Urbanisation de système d'information

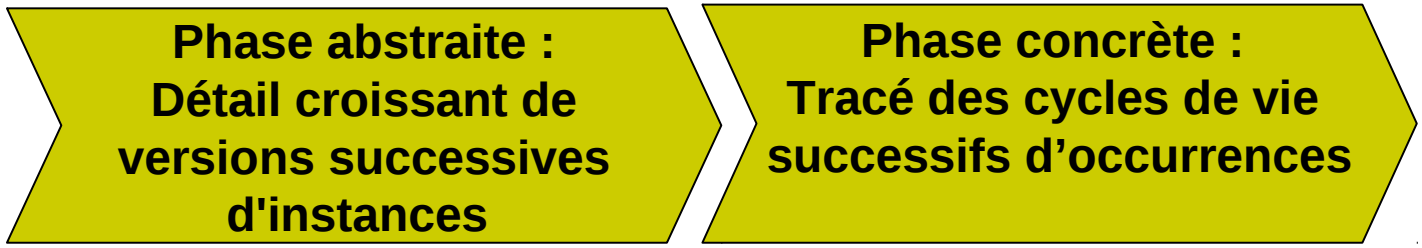
PLM 3 (Product Lifecycle Management)

Élaborations,

versions, variantes, configurations

Mise en gestes

- L'existence de tout **produit**, et de tout **service** commence par une phase ***abstraite*** de spécification avant une phase ***concrète*** de réalisation.
- La phase abstraite de **spécification** et la **modification** de chaque **objet**, **implique** une ou plusieurs **activités** de type **documentaire**.
- Les **activités** de spécification qui **modifient** un **objet** en **créent** des **versions successives,** portées par **des versions** de **documents**.
- Issue d'une **décision** en vue d'un **objectif, c**haque **activité est effectuée par** un **acteur**, **exerçant** un **rôle** qui **requiert** une **compétence**.
- Un **processus assemble** des **activités** pour **réaliser** un **objectif**.
- L'organisation des **activités** d'un **processus** est l'objet de **procédures**.
- L'urbanisation formalise les relations de traçabilité entre processus, procédures, activités, acteurs, compétences et versions d'objets métier.

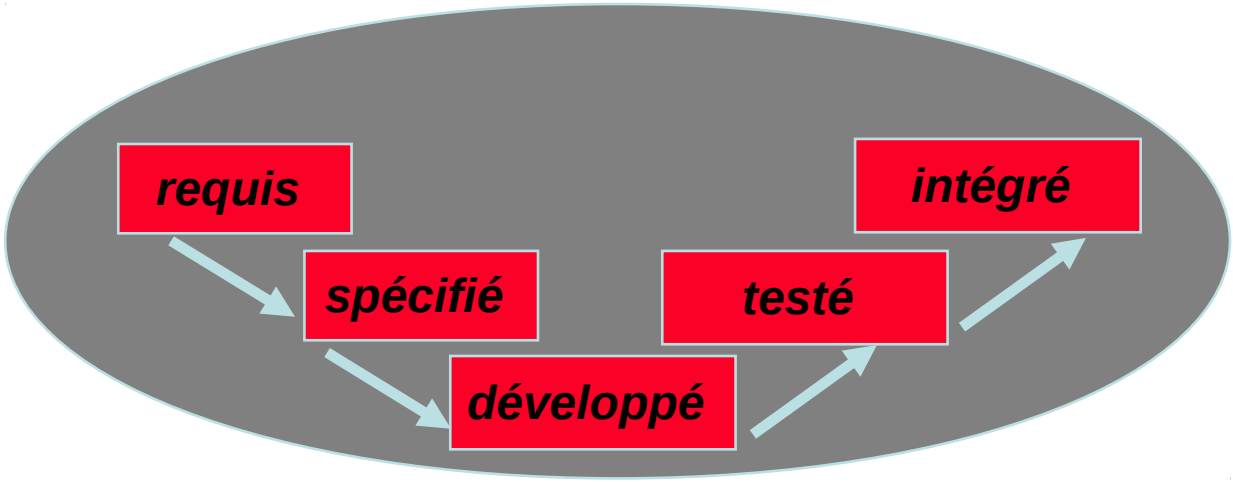


Phases de développement

- La conception **d'objets** complexes nécessitent autant de **phases** successives que de niveaux de complexité **identifiés**.
  - Avant-projet, projet, étude détaillée, étude de réalisation etc
- La définition d'un **produit** s'effectue selon un ordre croissant de **détail**.
- La synchronisation des **activités** implique la réalisation de **cycles** complets **d'étude** pour chaque niveau conventionnellement **identifié** de complexité.
- À chaque niveau de complexité **identifié** correspond une **phase d'étude,** avec un **objectif**, **sujet** d'évaluation :
  - **Décision** de poursuite.
  - Définition des **exigences** à appliquer au niveau de détail suivant.
  - Ramification de **variantes**, selon des **exigences.**
- À chaque **phase** correspond une **étape du cycle de vie** associée aux **objets** étudiés.

Cycles de Vie des données et des documents

- Paradoxe de « **cycles de vie** »,
  - attachés, par essence, à des entités « mortes » !
- Finalité de **cycles de vie** formalisés :
  - Ils tracent les **étapes d'itération** de l'élaboration d'informations,
  - Ils associent des **niveaux de qualité** correspondant à des exploitations spécifiques définis dans les **processus** de l'organisation.



Produits concrets et représentations abstraites

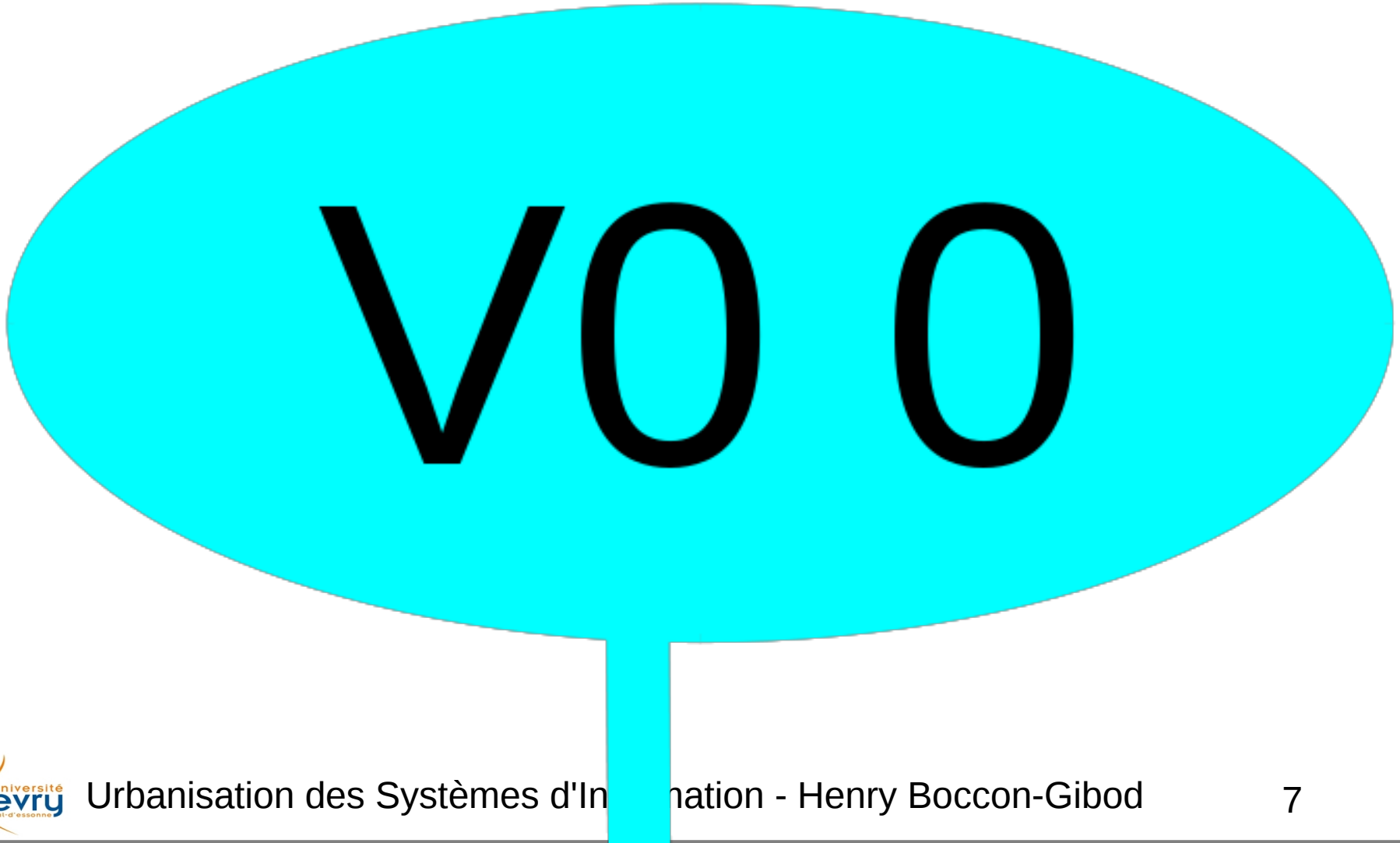
- La vie des **produits concrets** est
  - **précédée**,
  - **accompagnée**,
    - de représentations abstraites dans des **documents concrets**
- Les **produits** concrets ont leur propre **cycle de vie** :
  - **Monté**
  - **Testé**
  - **Vendu, sous garantie**
  - **Exploité, Maintenu, sans garantie**
  - **Exploité, non maintenu**
  - **Réformé**
  - **Détruit, recyclé**
- Le suivi de chaque phase de la vie d'un **produit** concret fait l'objet de représentations, matérialisées dans des **documents** ou des **bases de données.**
- La maintenance d'un produit peut faire évoluer sa **configuration** d'origine.

Diversités : versions, configurations d'usages

- Les défauts des **produits** nécessitent d'être corrigés au fur et à mesure de leur apparition.
- Les **exigences** de fonctionnalités évoluent dans le temps.
- La diversité des usages diversifie les **exigences** de fonctionnalités.
- Une **configuration** d'un **produit** répond à un assemblage **identifié d'exigences** sur des fonctionnalités.
  - Il est des usages qui les appellent "**version** majeure".
- Une **version** d'une **configuration de produit** correspond à l'amélioration de sa capacité à répondre aux **exigences** correspondantes.
  - Elle en corrige les défauts identifiés.
  - Il est des usages qui les appellent "**version** mineure".
- L'adoption de **versions** de correction de parties engendrent des **versions** de correction de leurs assemblages.
- L'intégration de nouvelles **configurations** de **parties** engendre de nouvelles **configurations** pour des assemblages où ils apparaissent.

versions et variantes de configurations

- **Identification** d'ensembles gérés composites.



Applicabilités

- Une **applicabilité** est caractérisée par un ensemble **d'exigences** associées à une **configuration**, nécessitées par un **usage identifié**.
- La gestion de **versions** et de **configurations** consiste à
  - Tracer les évolutions des contenus d'informations et de documents
  - Tracer les héritages des élaborations découplées, concurrentes, des informations et documents,
  - Associer les **produits**, **documents** et informations pertinents à une **Applicabilité**

Retour sur la question des noms, propres et communs

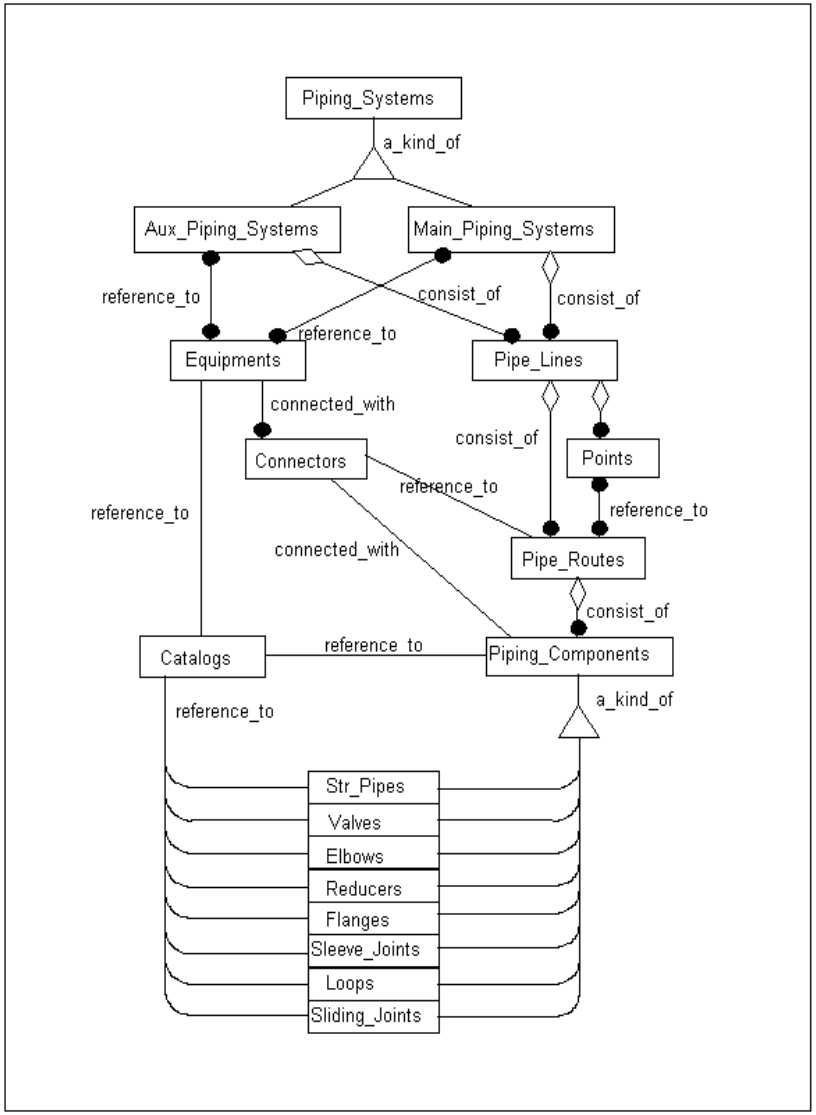
- **Nommer** les **objets** de façon pertinente est difficile
  - Ordinairement laissée à la discrétion de chaque utilisateur.
- Pour ce qui est des implémentations informatiques :
  - Les systèmes de fichiers se contentent d'interdire les conflits de noms.
  - Les systèmes de gestion de bases de données génèrent leur propres clés uniques pour tout enregistrement.
- La **cohérence** d'un **Système d'Information** exige de traiter les notions linguistiques nécessaires pour la **coopération** des **activités,**
  - de **noms propres**, associés à des **objets uniques,**
  - de **noms communs** associés à des typologies **d'objets** et de **caractéristiques** .

Noms propres : les identifiants d'occurrences

- Une gamme **d'objets** est caractérisée par
  - des niveaux de **configurations**,
  - des **versions** et/ou des dates de création,
  - des **phases de cycle de vie**,
  - des **applicabilités.**
- Une **identification** sémantique inefficace associerait chacun de ces éléments configurations:applicabilité:phase:version
  - par exemple le repère : **RRA:12VP:MO1:GRA2:PREL:V12.4** désignerait le **moteur 1** de la **vanne 12 VP** du **système RRA** à installer à **Gravelines 1**, en **version 12.4**, telle que validée pour la **phase préliminaire.**
- Pour éviter les ruptures de liens pointant vers la version valide d'un objet :
  - Une identification sémantique efficace de la version courante d'un objet n'associe que des **désignations de** sa **configuration.**
  - La **version** n'est éventuellement rattachée au **nom** que pour tracer sans conflit de noms l'histoire des **versions** obsolètes.

Noms communs : identifiants de catégories

- Les **noms communs** sont destinés à **catégoriser** les entités:
  - Les noms communs désignent une **caractéristique** principale d'un ensemble d'entités : ex: *les aveugles, les chanteurs .*
- L'ensemble des **noms communs** utilisés pour la désignation des entités de l'entreprise définissent son **champ lexical**.
- Les **concepts** et les **liens** entre **concepts** qu'ils désignent, forment le **métamodèle** des représentations des **objets** de l'entreprise.



Catalogues

- Un **catalogue** est un ensemble d'**objets abstraits**
  - Réalisables à l'identique sous forme d'**objets concrets.**
- Un objet d'un catalogue a un **identifiant,**
  - Qui est un **nom propre** ,
    - **Statique** : pour un **catalogue en extension .**
      - Le modèle Renault Twingo
    - **Dynamique** : pour un **catalogue en intention .**
  - Qui est (hélas ?) utilisable comme un **nom commun,**
    - Ma (vieille) Peugeot 205.
- Une **occurrence** d'un objet de catalogue,
  - A son propre **identifiant** : le "**numéro dans la série du type**".

De l'abstrait au concret : arborescence de spécialisation dans un catalogue

- Les Catégories de classement, les catalogues et les matériels exploités introduisent les notions de
  - **Classes**,
  - **Instances,** et
  - **Occurrences.**

- Les arborescences de spécialisation abstraites identifient des Classes (sortes) d'objets métier (*abstraits*).
  - formaliser les **caractéristiques** dont chaque niveau fige la **valeur.**
  - formaliser les **caractéristiques** que chaque niveau **introduit.**

- Les **valeurs** de **caractéristiques** d'**objets métier** concrétisables, **identifient** des **instances** (*abstraites*) dénombrables dans un **catalogue**
  - formaliser les **valeurs** de **caractéristiques** de détermination de leur choix.

- Les représentations d'objets réels permettent de tracer les **occurrences d'objets exploités** (*concrets*).
  - enregistrer les **valeurs** de **caractéristiques** observées.
  - **identifier** les **caractéristiques contextuelles temporelles.**

Abstraction et concrétisation des liens caractéristiques

- Philosophie :
  - les « concepts » s'identifient et se caractérisent par leurs liens mutuels
    - Cf. notion de thésaurus, de définition dans un dictionnaire.
  - un objet est représenté par des valeurs de « caractéristiques » associées à un concept.
- Pragmatique
  - Les **valeurs** de **caractéristiques de classification** identifient les classes :
    - Types, sous-type etc. exemple : pompe centrifuge verticale.
  - Les **valeurs** de **caractéristiques d'instances** les identifient, et concrétisent la possibilité d'existence d'une occurrence.
    - Exemple : Vérin X23-200x600, de diamètre 200mm et de course 600mm
  - Les **valeurs** de **caractéristiques d'occurrence** tracent **l'histoire d'une occurrence**.
    - Exemple : valeur de déformation, cotes d'usure
  - Les **valeurs** de **caractéristiques temporelles** ou **contextuelles** identifient les états possibles temporellement réversibles d'une occurrence.
    - Exemple : état ouvert ou fermé d'un commutateur.

L'abstrait et le concret : les descriptions en "extension" et en "intention"

- Questions **d'identification** d'objets :
  - Comment nommer, référencer, par une appellation ou un code non ambigu les objets d'un **catalogue** ?
- Questions de philosophie d'élaboration :
  - Est-il vraiment nécessaire de les référencer individuellement dans le **catalogue** ?
- Une **description en extension** affecte une **référence unique** à toute **instance** dans un **catalogue.**
  - Exemple : le catalogue de la Redoute, d'Ikéa etc.
- Une **description en intention** affecte une **règle de définition** à des ensembles possibles **d'instances** à partir de **valeurs** de **caractéristiques** de construction.
  - Exemple : la représentation d'un objet tridimensionnel par composition de primitives géométriques par opposition à sa représentation par ses contours.

Fin du module