

Modélisation UML

Principes de modélisation

Plan du cours

- Principes de base de la modélisation conceptuelle
- Définitions de base

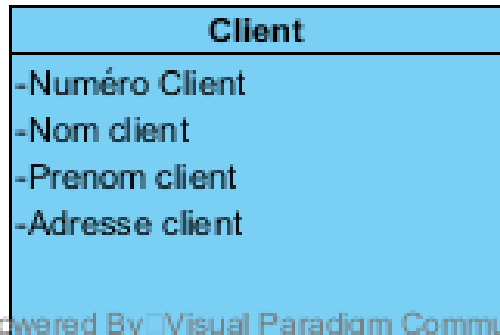
Le modèle conceptuel de données

- Définition (source: G Roy)

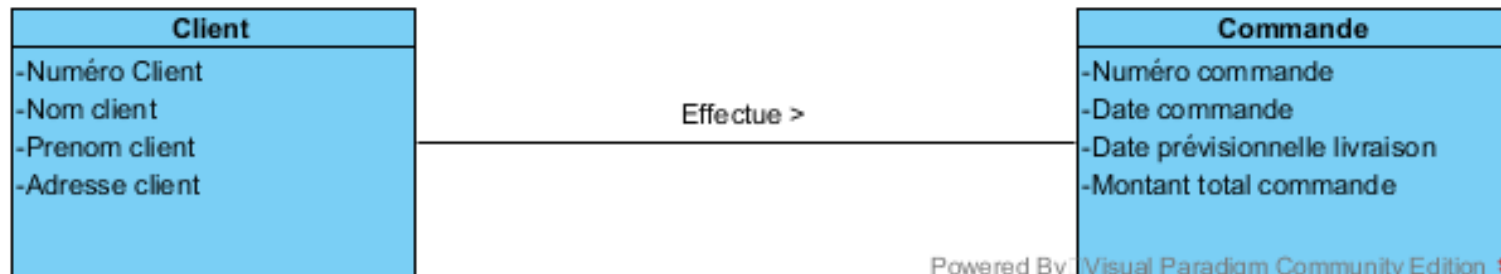
Un modèle conceptuel de données est une représentation des besoins en matière de données pour un système d'information. Il met en évidence les **entités**, leurs **attributs**, les **associations** et **contraintes** entre ces entités pour un domaine donné.

- **Entité**: Objet concret ou abstrait du monde réel au sujet duquel une organisation est susceptible de conserver des données (*Entity*).
- **Attribut**: Donnée élémentaire qui sert à caractériser une propriété des entités et des associations dans un modèle conceptuel de données (*Attribute*).
- **Association**: Lien sémantique qui existe entre deux entités ou plus. Elle représente souvent la mémoire d'un événement qui a permis d'établir un lien logique entre ces entités (*Relationship*).

Exemple: relation d'association entre 2 entités Client et Commande



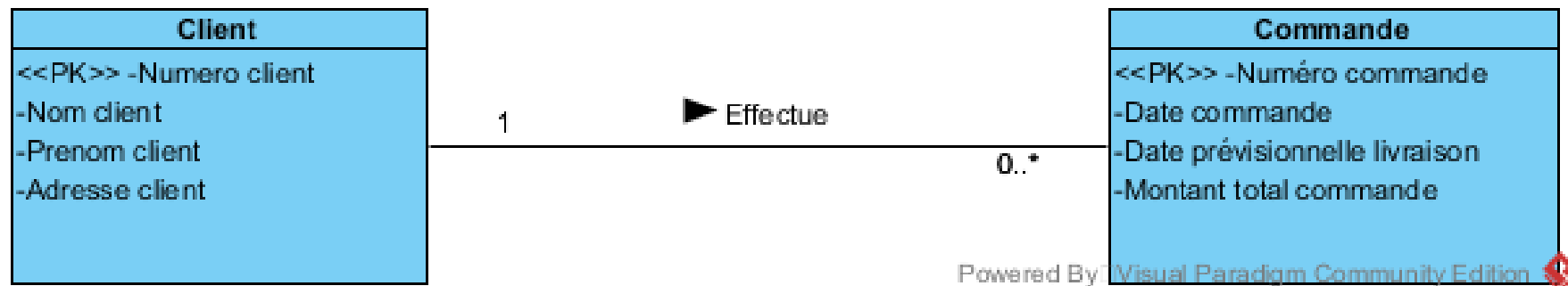
Powered By Visual Paradigm Community



Powered By Visual Paradigm Community Edition

Association entre l'entité Client et l'entité Commande

Exemple: indication des multiplicités et identifiants



Multiplicité:

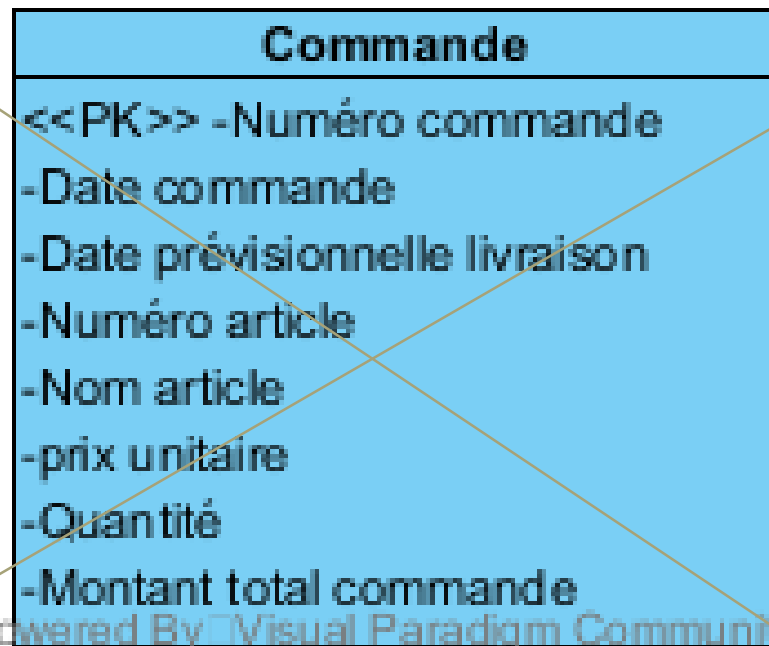
Contrainte inscrite à chaque extrémité d'une association binaire comportant un couple de valeurs (minimum–maximum) qui établit, pour chaque entité de l'association, les nombres minimum et maximum d'occurrences de l'autre entité qui peuvent lui être associées (*Multiplicity*).

Règles de base

- Contrainte de domaine et monovaluation
- Chaque attribut d'une entité présente dans le modèle conceptuel ne peut prendre qu'une seule valeur à la fois pour son occurrence. Cette valeur doit provenir du domaine décrit par les contraintes de domaine de l'attribut.
- Contrainte d'unicité – non redondance
- Chaque attribut du modèle conceptuel est unique, il ne peut y apparaître plus d'une fois à travers plusieurs entités du modèle.

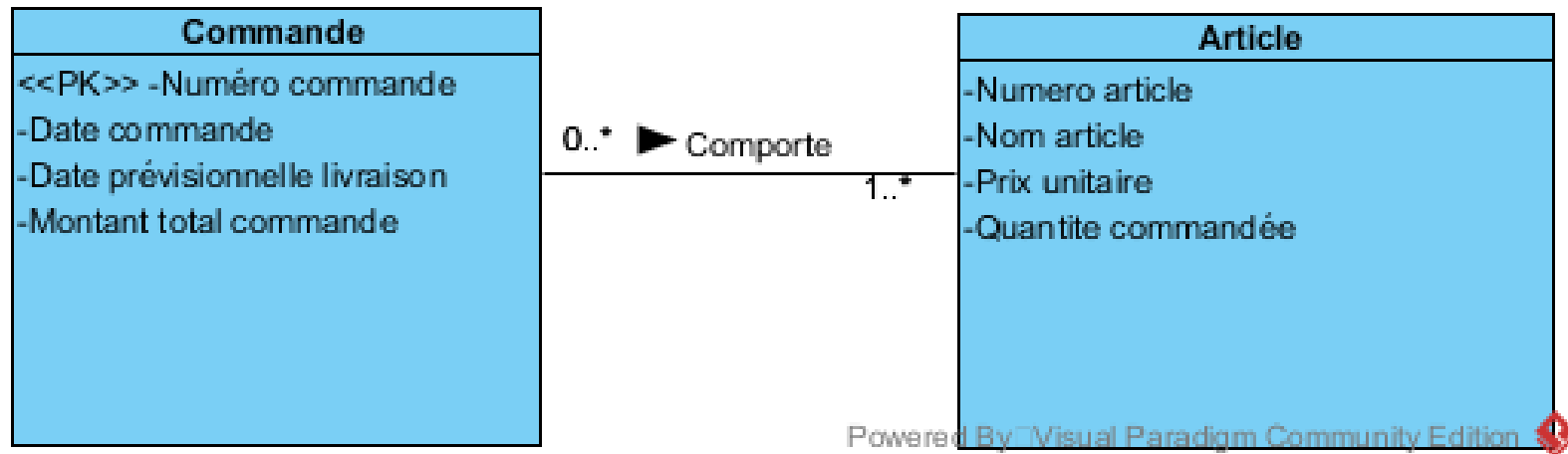
Règles de construction

- Séparation des entités
- Une entité ne peut en contenir une autre. En particulier, une entité ne peut contenir un attribut d'une autre entité.



Powered By Visual Paradigm Community Edition

Séparation des entités avec une relation d'association



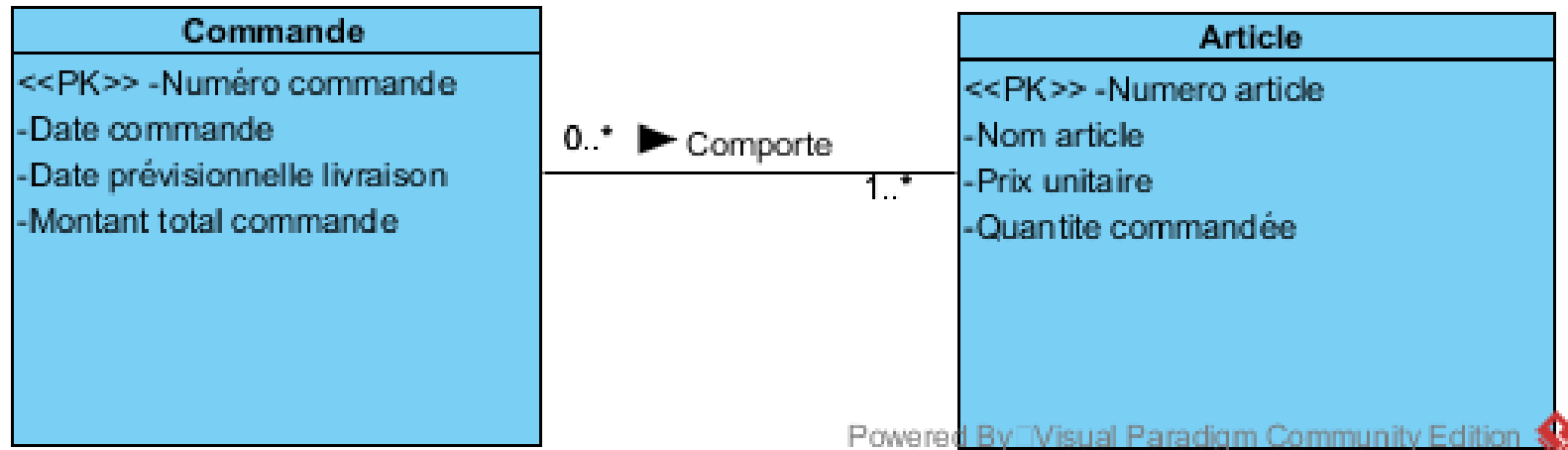
Séparation nette entre les attributs qui décrivent une commande et ceux qui décrivent un article.

Utilisation d'une relation d'association qui fait le lien entre les 2 entités.

Critique de la solution

- Concept de dépendance fonctionnelle
- On dit qu'il y a **dépendance fonctionnelle** si l'ensemble des attributs d'une entité dépend de l'identifiant (la clef primaire).
- Si il n'y a pas d'indépendance fonctionnelle d'un ou plusieurs attributs, on doit s'interroger si ces attributs ne dépendent pas plutôt de la relation

Exemple



Nom article dépend de Numéro article → Règle de dépendance fonctionnelle respectée

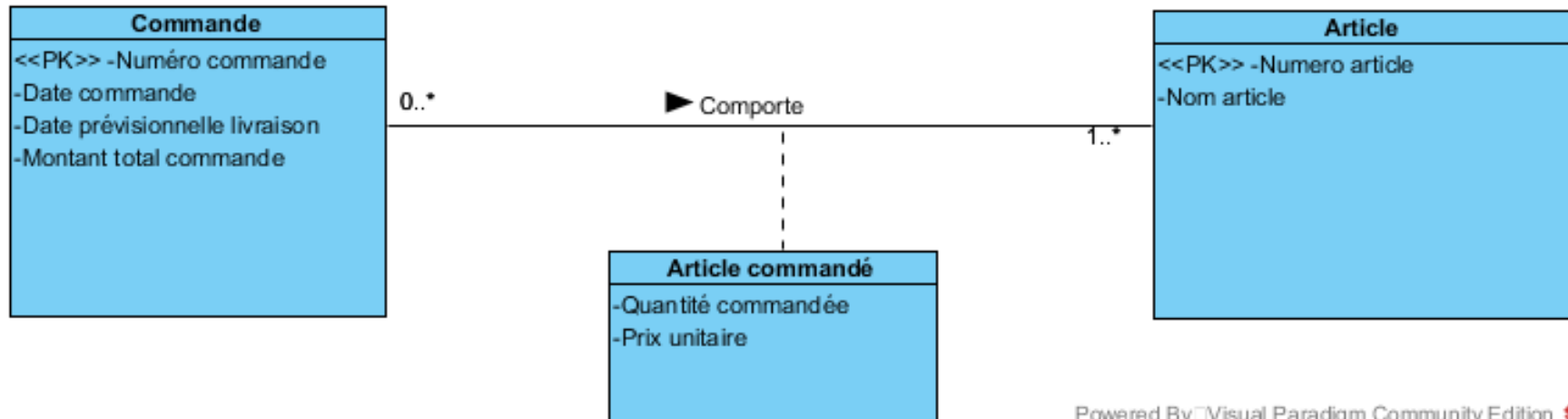
En revanche « Prix unitaire » et « Quantité commandée » ne dépendent pas du numéro d'article, mais de la relation entre Commande et Article

Ces attributs doivent être modélisés comme des attributs de la relation et faire partie de ce qu'on appelle une *Entité faible*

Entité faible

- Définition: G Roy

Type d'entité dont l'existence dépend de deux ou plusieurs entités. Son identifiant se définit en fonction des identifiants des entités dont elle dépend. Elle ne possède pas d'attribut qui puisse servir d'identifiant et qui lui appartienne en propre



Exercices – G Roy

- Cas n°1
- Créer avec le formalisme UML un modèle conceptuel de données qui illustre les besoins métier suivant:
 - 1) Chaque société contrôle des départements ;
 - 2) Chaque département ne peut appartenir qu'à une seule société
 - 3) Chaque département emploie un ou plusieurs employés
 - 4) Chaque employé travaille pour un seul département
 - 5) Un employé peut ne pas avoir de subalternes rattachés ou en avoir plusieurs
 - 6) Chaque subalterne est rattaché à un seul chef employé