

Abdelhak-Djamel Seriali
seriali@lirmm.fr

MaREL/LIRMM

Génie logiciel

- ❁ Constat : complexité du logiciel
 - ❁ Problème : coût et qualité non maîtrisés

- ❁ Objectifs du GL
 - ❁ Réduire les coûts du développement, d'évolution et de maintenance
 - ❁ Maîtriser/Améliorer la qualité et la fiabilité

- ❁ Automatisation du cycle de développement
 - ❁ Concepts : Objet, composant, service, aspect, ...
 - ❁ Méthodes : V, agile, UML, ...
 - ❁ Langages : De programmation, d'architecture, de conception, etc.

Quelques axes de travaux

- ❁ Réutilisation/évolution
 - ❁ Ligne de produits logiciels
 - ❁ Architecture à base de composants logiciels

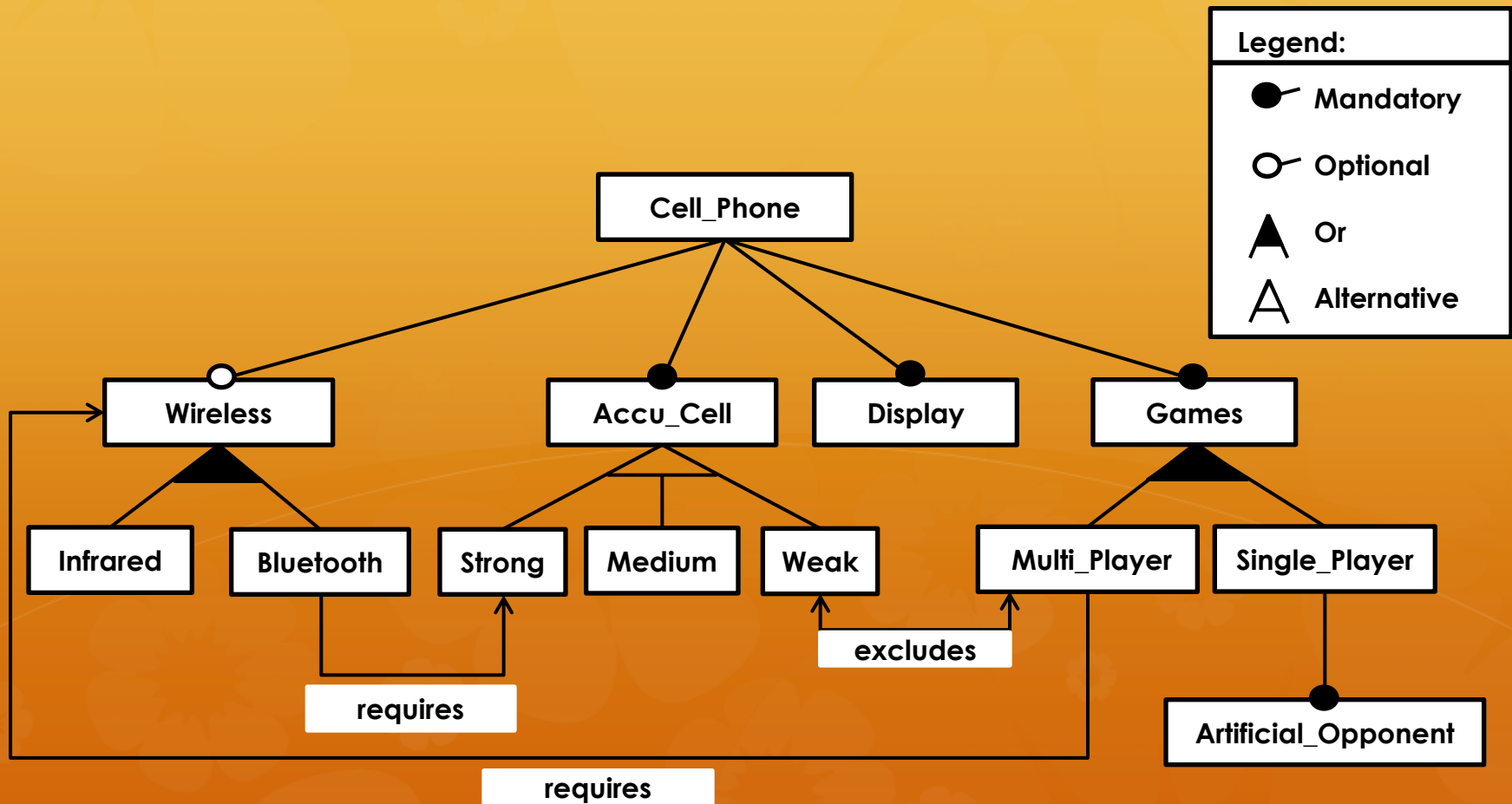
- ❁ Ingénierie dirigée par les modèles
 - ❁ Dérivation de produits (codes, modèle exécutable) spécifiques par raffinement et transformation successifs de modèles abstraits
 - ❁ Model drive architecture : CIM, PIM, PSM,

- ❁ Sémantique et implémentation des langages de programmation
 - ❁ Orientés objets
 - ❁ À base de composants

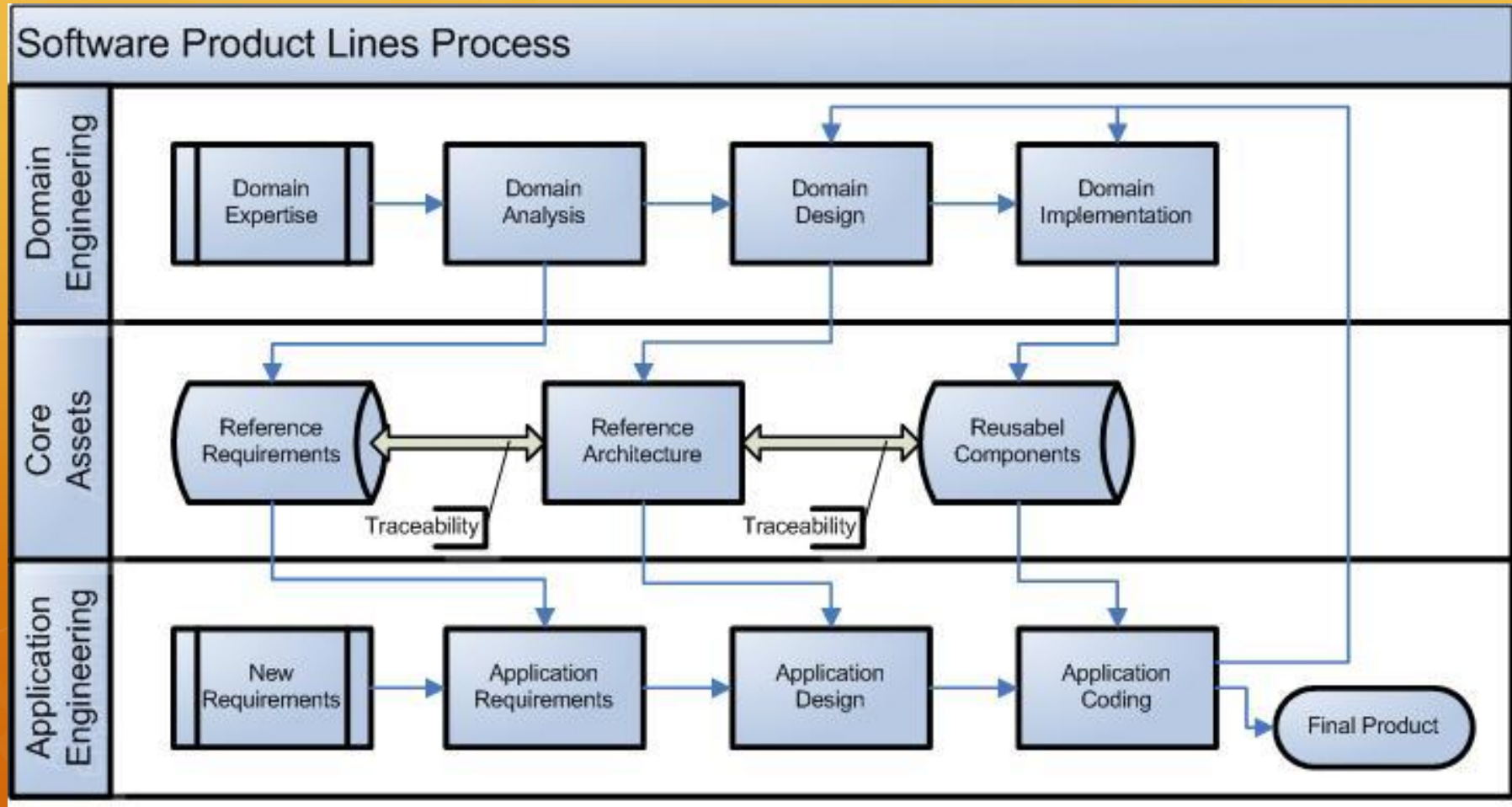
Lignes de produits logiciels

- ✿ Une démarche de développement d'une famille de logiciels
- ✿ Identification, modélisation et gestion des **caractéristiques communes et variables** des membres d'une famille logicielle
- ✿ Exemples
 - ✿ Une application de rencontre sur mobile
 - ✿ Application « Centre informatique »

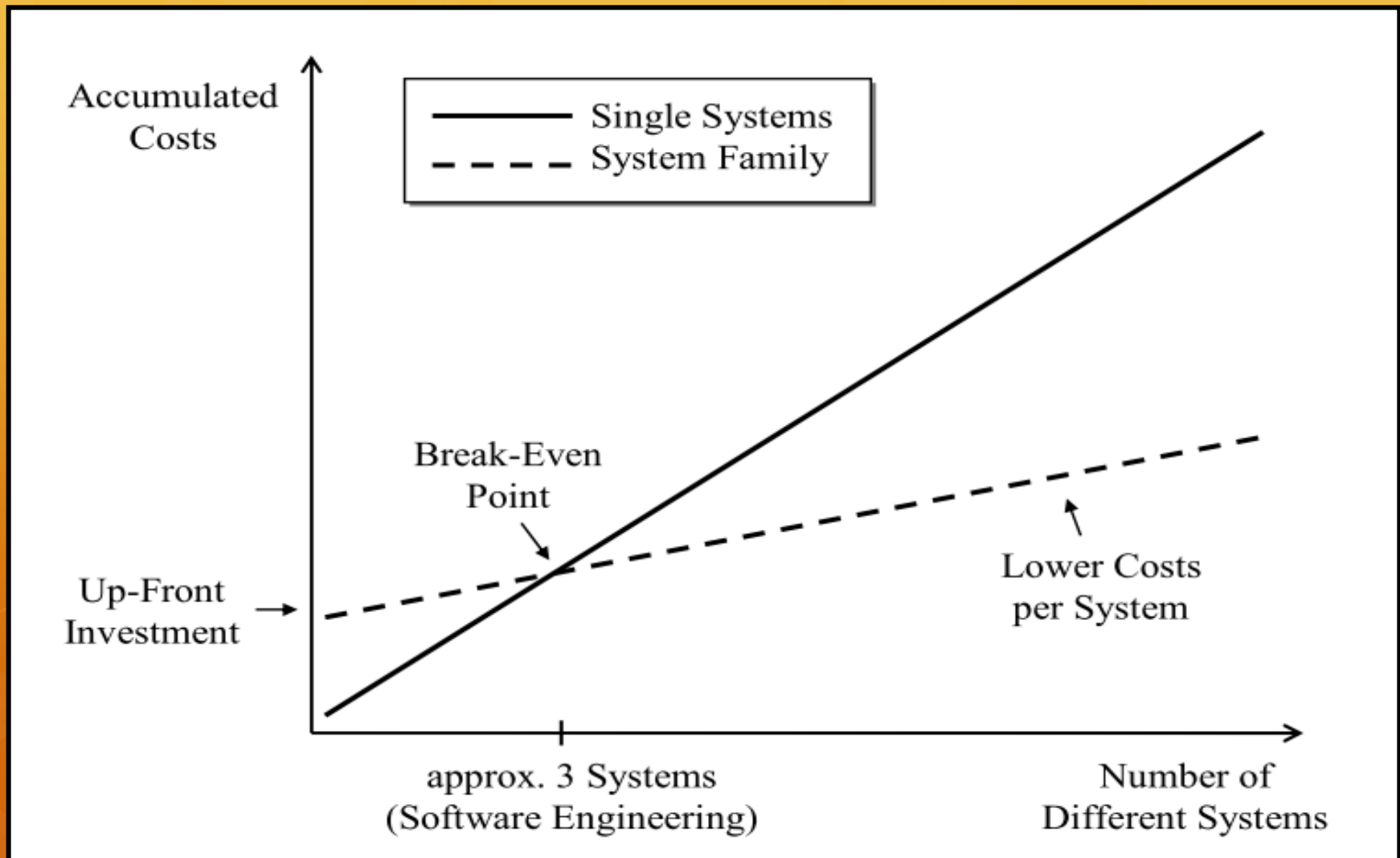
Lignes de produits logiciels



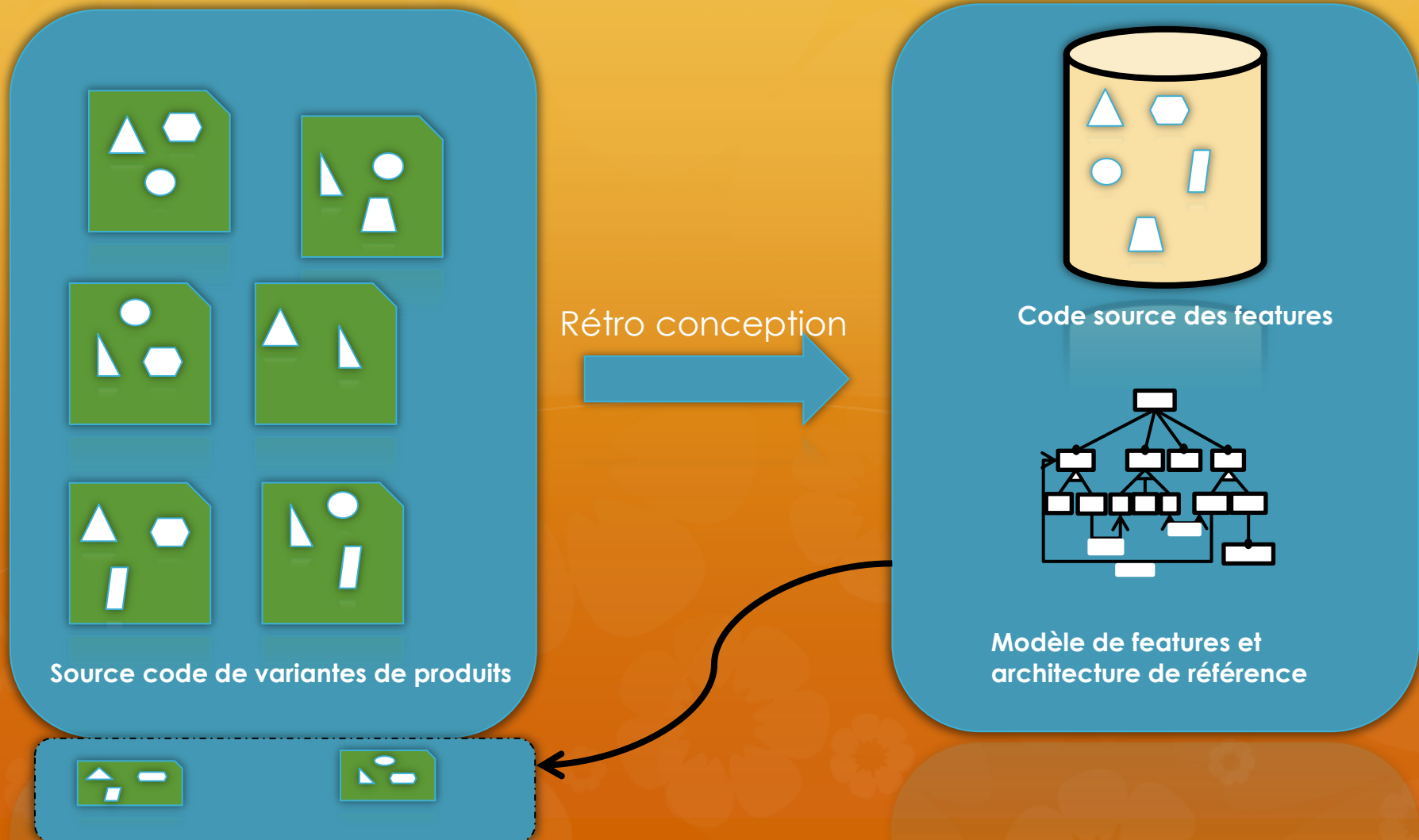
Lignes de produits logiciels



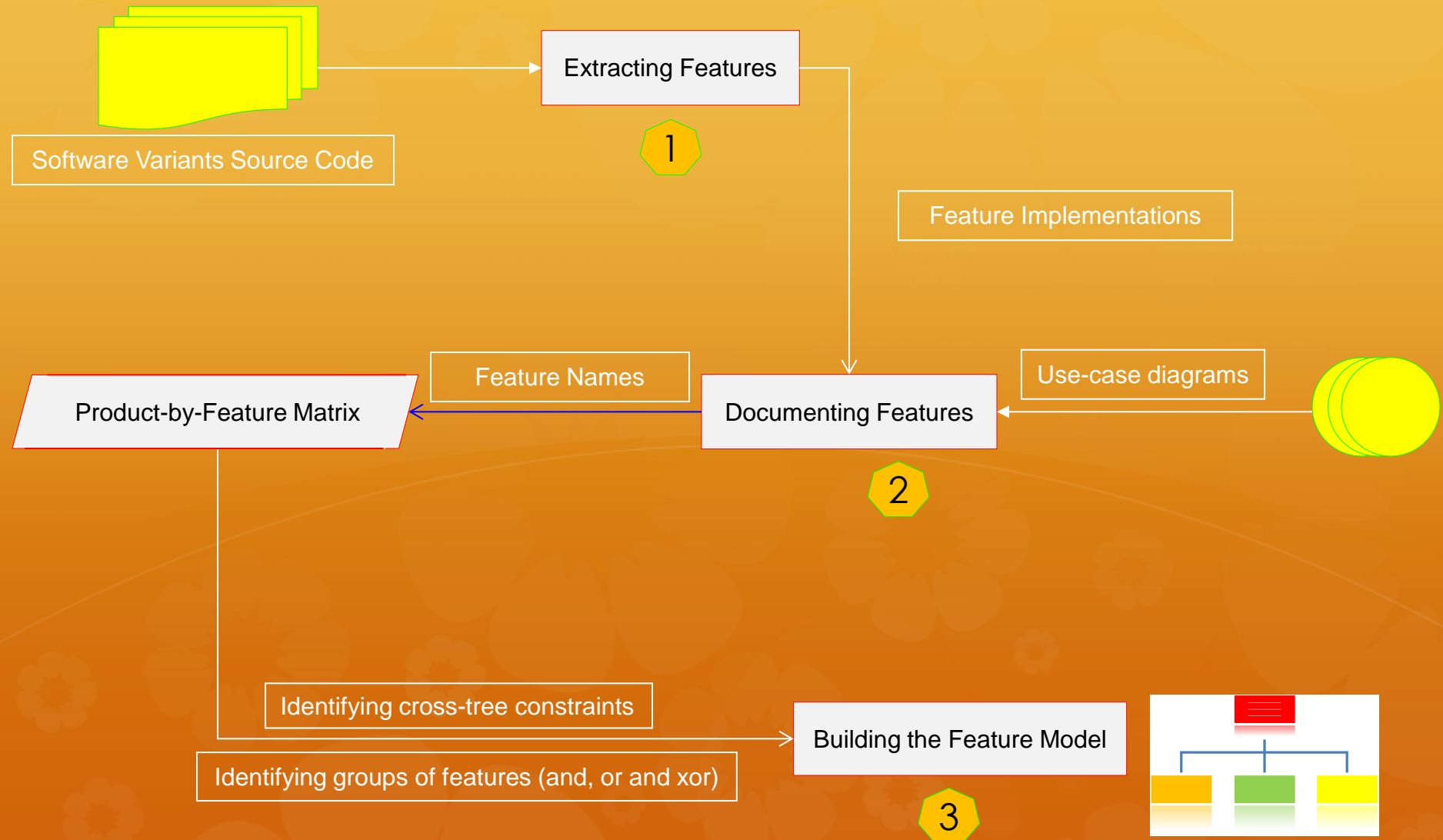
Lignes de produits logiciels



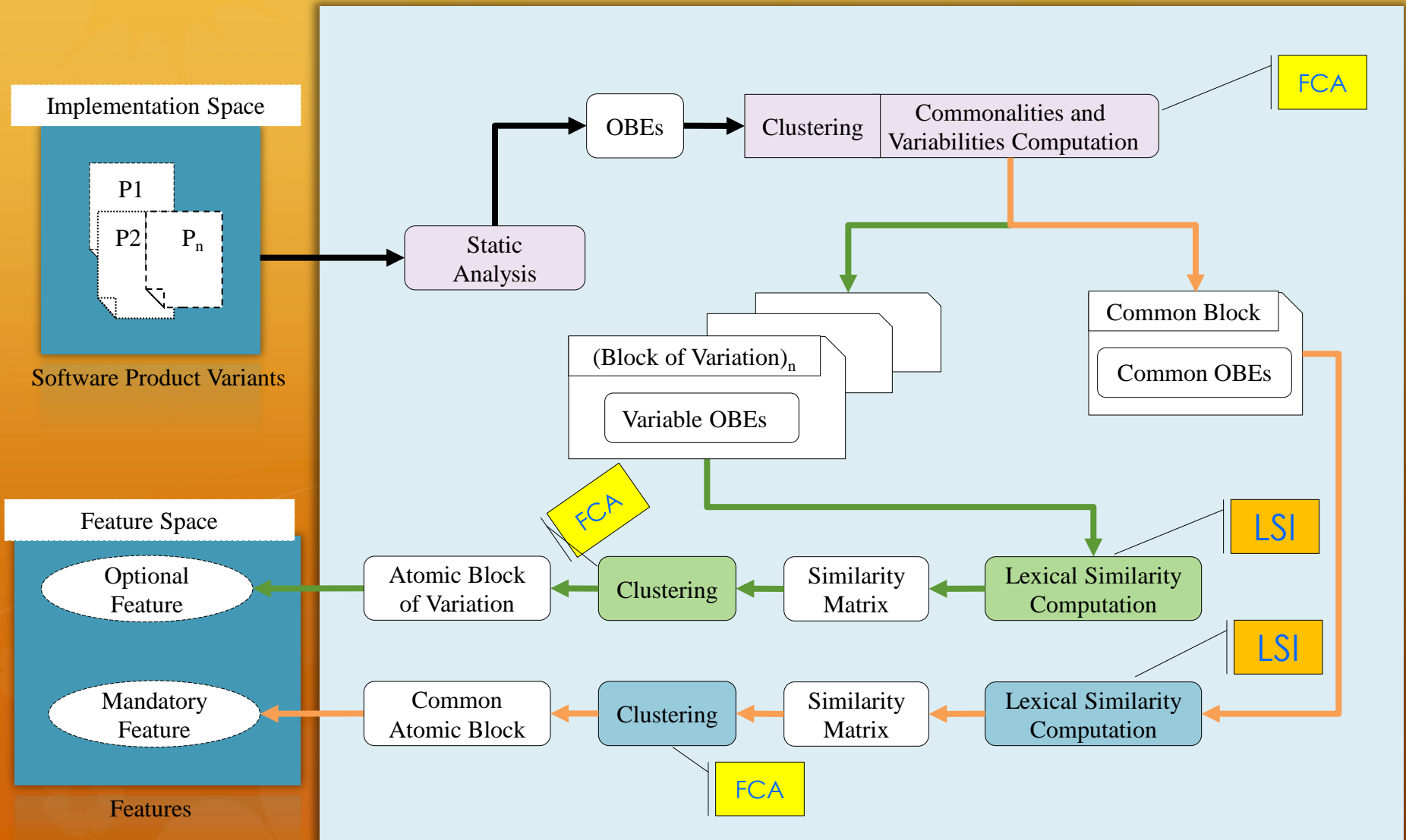
Rétro ingénierie des lignes de produits



Rétro ingénierie des lignes de produits

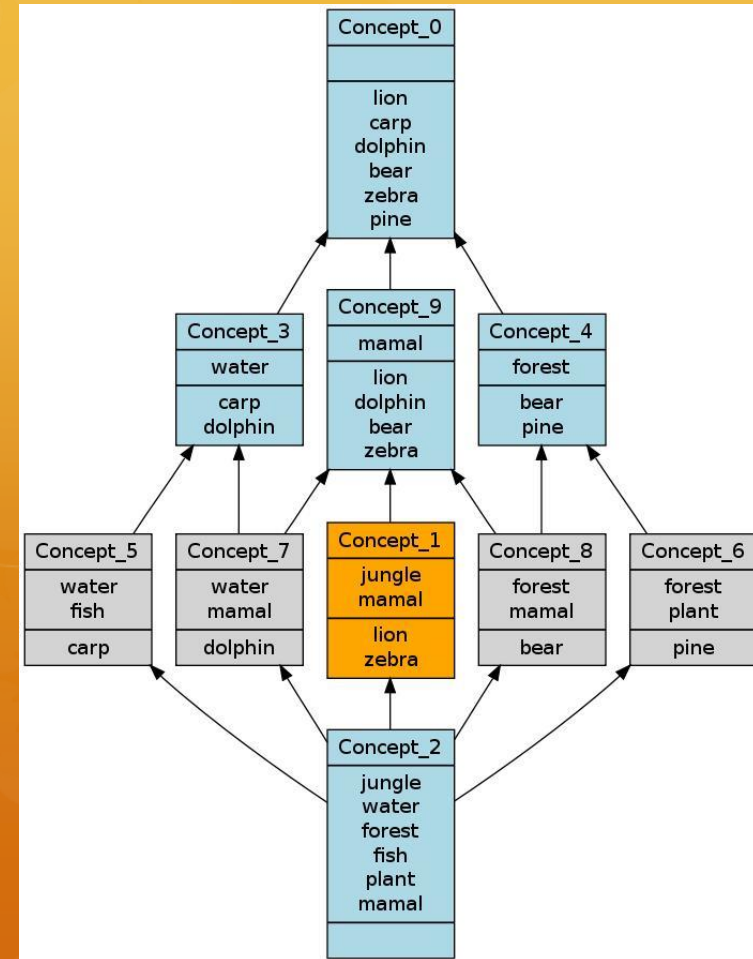


Rétro ingénierie des lignes de produits



Rétro ingénierie des lignes de produits

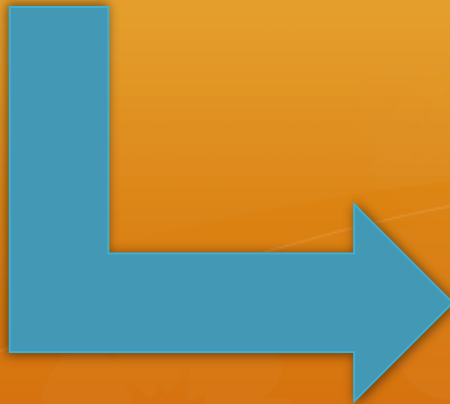
	jungle	water	forest	fish	plant	mamal
lion	x					x
carp		x		x		
dolphin		x				x
bear			x			x
zebra	x					x
pine			x		x	



Rétro ingénierie des lignes de produits

TEXT EDITOR VARIANTS DESCRIBED BY THEIR FEATURES

Variant Name	Features
Editor_1	Core (Open, Close, Print)
Editor_2	Core, Select_all
Editor_3	Core, Copy, Paste



The formal context

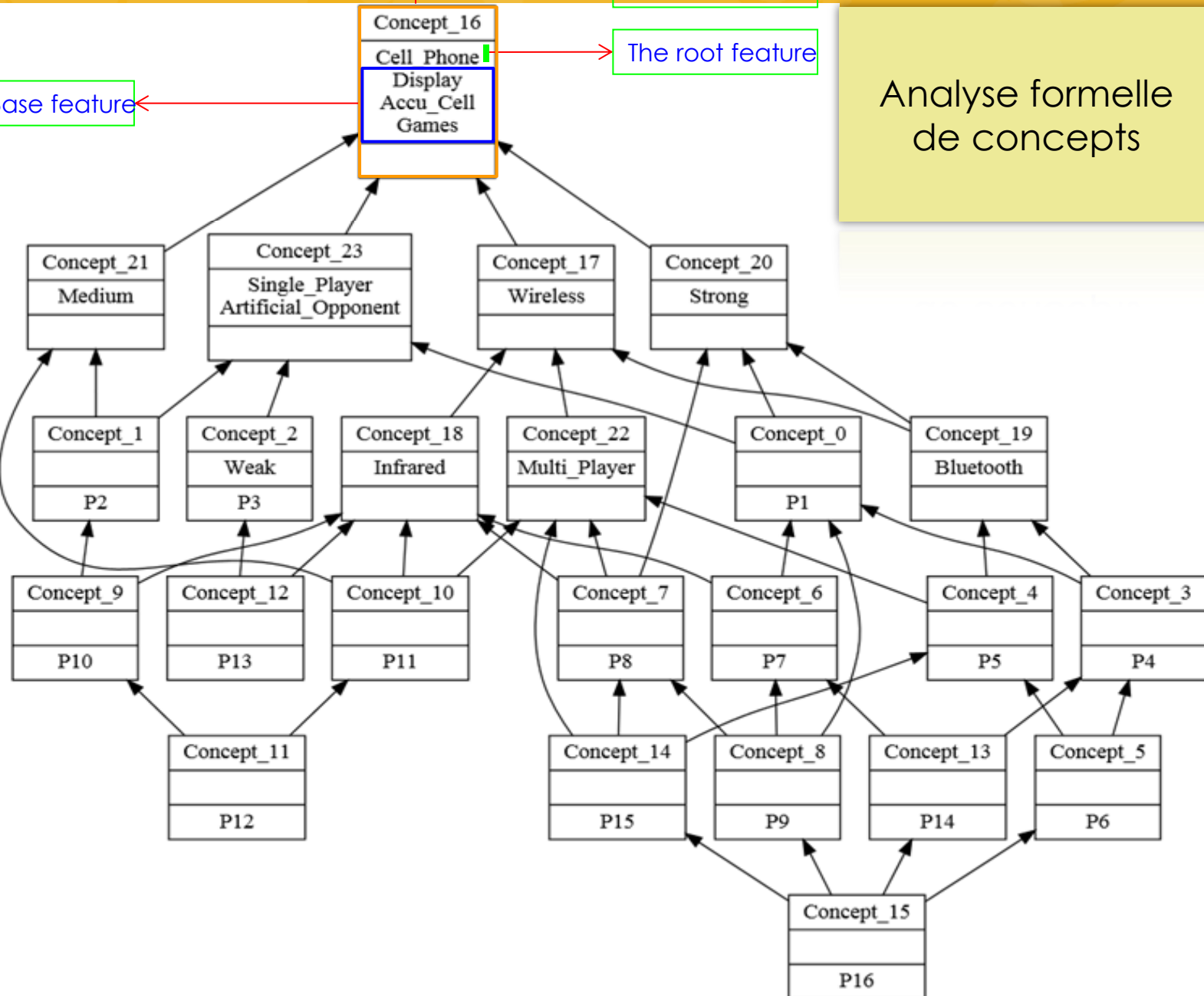
	Package(Editor.Managment)	Class (Close_Editor.Managment)	Class (Open_Editor.Managment)	Class (Print_Editor.Managment)	Package(Editor.Copy)	Class (CopyText_Editor.Copy)	Method (CopySettings_CopyText)	Package(Editor.SelectAll)	Class (SelectAllSettings_SelectAll)	Package(Editor.Paste)	Class (PasteText_Editor.Paste)	Method (PasteSettings_PasteText)
Editor_1	X	X	X	X								
Editor_2	X	X	X	X				X	X			
Editor_3	X	X	X	X	X	X	X			X	X	X

The Base feature

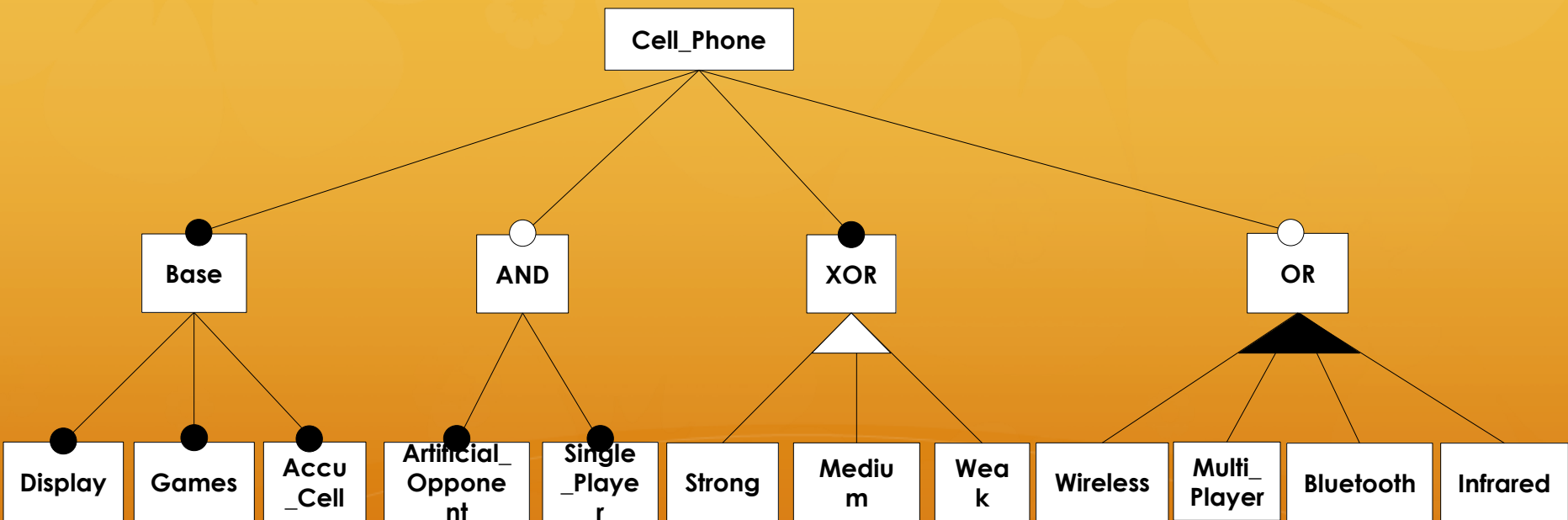
The top concept

The root feature

Analyse formelle
de concepts



Rétro ingénierie des lignes de produits



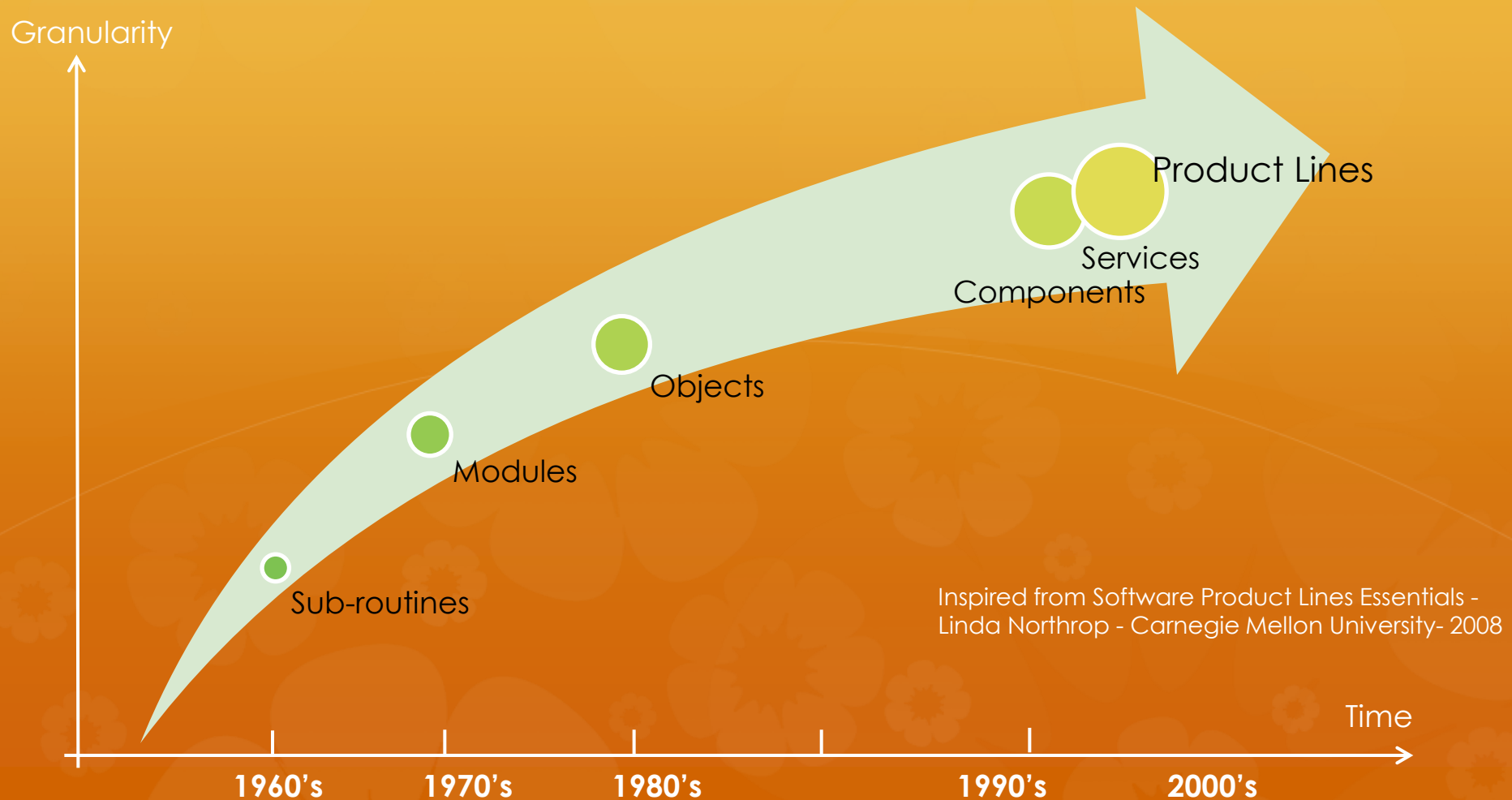
Weak \Rightarrow Single_Player
 Weak \Rightarrow Artificial_Opponent
 Infrared \Rightarrow Wireless
 Bluetooth \Rightarrow Wireless
 Bluetooth \Rightarrow Strong
 Multi_Player \Rightarrow Wireless

\neg (Bluetooth \wedge Medium)
 \neg (Multi_Player \wedge Weak)
 \neg (Bluetooth \wedge Weak)

Rétro ingénierie des lignes de produits

Case Study	Feature		Evaluation Metrics			
Mobile Media Features	Common	Optional	K	Precision	Recall	F-Measure
Album Management	×		0.05	83%	62%	70%
Splash Screen	×		0.05	71%	57%	63%
Create Album	×		0.05	81%	58%	67%
Delete Album	×		0.05	80%	62%	69%
Create Photo	×		0.05	81%	52%	63%
Delete Photo	×		0.05	78%	63%	69%
View Photo	×		0.05	87%	68%	76%
Exception handling		×	0.03	100%	70%	82%
Edit Photo Label		×	0.02	100%	77%	87%
Favourites		×	0.04	100%	80%	88%
Sorting		×	0.06	100%	78%	87%
ArgoUML Features	Common	Optional	K	Precision	Recall	F-Measure
Class Diagram	×		0.03	72%	56%	63%
Diagram		×	0.06	100%	80%	88%
Deployment Diagram		×	0.05	100%	74%	85%
Collaboration Diagram		×	0.06	100%	67%	80%
Use Case Diagram		×	0.03	100%	64%	78%
State Diagram		×	0.03	100%	69%	81%
Sequence Diagram		×	0.02	100%	67%	80%
Activity Diagram		×	0.06	100%	63%	77%
Cognitive Support		×	0.01	100%	70%	82%
Logging		×	0.02	100%	60%	75%

Architecture à base de composants

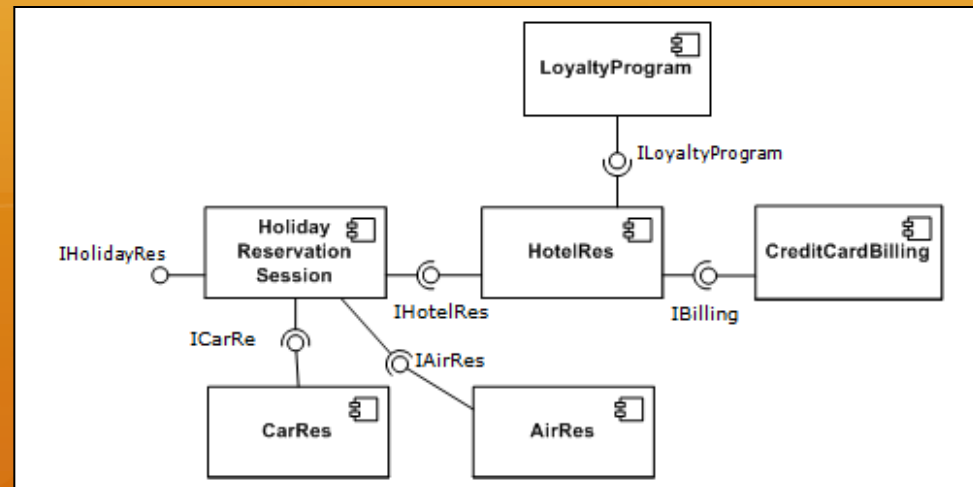
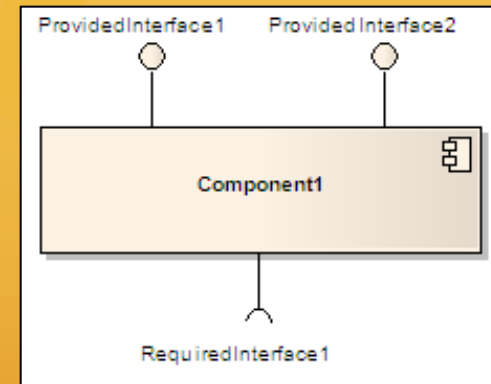


Architecture à base de composants

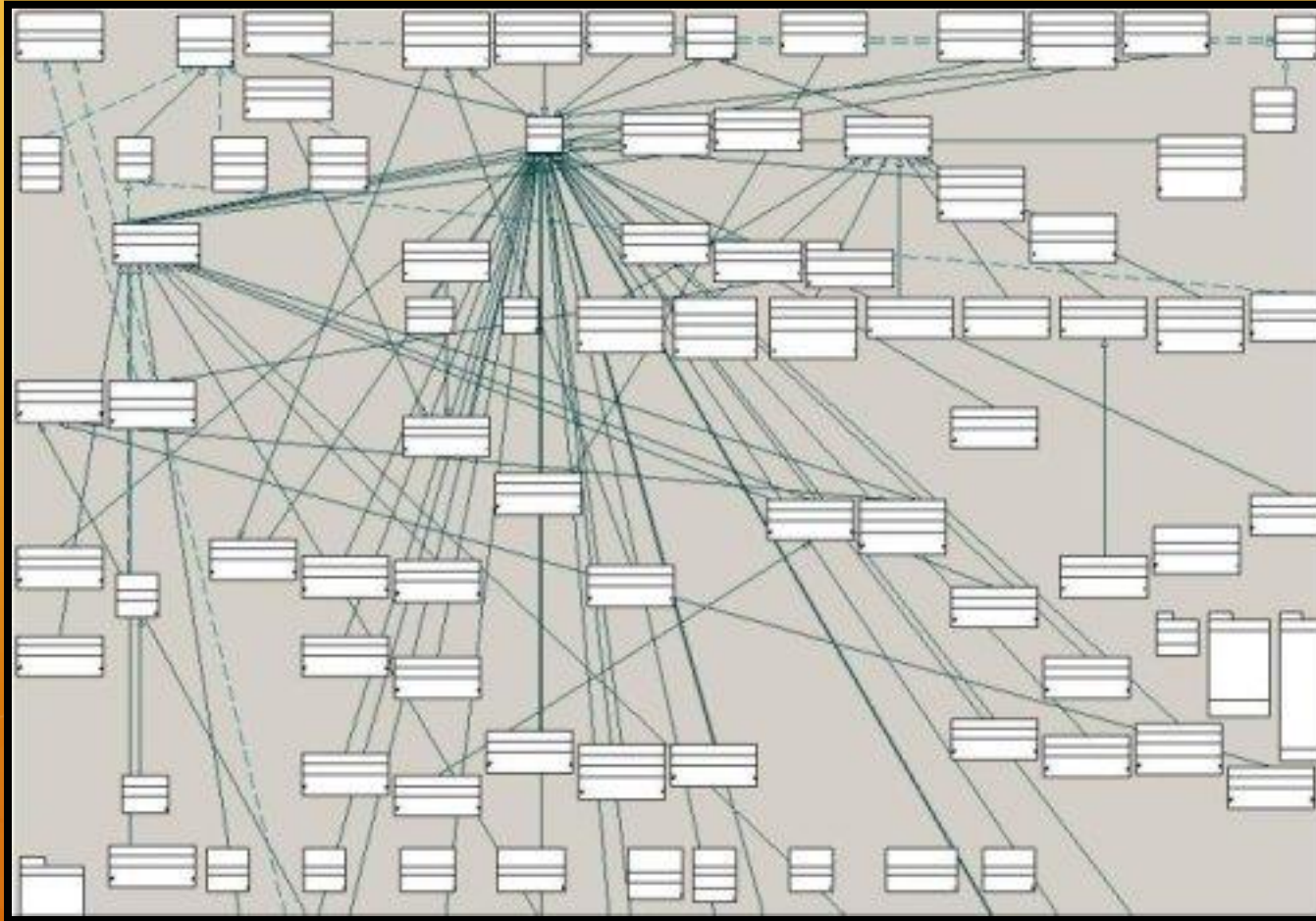
- ❁ Composant
 - ❁ Communication via des interfaces
 - ❁ Explicitation des dépendances

- ❁ Architecture à base de composants

- ❁ Langages
 - ❁ de description d'architecture à base de composants
 - ❁ de programmation à base de composants

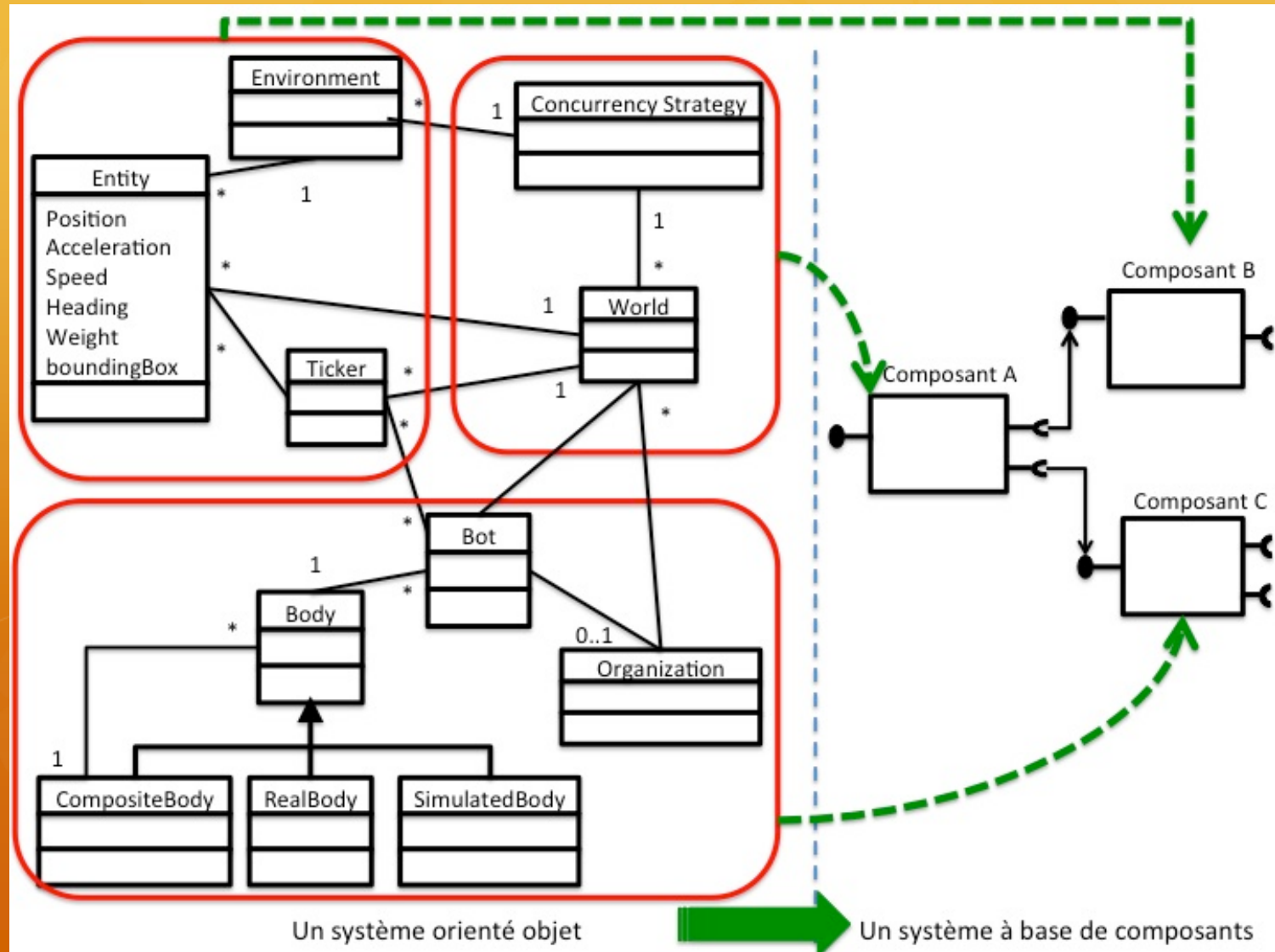


Migration vers des architectures à base de composants



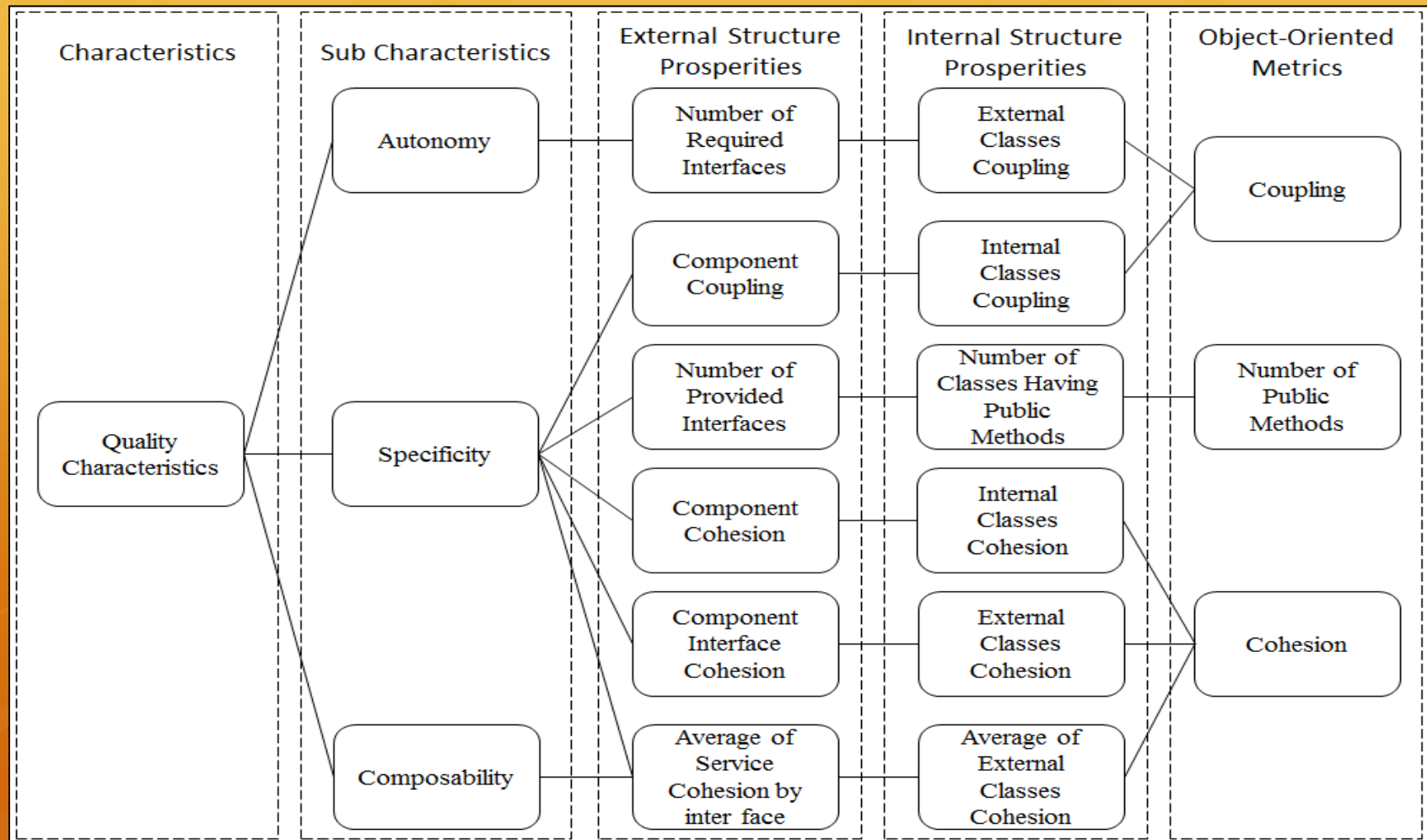
Applications orientées objets

Migration vers des architectures à base de composants



Migration vers des architectures à base de composants

- ❁ Fonction objectif
 - ❁ Maximiser la qualité des composants
 - ❁ Composabilité
 - ❁ Autonomie
 - ❁ Fonctionnalité
 - ❁ Maintenabilité
 - ❁ Fiabilité



Migration vers des architectures à base de composants

- Composabilité

$$C(E) = \frac{1}{|B|} \cdot \sum_{i \in B} LCC(i)$$

- Autonomie

$$A(E) = \text{couplageExt}(E) = 100 - \text{Couplage}(E)$$

- Spécificité

$$Spe(E) = \frac{1}{5} \cdot ((\frac{1}{|B|} \cdot \sum_{i \in B} LCC(i)) + LCC(B) + LCC(E) + Coup(E))$$

- Fonction objectif

$$S(E) = \sum_i \frac{1}{\lambda_i} (\lambda_1 \cdot C(E) + \lambda_2 \cdot A(E) + \lambda_3 \cdot Spe(E))$$

Migration vers des architectures à base de composants

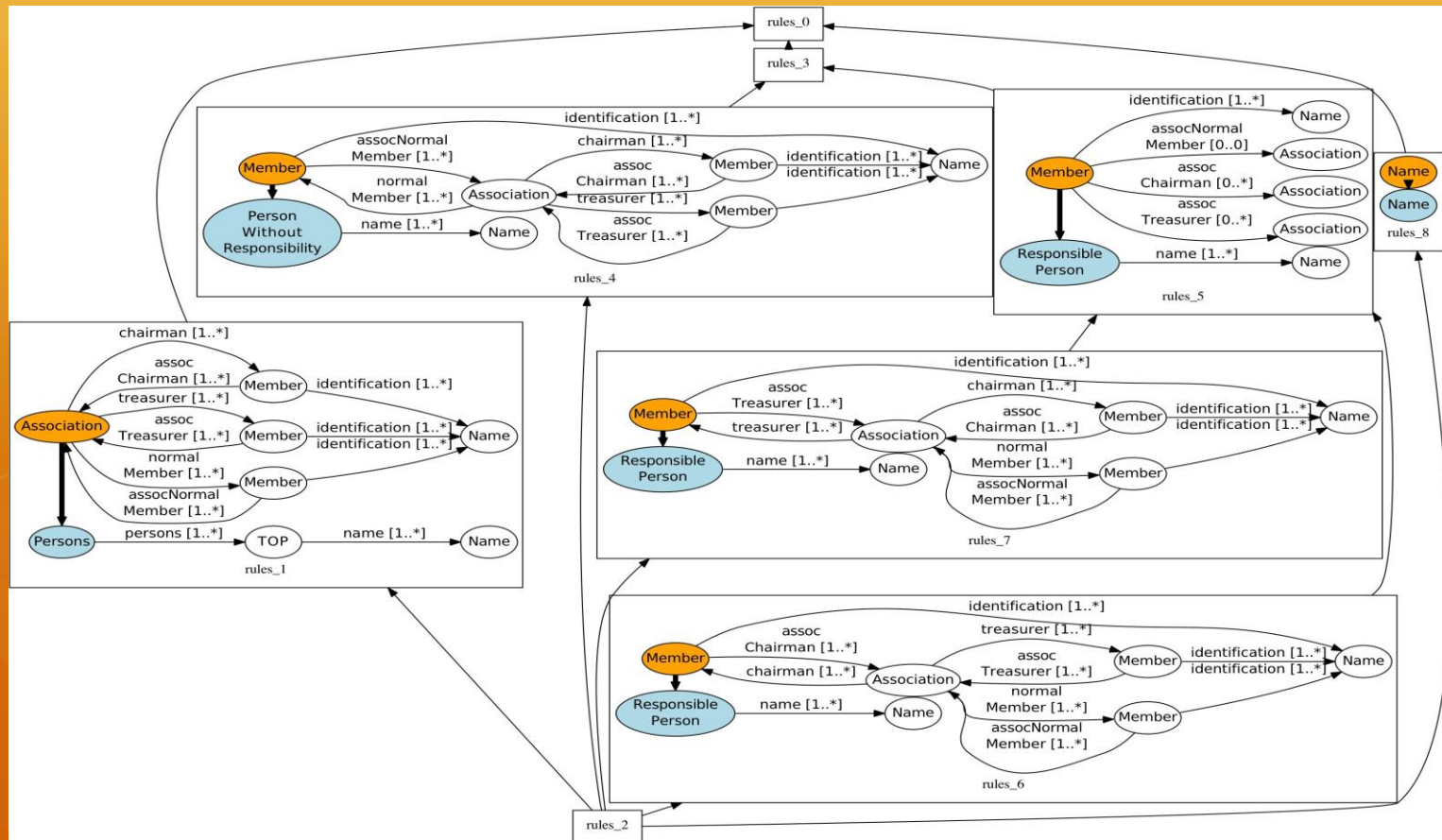
❁ Algorithmes

- ❁ Regroupement (Clustering) hiérarchique
- ❁ Génétique
- ❁ Recuit simulé

❁ Exemples

- ❁ ArgoUML : +1500 classes → 13 composants (premier niveau)
 - ❁ Composant «génération de code
 - ❁ Composant «gestion de diagrammes"
 - ❁ Composant «IU»
 - ❁ Etc.
- ❁ Jigsaw : + 300 classes → 8 composants

Generation of model transformation patterns



COMPO : a reflexive component-based language

All handled entities are components

```

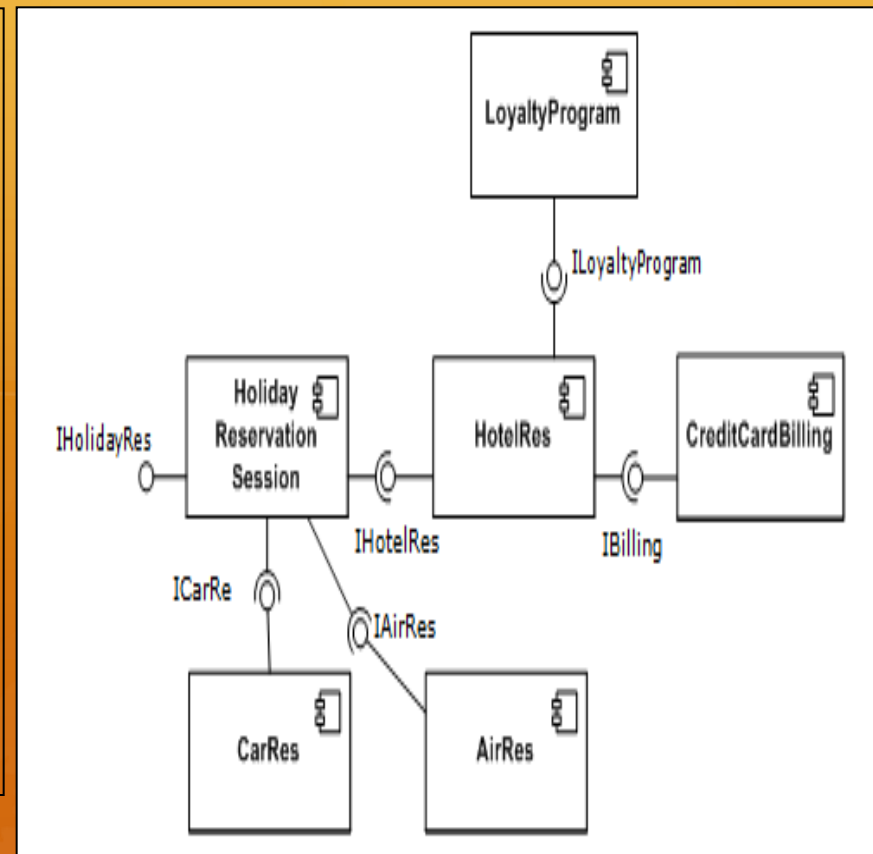
public component class WebServer {

  private final owned Router r = new Router();
  connect r.request, create;
  connect pattern Router.workers, Worker.serve;

  public void run() { r.listen(); }

  private port create {
    provides r.workers requestWorker() {
      final owned Worker newWorker = new Worker();
      r.workers connection = connect(r.workers,
        newWorker.serve);
      return connection;
    }
  }
  ...
}

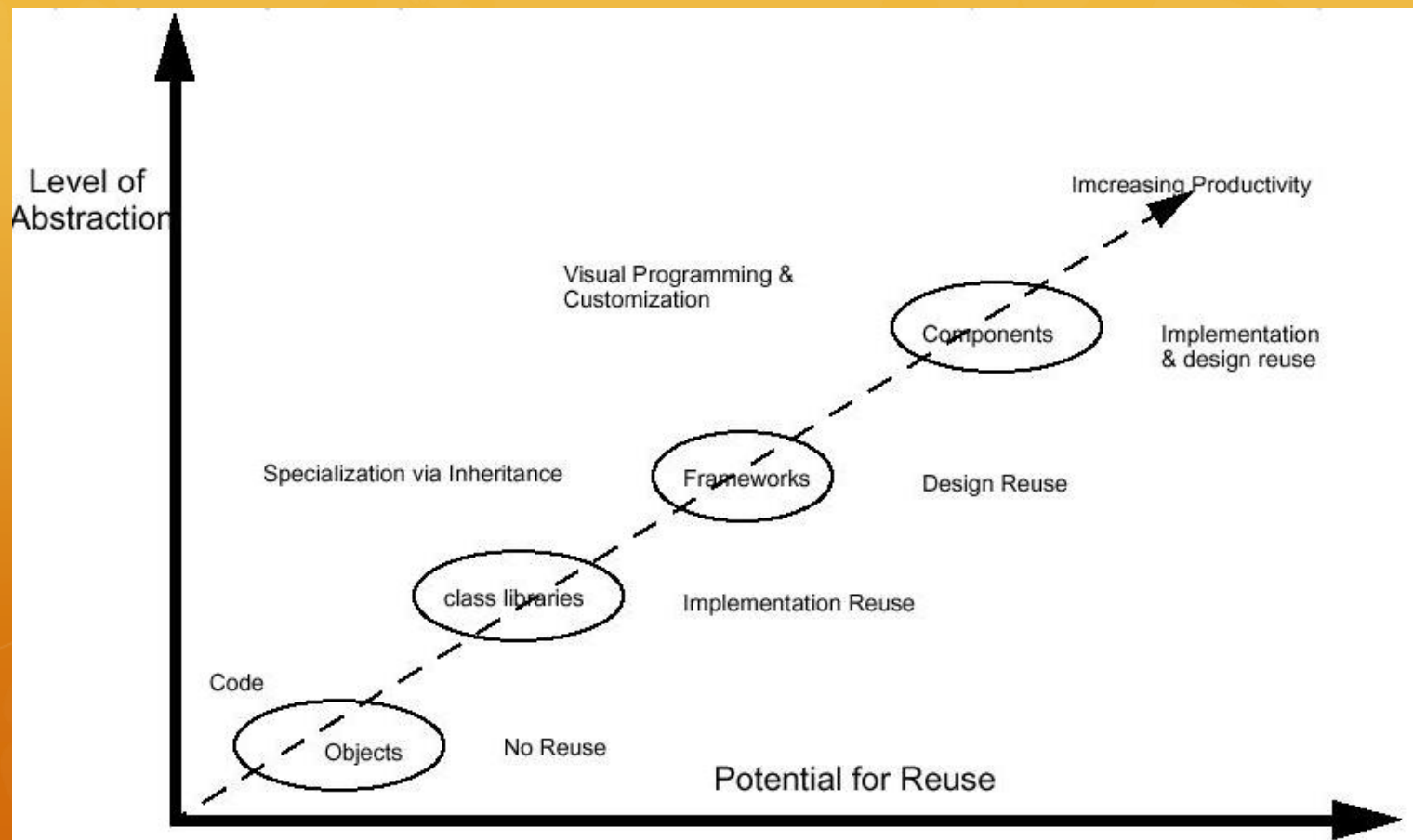
```



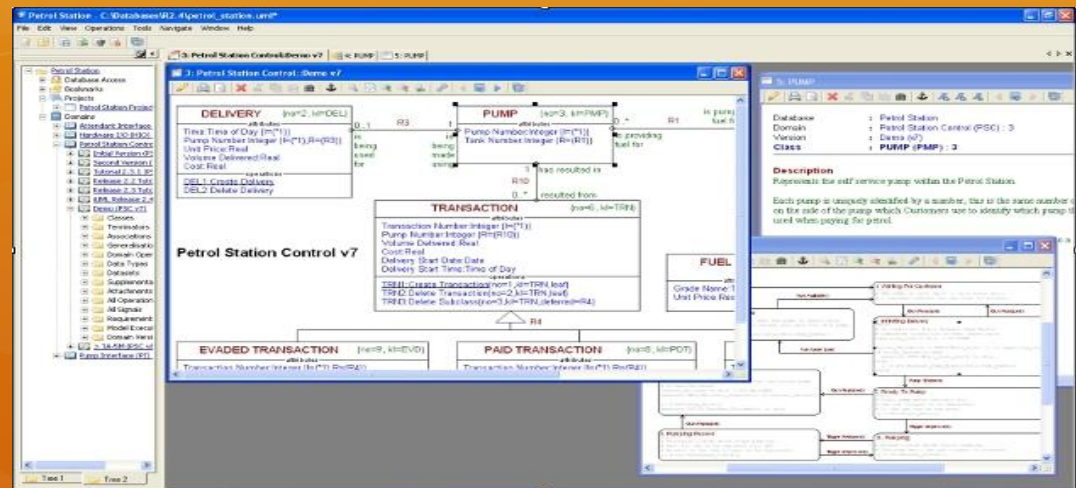
Projection vers domaine de l'embarqué

- ❁ Lignes de produits pour logiciels embarqués
 - ❁ Conception
 - ❁ Retro-conception et Réingénierie
 - ❁ Reconfiguration dynamique
- ❁ Architecture logicielle embarquée
 - ❁ Co-Conception architecture logicielle/matérielle
 - ❁ Coévolution architecture logicielle/matérielle
 - ❁ Architecture pour objets connectés
- ❁ Développement par Réutilisation
 - ❁ Modèles et langages pour composants logiciels embarqués
 - ❁ Identification de composants logiciels embarqués
- ❁ Compréhension et migration des logiciels embarqués
 - ❁ Retro-conception de l'architecture logicielle
- ❁ Etc.

Merci



Successive steps of derivation



Executable models (programs in changing technologies)