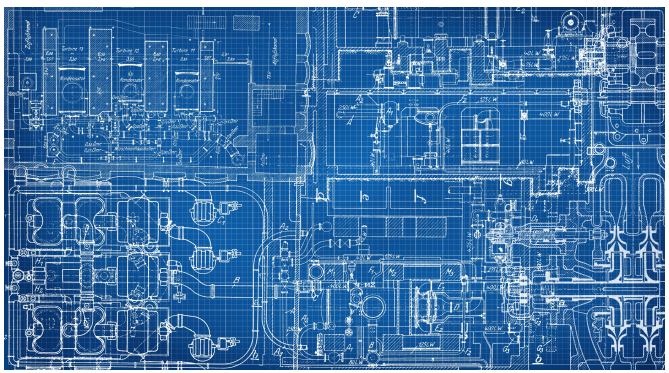
Marvin Minsky

Prix Turing de 1969 pour ses contributions à l'intelligence artificielle

Modélisation

- Un modèle est une représentation abstraite et simplifiée du monde réel dans le but de le décrire, de l'expliquer ou le prévoir
- Un modèle permet de réduire la complexité d'un phénomène en éliminant les détails qui n'influencent pas son comportement

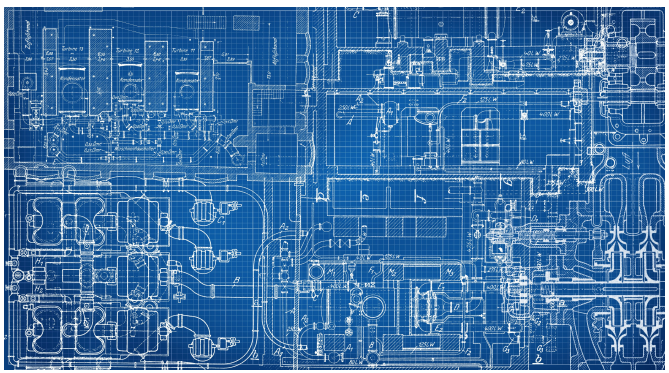
Système réel



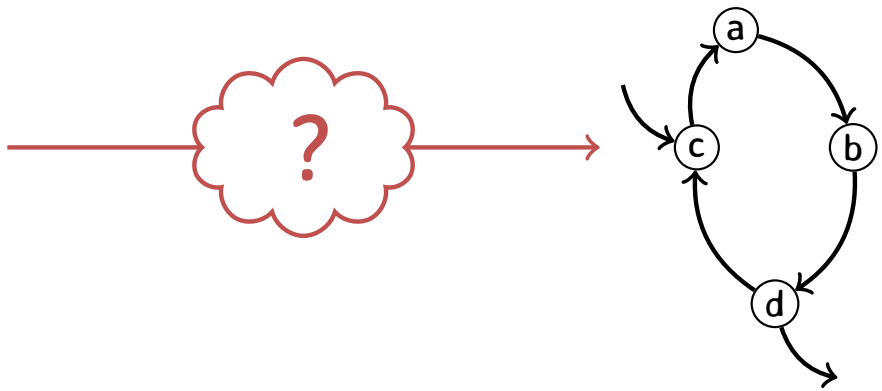
Modélisation

- Un modèle est une représentation abstraite et simplifiée du monde réel dans le but de le décrire, de l'expliquer ou le prévoir
- Un modèle permet de réduire la complexité d'un phénomène en éliminant les détails qui n'influencent pas son comportement

Système réel



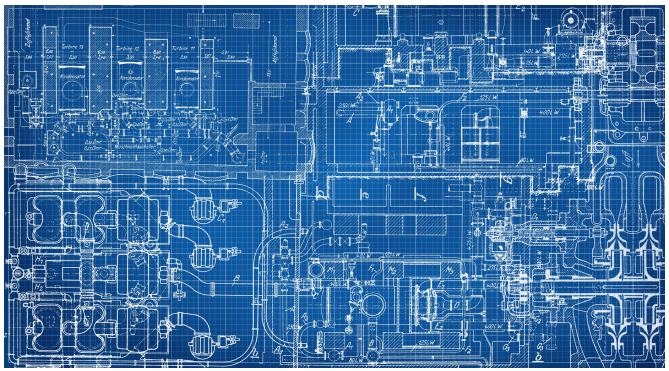
Modèle



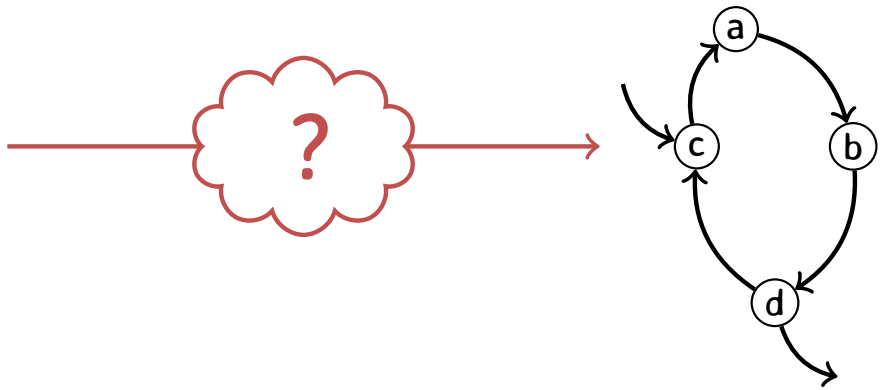
Modélisation

- Un modèle est une représentation abstraite et simplifiée du monde réel dans le but de le décrire, de l'expliquer ou le prévoir
- Un modèle permet de réduire la complexité d'un phénomène en éliminant les détails qui n'influencent pas son comportement

Système réel



Modèle

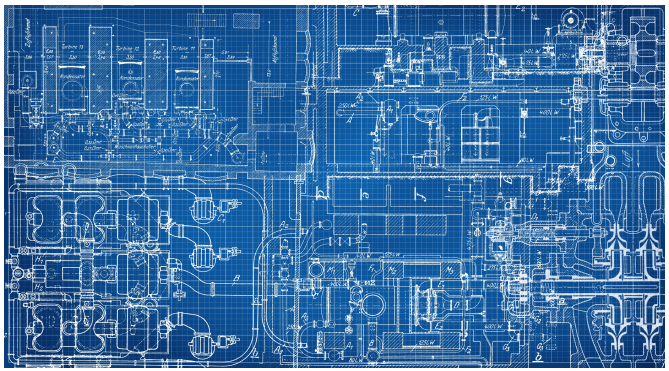


- Complexe
- Nombreux aspects

Modélisation

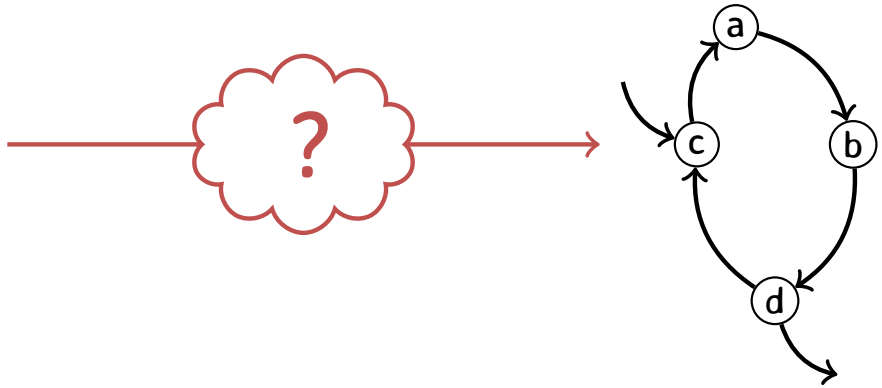
- Un modèle est une représentation abstraite et simplifiée du monde réel dans le but de le décrire, de l'expliquer ou le prévoir
- Un modèle permet de réduire la complexité d'un phénomène en éliminant les détails qui n'influencent pas son comportement

Système réel



- Complexe
- Nombreux aspects

Modèle

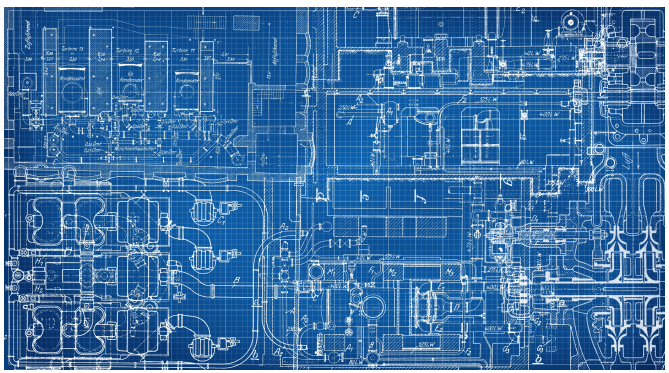


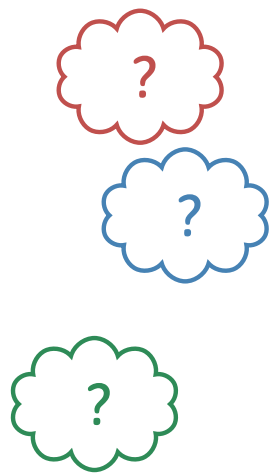
- Plus simple
- Seulement les aspects pertinents
- Raisonnements rigoureux

Intention d'un modèle

- Un modèle est construit avec une **intention**, un but, un usage prévu.
- La qualité d'un modèle est relative à cette intention.

Système réel

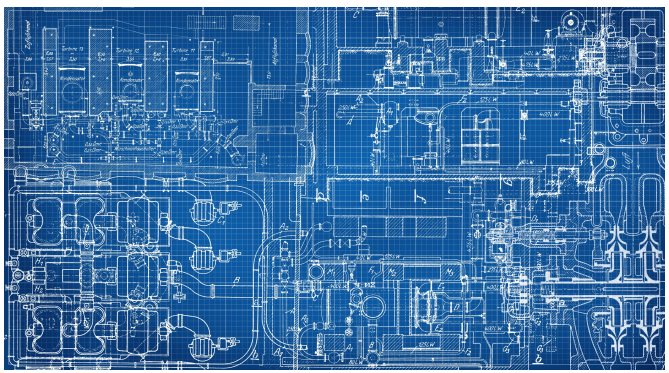


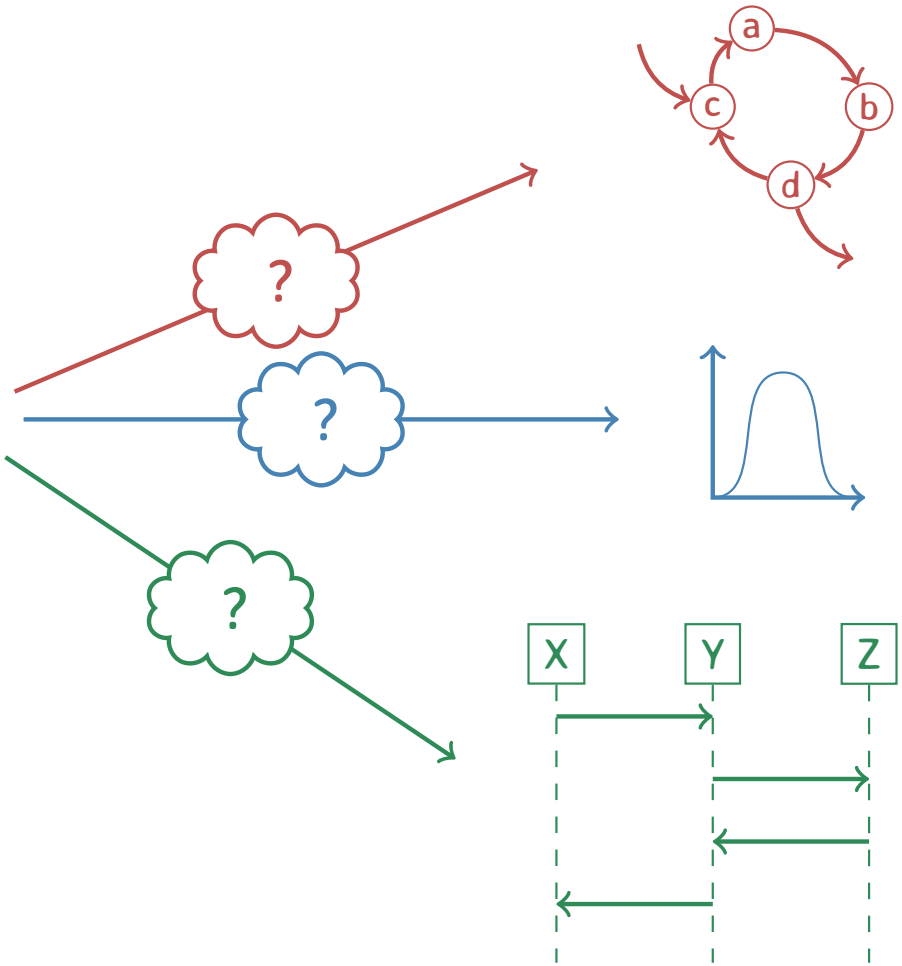


Intention d'un modèle

- Un modèle est construit avec une **intention**, un but, un usage prévu.
- La qualité d'un modèle est relative à cette intention.

Système réel





Fidélité et Efficacité

Un bon modèle doit trouver un équilibre entre ces deux extrêmes :

- ☞ Modèle trop fidèle (maquette)
  - Trop compliqué : n'aide pas à comprendre le système et ne permet pas de répondre efficacement aux questions posées
  - Trop long/coûteux à développer
- ☞ Modèle trop abstrait (poulet sphérique dans le vide)
  - Néglige des aspects importants du système
  - Les réponses ne sont pas pertinentes en pratique

Modélisation $\neq$ Modélisme !

Modélisation

Pourquoi modéliser ?

- Pour comprendre le fonctionnement d'un système ou le construire
- Un bon moyen pour maîtriser sa complexité et assurer sa cohérence.
- Avoir un langage commun connu par tous les membres de l'équipe.
- Dans l'Ingénierie de Logiciel, répartir les tâches dans une équipe.
- C'est un facteur de réduction des coûts et des délais. (génération de code, génération de modèle à partir du code, ...)

Modélisation

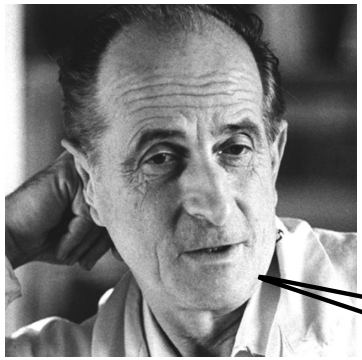
Pourquoi modéliser ?

- Pour comprendre le fonctionnement d'un système ou le construire
- Un bon moyen pour maîtriser sa complexité et assurer sa cohérence.
- Avoir un langage commun connu par tous les membres de l'équipe.
- Dans l'Ingénierie de Logiciel, répartir les tâches dans une équipe.
- C'est un facteur de réduction des coûts et des délais. (génération de code, génération de modèle à partir du code, ...)

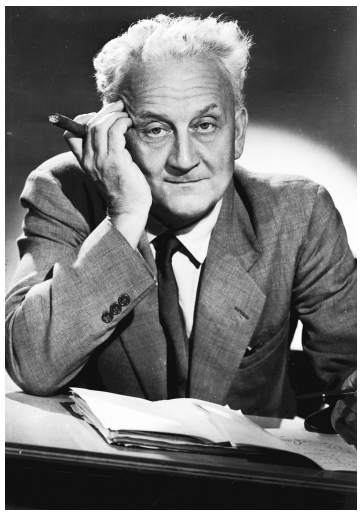
Quatre principes de la modélisation

- Le choix des modèles à créer influe sur la manière d'aborder un problème et sur la nature de la solution
- Les modèles peuvent avoir différents niveaux de précision
- Les meilleurs modèles ne perdent pas le sens de la réalité
- Un seul modèle n'est pas suffisant d'où la nécessité de décomposer un système en plusieurs petits modèles presque indépendants

« Any map will do »



Miroslav Holub

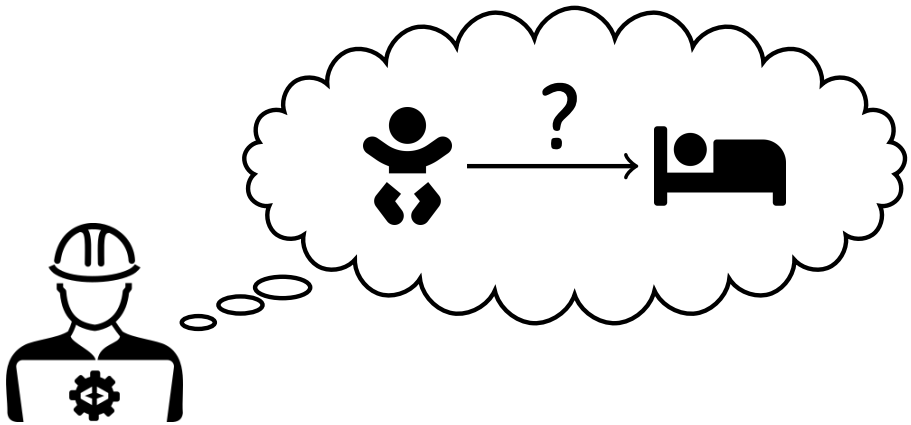


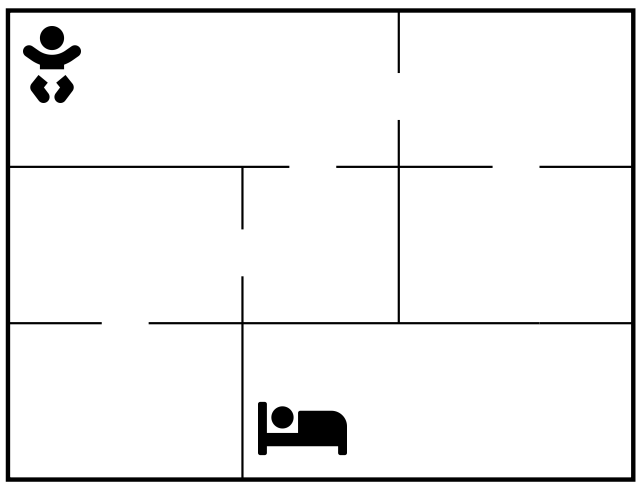
Albert Szent-Györgyi
prix Nobel de médecine 1937
pour ses travaux sur la
vitamine C

Albert Szent-Györgyi, who knew a thing or two about maps, by which life moves somewhere or other, used to tell this story from the war, through which history moves somewhere or other.
From a small Hungarian unit in the Alps a young lieutenant sent out a scouting party into the icy wastes. At once it began to snow, it snowed for two days and the party did not return. The lieutenant was in distress : he had sent his men to their deaths.
On the third day, however, the scouting party was back. Where had they been ? How did they manage to find their way ? Yes, the man explained, we certainly thought we were lost and awaited our end. When suddenly one of our lot found a map in his pocket. We felt reassured. We made a bivouac, waited for the snow to stop, and then with the map found the right direction. And here we are.
The lieutenant asked to see that remarkable map in order to study it. It wasn't a map of the Alps But the Pyrenees.
Goodbye.

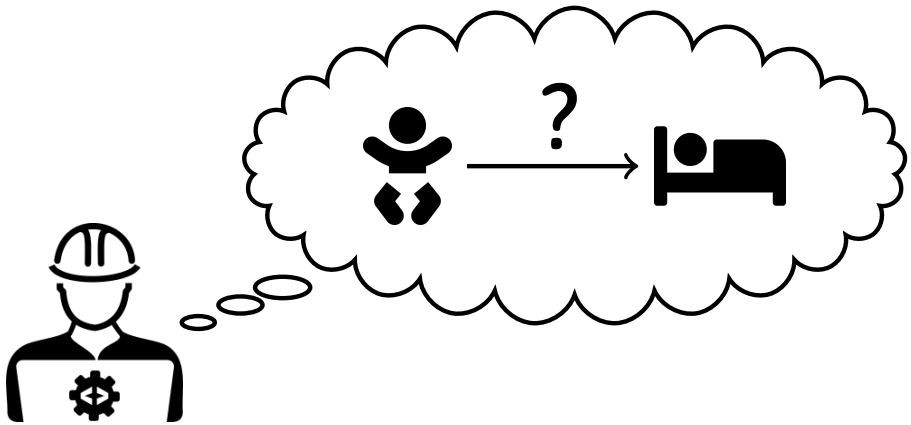
« Brief Reflection on Maps » (1977)

Au lit !
Exemple

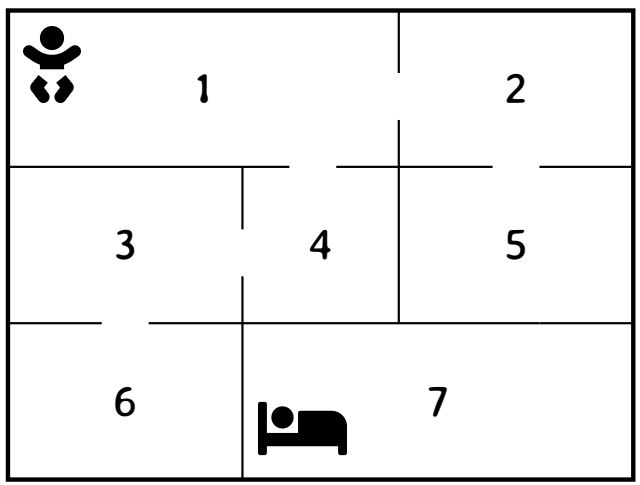




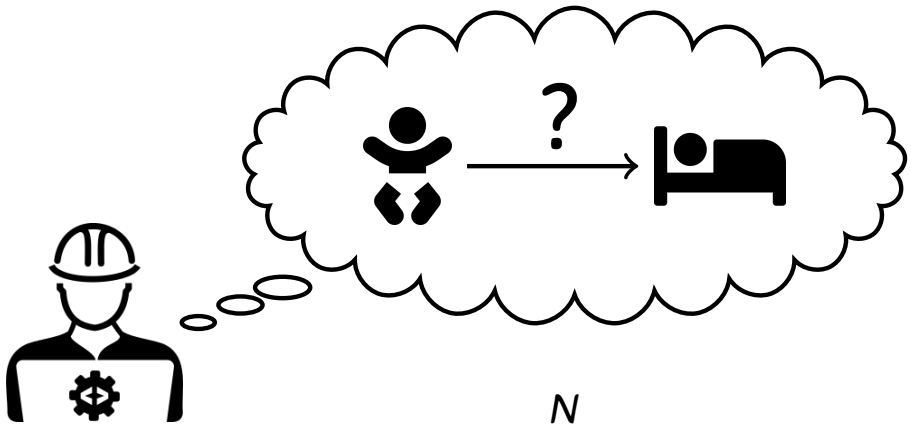
Au lit !
Exemple



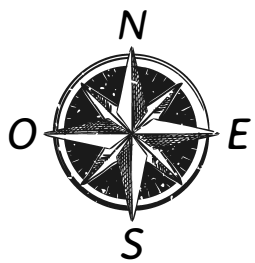
☞ États : pièces de 1 à 7 ;

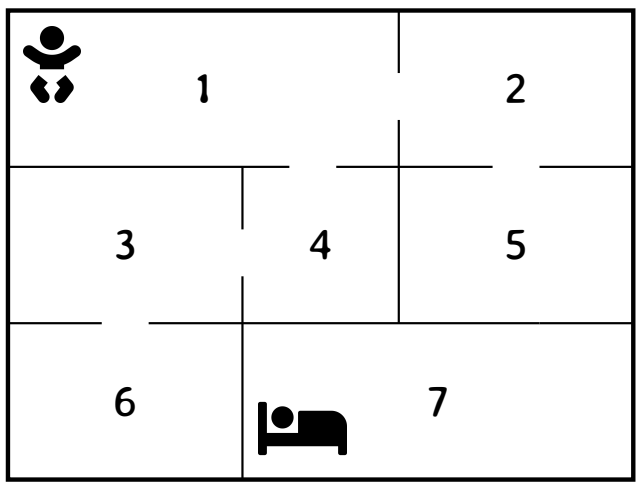


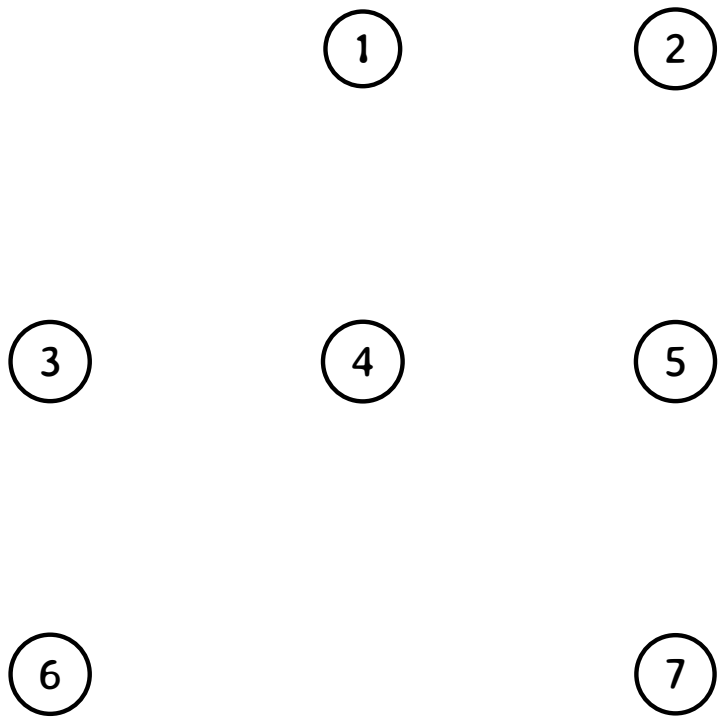
Au lit !
Exemple



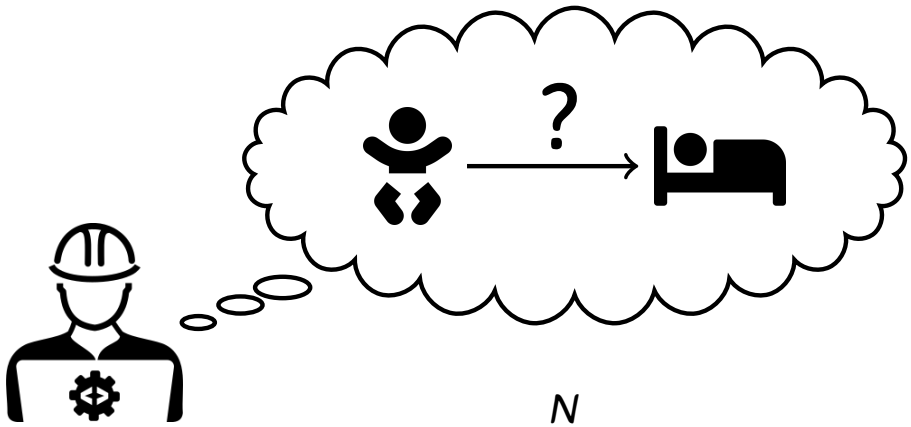
- États : pièces de 1 à 7 ;
- Alphabet : $\{N, S, E, O\}$;



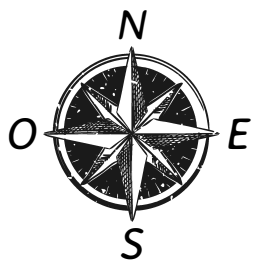


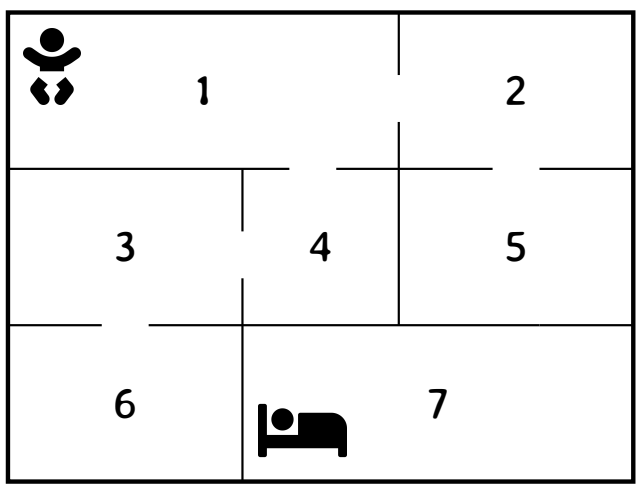


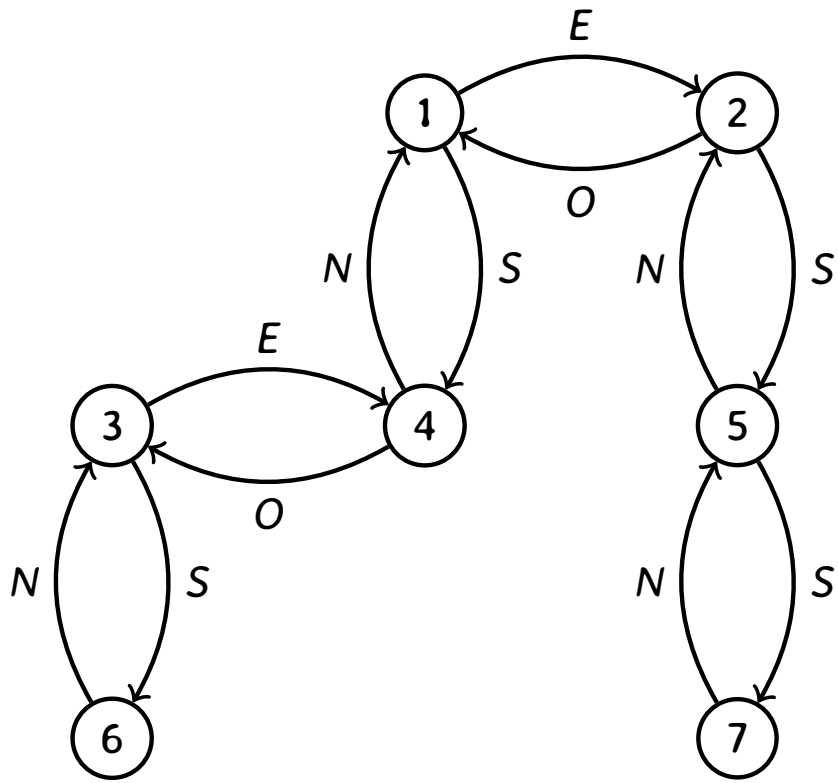
Au lit !
Exemple



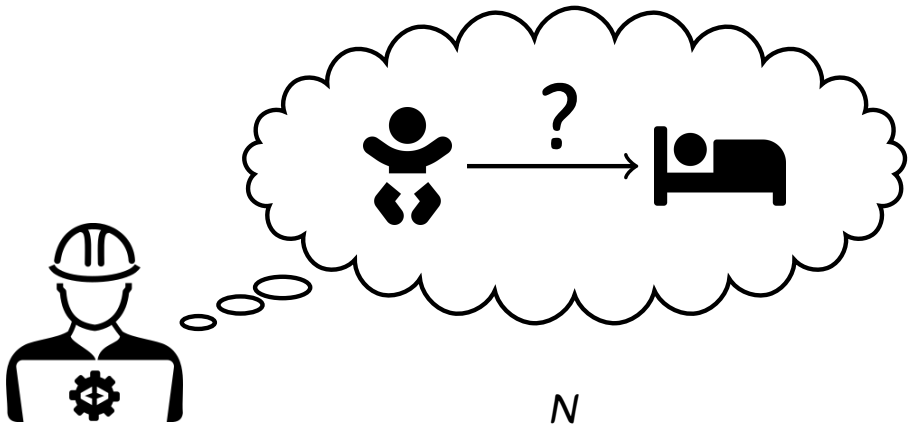
- États : pièces de 1 à 7 ;
- Alphabet : $\{N, S, E, O\}$;
- Transitions : portes ;



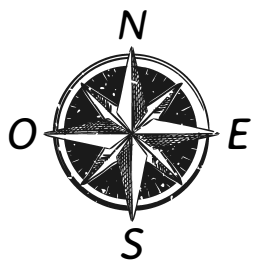


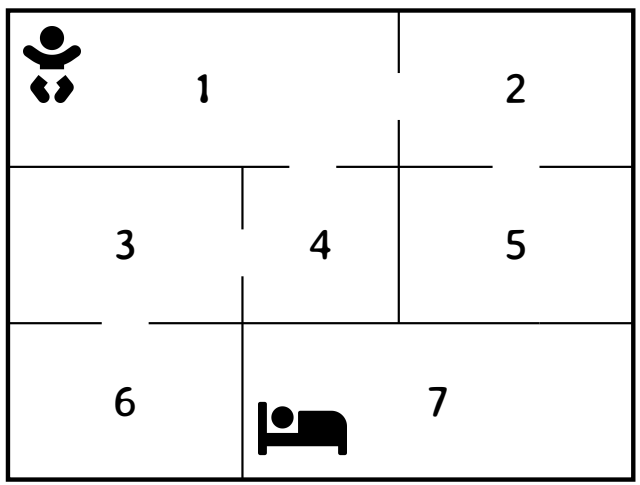


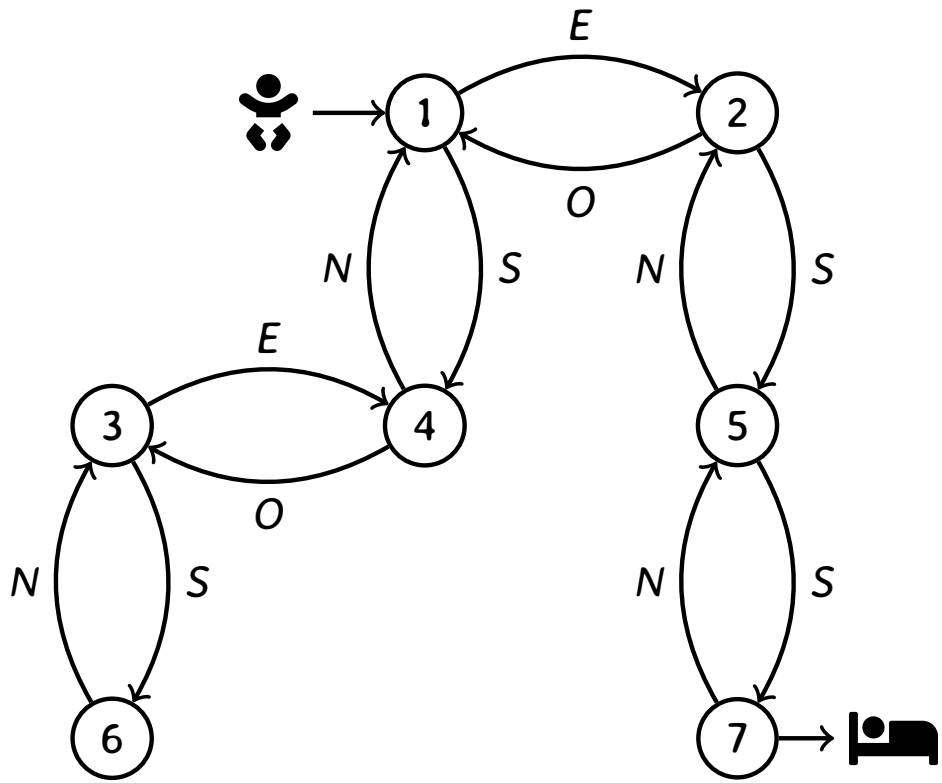
Au lit !
Exemple



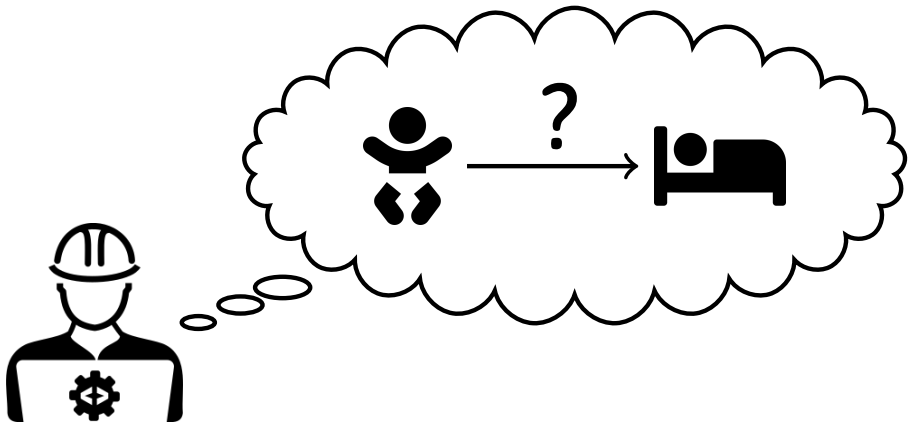
- États : pièces de 1 à 7 ;
- Alphabet : $\{N, S, E, O\}$;
- Transitions : portes ;
- États initial/final.







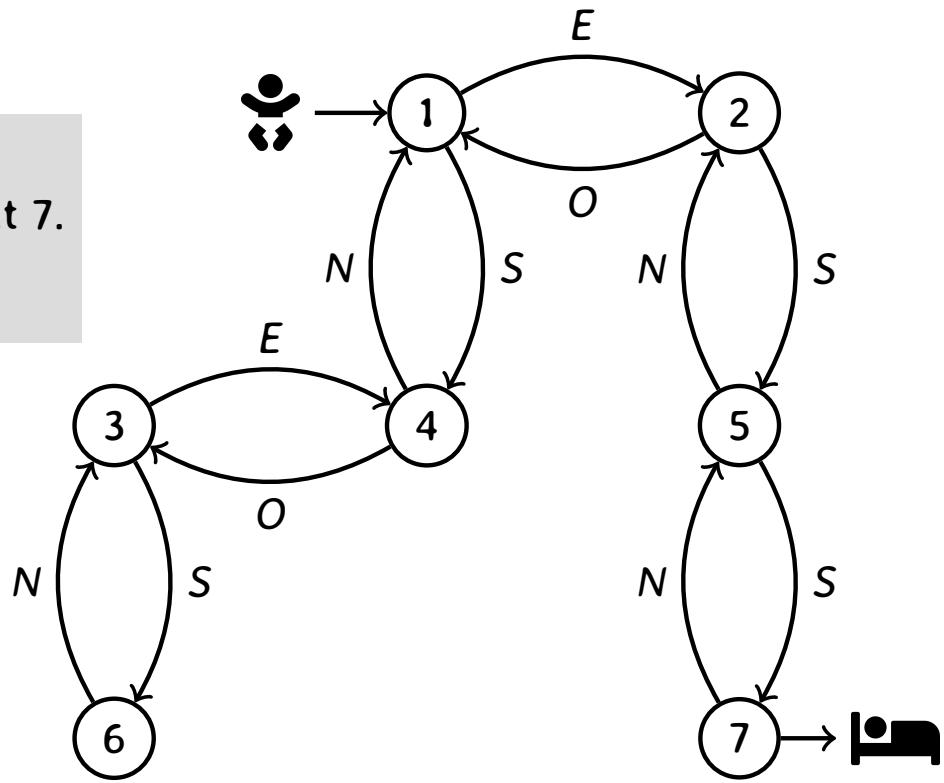
Au lit !
Exemple



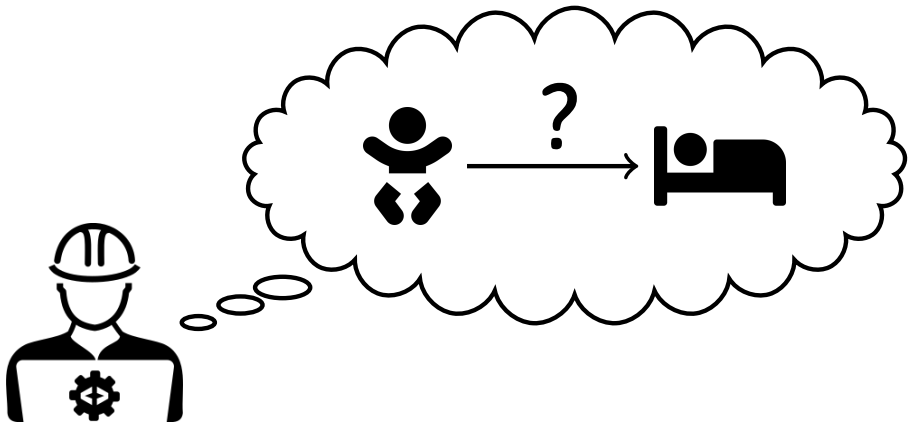
- États : pièces de 1 à 7 ;
- Alphabet : $\{N, S, E, O\}$;
- Transitions : portes ;
- États initial/final.

Langage de l'automate :

- Chemins menant de l'état 1 à l'état 7.
- Itinéraires de  à  .



Au lit !
Exemple



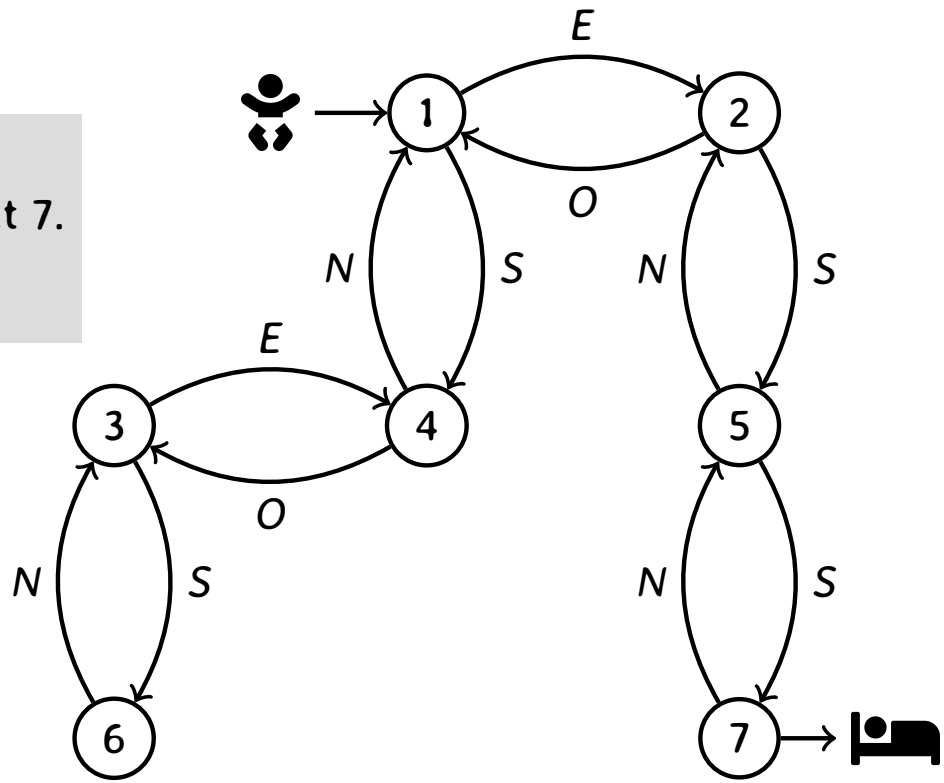
- États : pièces de 1 à 7 ;
- Alphabet : $\{N, S, E, O\}$;
- Transitions : portes ;
- États initial/final.

Langage de l'automate :

- Chemins menant de l'état 1 à l'état 7.
- Itinéraires de  à  .

Absent du modèle :

- Distances,
- Points de passage,
  salle de bain (couche),
  cuisine (biberon) ...
- ...



Introduction
Plan

1. Présentation du cours

2. Systèmes d'information et modèles
   2.1. Systèmes d'information
   2.2. Modèles
   2.3. Exemple

3. Analyse et Conception Orientées Objet
   3.1. Origine
   3.2. Principes
   3.3. Difficultés

4. Unified Modeling Language

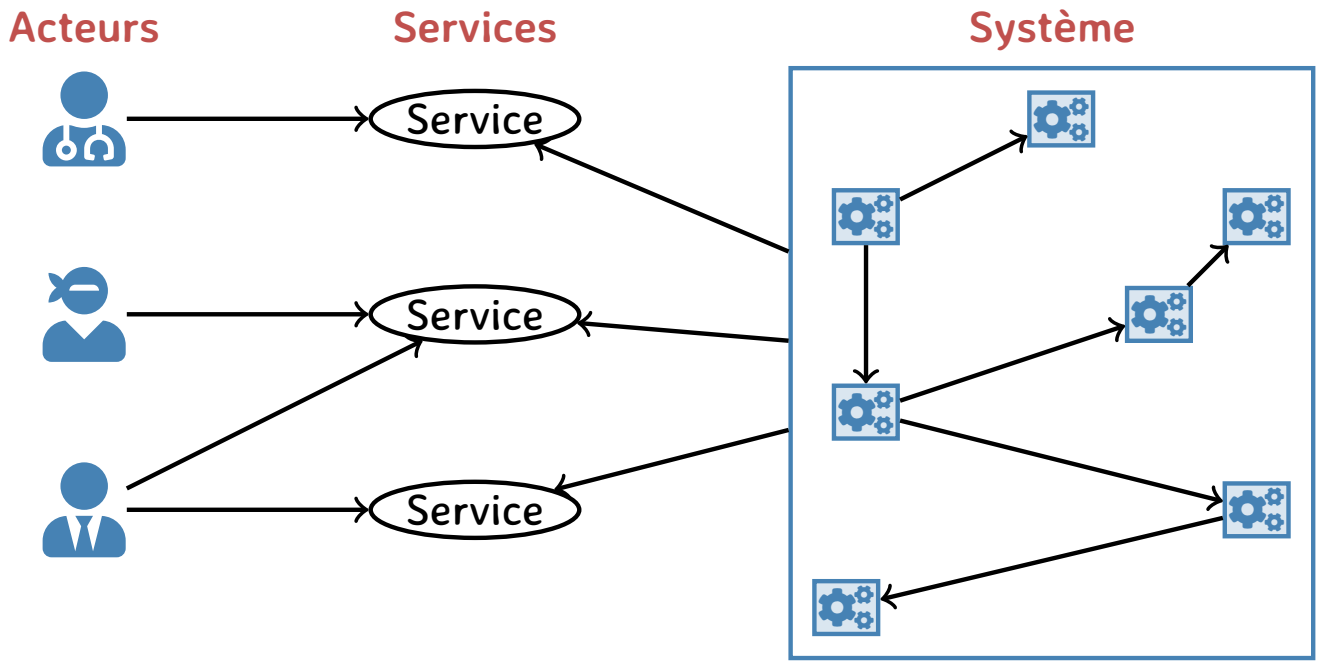
Analyse et Conception Orientées Objet
Origine

- Les langages de programmation
- L'orientation objet en programmation : Pourquoi ?
  - Brique de base de la structuration du logiciel
  - La qualité du logiciel

    Extensibilité, **Réutilisabilité**, **Compatibilité**
  - Lisibilité
  - Sécurité et maintenance

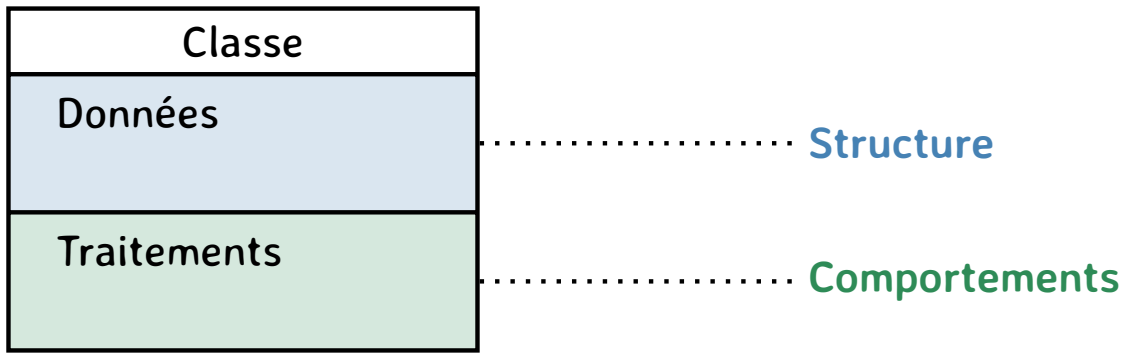
définition (point de vue objet)

Le S.I. est vu comme une collection d'objets qui interagissent pour fournir aux acteurs les services attendus.
Ces objets communiquent avec l'extérieur et entre eux, en échangeant des messages.



Quel changement pour l'analyse et la conception de S.I. ?

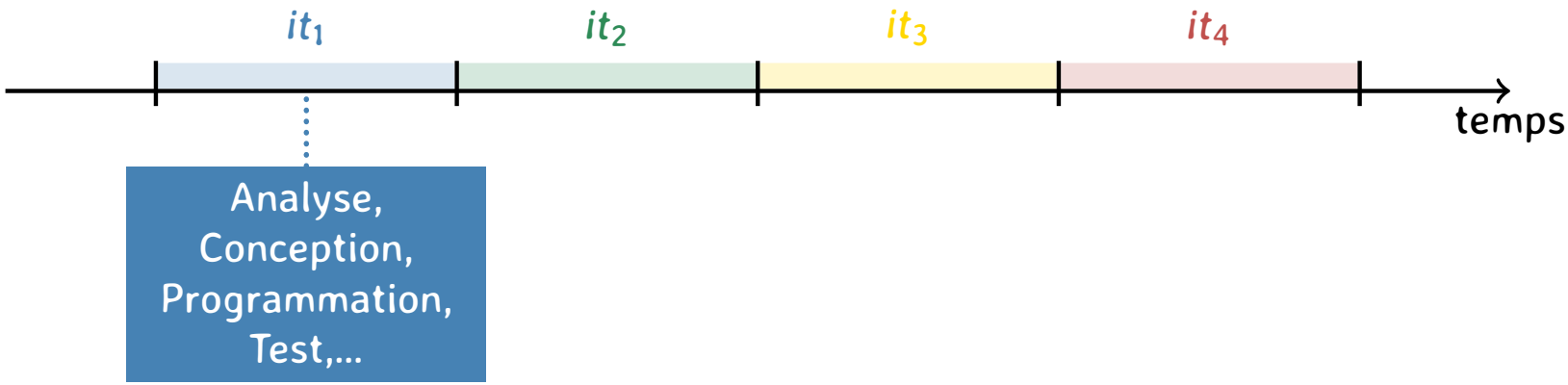
- La remise en cause de la séparation données/traitements
- La classe comme une nouvelle forme d'abstraction



- L'unification du langage de modélisation pour l'analyse, la conception et l'implantation
  - Utilisation des mêmes concepts généraux pour décrire différents phénomènes
  - L'objet pour décrire les objets applicatifs, les objets informatiques, les acteurs
- Traçabilité / Cohérence

Quel changement pour l'analyse et la conception de S.I. ?

- Le processus de développement est itératif
  - Chaque étape conduit à élaborer des résultats précis (documents, diagrammes, modèles…) ou produits du développement
  - Chaque étape comporte un ensemble de phases, d'activités, de tâches
  - Chaque étape utilise des concepts et des règles appropriées (formalisme)
- Les produits de développement sont construits de manière incrémentale



- Les modèles sont modifiés, étendus, raffinés au fil des itérations, en fonction :
  - des besoins
  - des retours (tests systématiques, expériences clients, évaluations…)

Analyse et Conception Orientées Objet
Difficultés

- Le poids des langages de programmation O.O.
- La difficulté de modéliser le comportement d'une collection d'objets
- Le manque de métriques pour mesurer la qualité d'un schéma objet
- L'acquisition et l'expression des besoins sont oubliées

Introduction
Plan

1. Présentation du cours

2. Systèmes d'information et modèles
   - 2.1. Systèmes d'information
   - 2.2. Modèles
   - 2.3. Exemple

3. Analyse et Conception Orientées Objet
   - 3.1. Origine
   - 3.2. Principes
   - 3.3. Difficultés

4. Unified Modeling Language

The rise of UML

- En 1994 : jusqu'à 50 méthodes objet
  - notation et processus spécifiques : OMT, OOAD, OOSE,
  - mais une certaine convergence sur les concepts
- Création d'UML
  - fusion des notations (1996)
  - Adopté comme un standard de l'OMG (1998)
  - Spécification : UML 2.2
- Les objectifs d'UML
  - Créer des modèles décrivant des logiciels orientés objet et des systèmes d'entreprise
- Principes de l'approche UML
  - L'orientation objet
  - L'analyse/conception guidée par les acteurs et les services attendus
  - Un processus itératif et incrémental

Unified Modeling Language
Caractéristiques

- Langage de modélisation visant à :
  - comprendre et décrire des besoins
  - spécifier et documenter des systèmes
  - présenter des architectures logicielles
  - concevoir des solutions
  - communiquer des points de vue
- Langage graphique :
  - éléments de modélisation / éléments de visualisation
  - 14 types de diagrammes
- Langage général :
  - indépendant d'un langage de programmation
  - indépendant d'un type d'architecture
  - indépendant d'un processus de conception
- Langage adaptable :
  - Un mécanisme d'extension permettant d'étendre le langage

Unified Modeling Language
Caractéristiques

Le langage UML s'appuie sur le paradigme Objet répondant aux principes suivants :

Abstraction : concept de classe, d'objet, instance

Encapsulation : concept d'interface

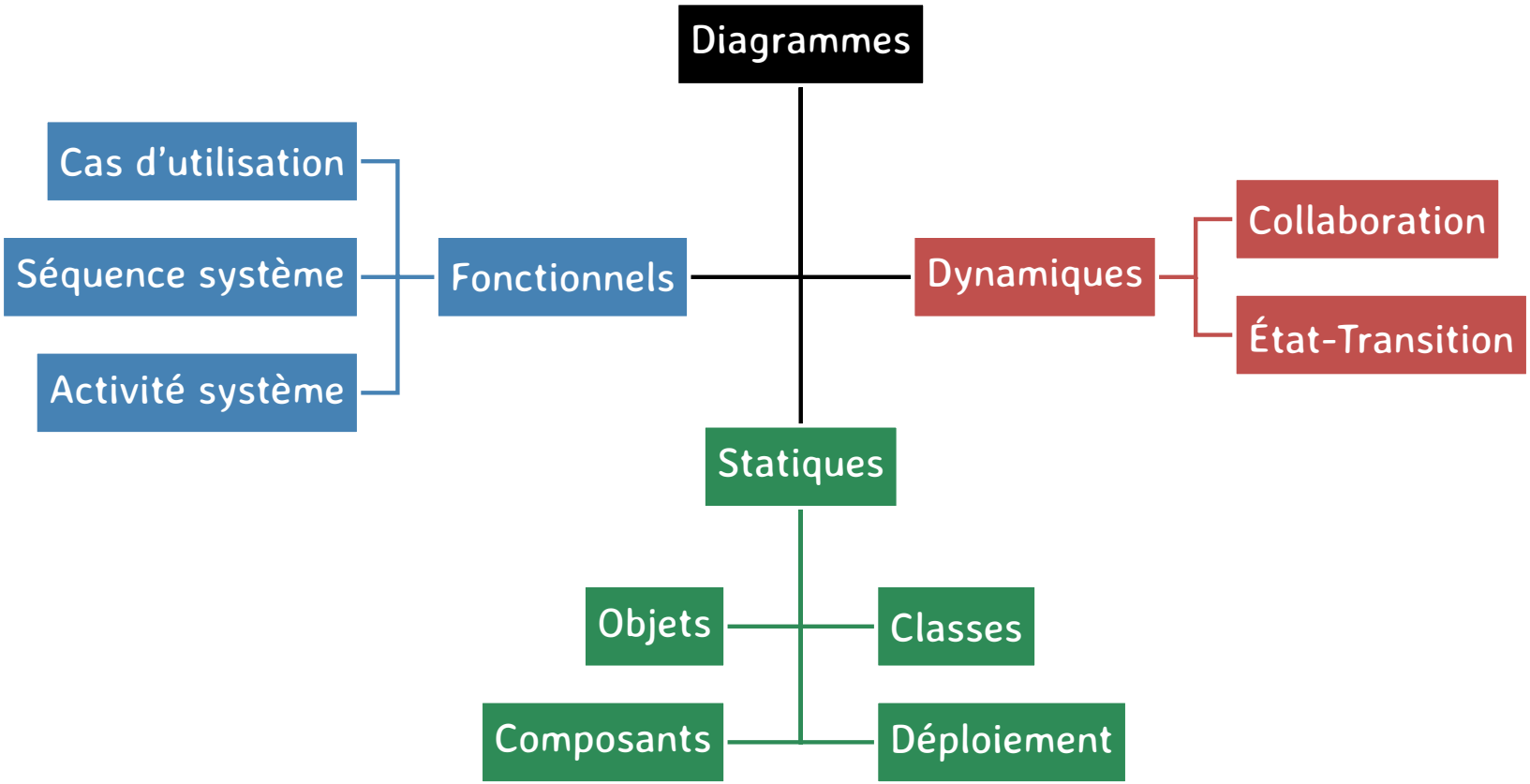
Modularité : Classe, Composant, Package

Hiérarchie : concept d'héritage

Unified Modeling Language
Vocabulaire

- **Les éléments de base**
  Cas d'utilisation, classe, interface, composant, nœud, etc.
- **Les relations**
  Association, Généralisation/Spécialisation, Dépendances, Réalisation
- **Les diagrammes**
  Diagramme de cas d'utilisation, diagramme de classes, diagrammes d'états transitions, etc.

Unified Modeling Language
Les diagramme



- La possibilité de présenter le même diagramme à différents niveaux de détail
- La possibilité d'utiliser le même diagramme dans différentes situations
- Des mécanismes communs

Unified Modeling Language
Les diagrammes

- diagrammes de cas d'utilisation : fonctionnalités du système
- diagrammes de classes : structure du système
- diagrammes d'objets : illustration des structures de classes
- diagrammes de composants : architecture logicielle du système
- diagrammes de déploiement : architecture physique du système
- diagrammes d'états/Transitions : cycle de vie des objets
- diagrammes d'activités : règles d'enchaînement des activités dans le système
- diagrammes de séquence/Collaboration : échange de messages entre objets

UML 2.5 rassemble 14 diagrammes

Unified Modeling Language
Apports et limites

Apports

- Le langage UML est une **notation standardisée** qui permet d'exprimer les résultats (produits) du développement c.à.d. la solution
- Un ensemble **cohérent et complet** de techniques de modélisation
  - les différentes dimensions (statique, dynamique, fonctionnel)
  - les différents niveaux d'abstraction(conceptuel, logique, physique)
  - les différents niveaux de détail (général, détaillé)
- L'existence de **règles de passage** entre UML et Java

Limites

- Les travaux autour d'UML ont mis l'accent sur la notation graphique.
- Ce n'est pas une méthode.
- La difficulté de choisir les diagrammes à utiliser.