

Modèle et DTD

Jean-Baptiste Camps & Simon Gabay

Modéliser son projet

- Modèle des données
- C'est l'analyse et la conception de l'information contenue dans le système.
- Les valeurs des données individuelles observées sont ignorées au profit de la structure, des relations, des noms et des formats des données pertinentes
- Le modèle de données ne doit pas seulement définir la structure de données, mais aussi ce que les données veulent vraiment signifier (sémantique).

Modéliser son projet

- Modèle des données
 - Des données
- Indiana Jones
- Raiders of the Lost Ark
- Steven Spielberg
- Harrison Ford
- Karen Allen

Modéliser son projet

- Modèle des données
 - Des données
 - Indiana Jones→personnage
 - Raiders of the Lost Ark→titre
 - Steven Spielberg→réalisateur
 - Harrison Ford→acteur
 - Karen Allen→actrice

Modéliser son projet

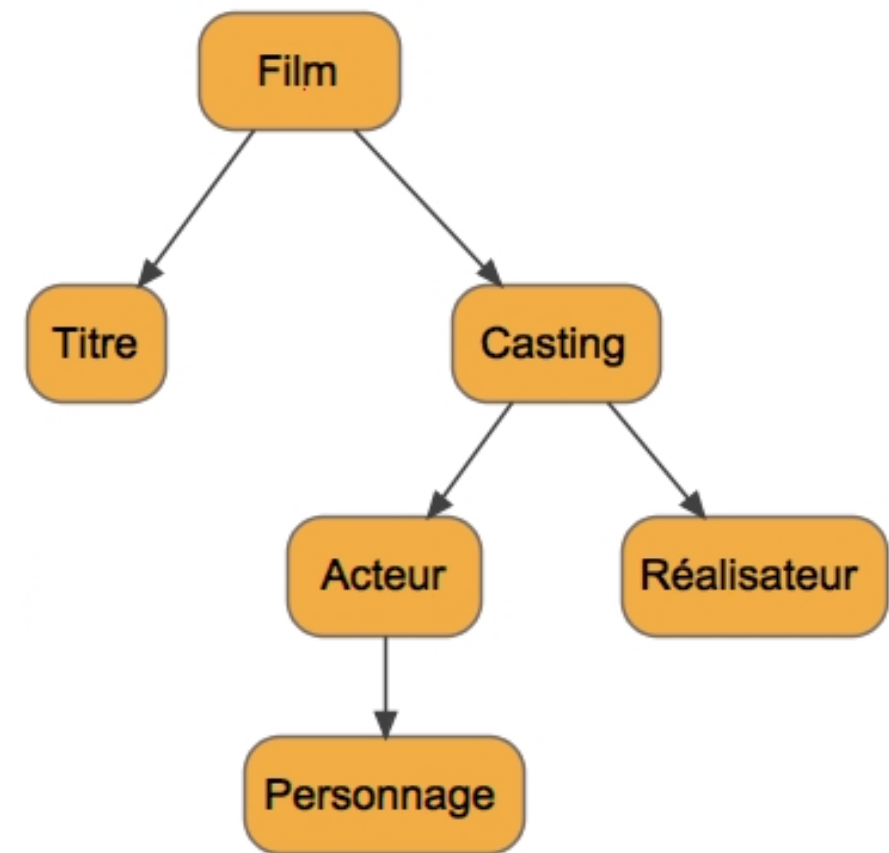
- Modèle des données
 - Des données
 - Une structuration
- Quel est le rapport entre ces données ?
 - C'est un film
- Un acteur est-il la même chose qu'une actrice ?
 - Oui... et non
- Y'a-t-il un rapport entre un acteur et un réalisateur ?
 - Un peu, ils sont membres du casting
- Le prénom et le nom sont-ils deux informations différentes ?

Modéliser son projet

- Modèle des données
 - Des données
 - Une structuration
 - Une exploitation
- Quel type de données doit-on encoder ?
- (D'autres personnes ont-elles déjà eu affaire à ce type de données ?)
- Quel volume de données doit-on encoder ?
- Que veut-on faire de ces données ?
- Quelle est la meilleure solution informatique ?

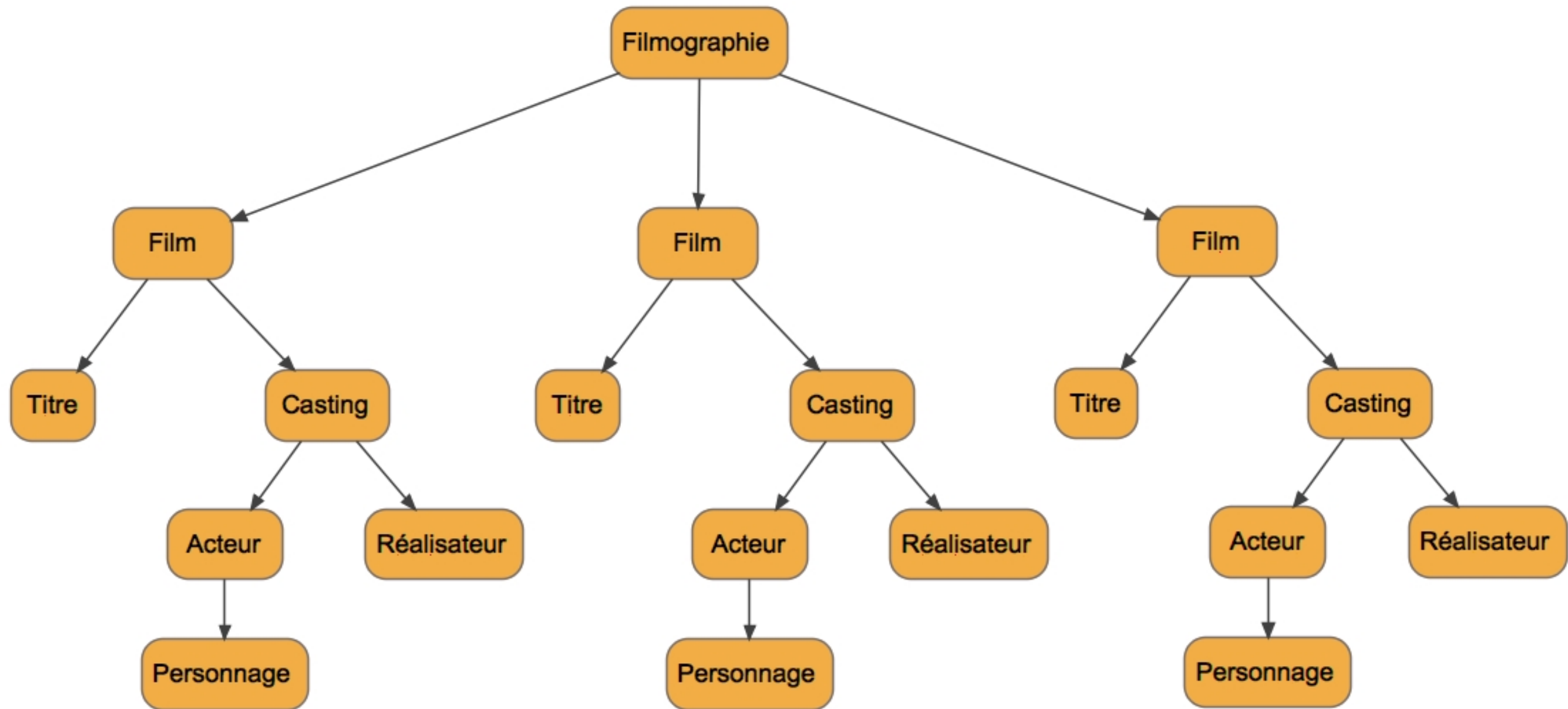
Modéliser son projet

- Un document XML?



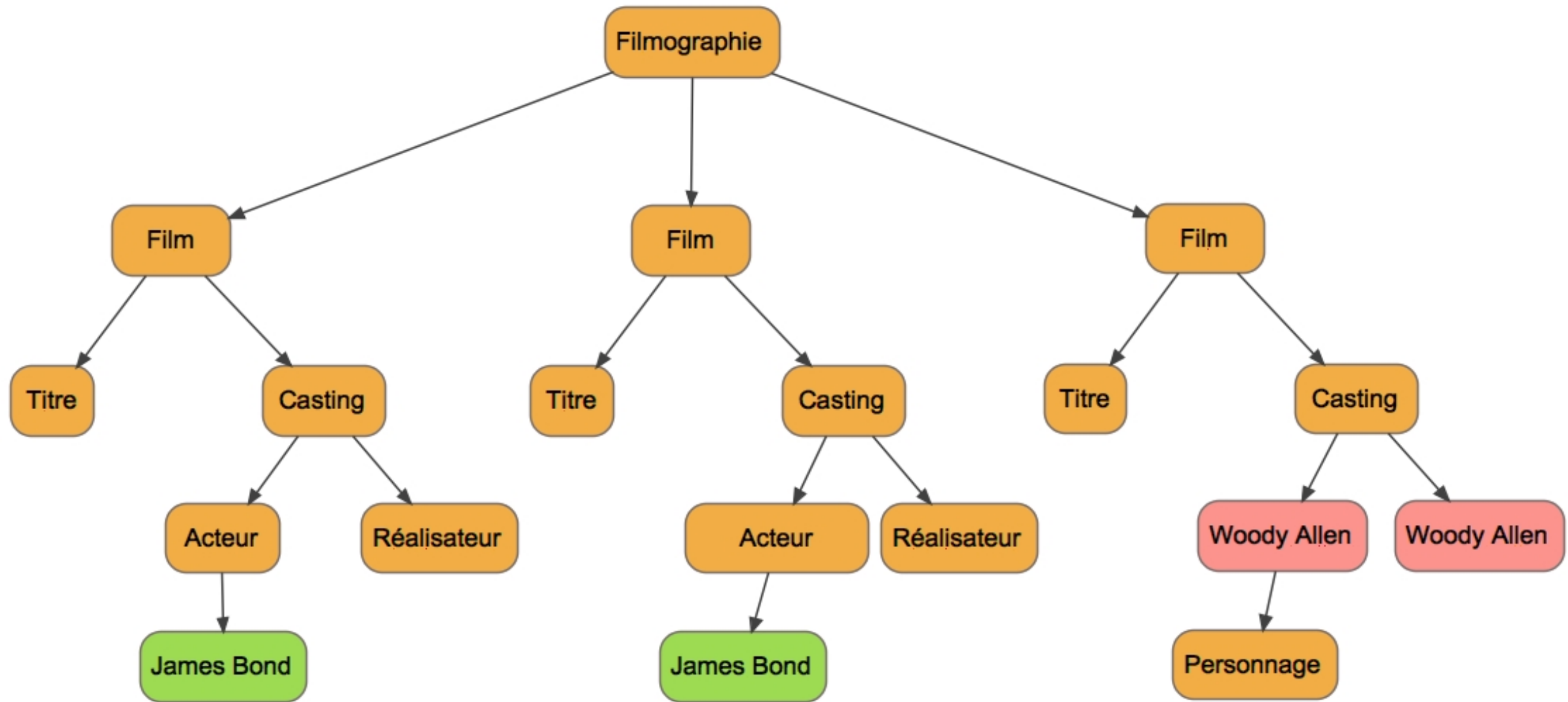
Modéliser son projet

- Un document XML?



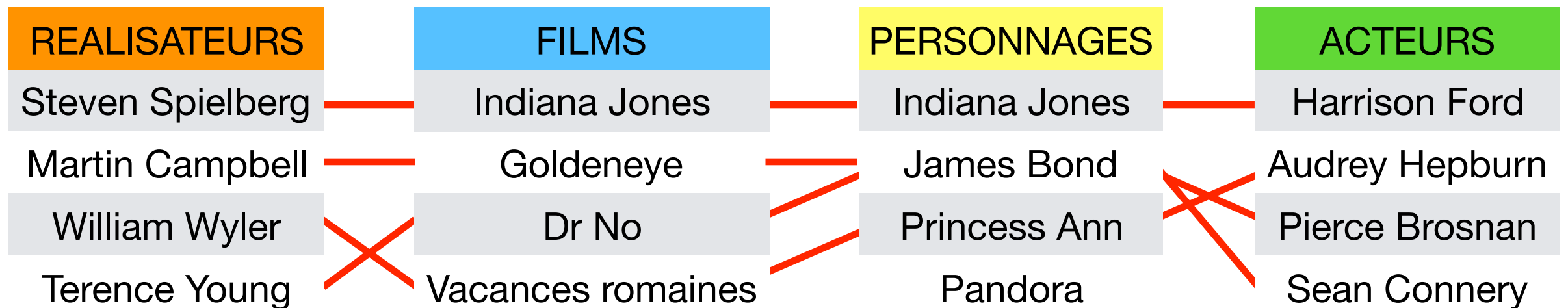
Modéliser son projet

- Un document XML?



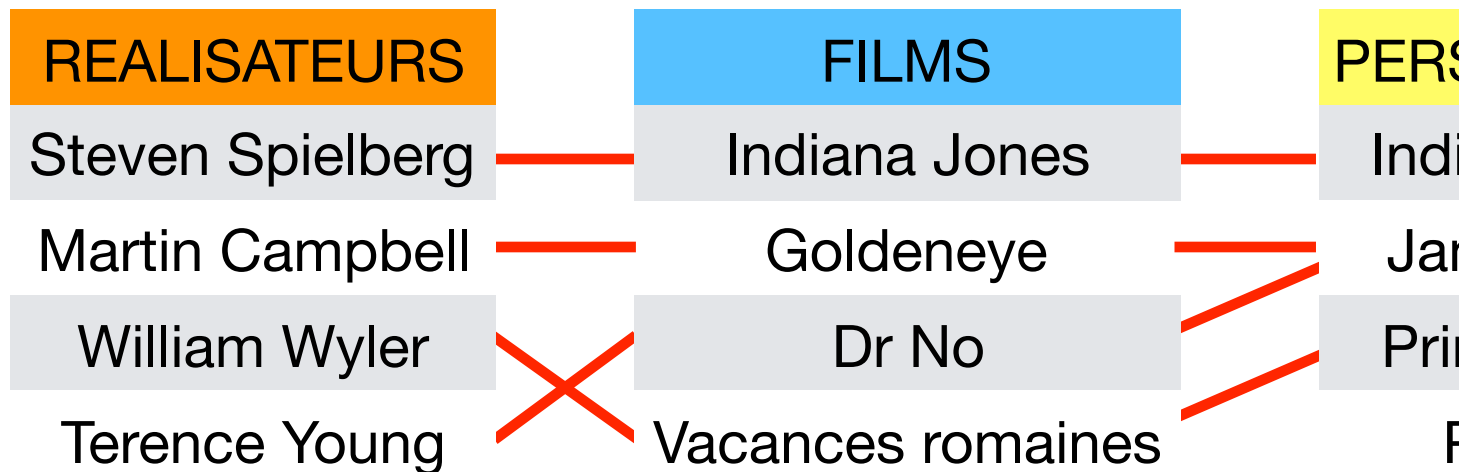
Modéliser son projet

- Un document XML?
- Une base de données relationnelle?



Modéliser son projet

- Un document XML?
- Une base de données relationnelle?



```

1  -----
2  --CREATION DE LA TABLE DES FILMS
3  -----
4  CREATE TABLE films (
5      id INT(11) NOT NULL AUTO_INCREMENT,
6      title VARCHAR(30) NOT NULL,
7      PRIMARY KEY (id)
8  );
9  -----
10 --On remplit la table des films
11 -----
12 INSERT INTO films (title)
13     VALUES ('Indiana Jones');
14 INSERT INTO films (title)
15     VALUES ('Goldeneye');
16 INSERT INTO films (title)
17     VALUES ('Dr No');
18 INSERT INTO films (title)
19     VALUES ('Vacances romaines');
20 -----
21 --CREATION DE LA TABLE DES RALISATEURS
22 -----
23 CREATE TABLE directors (
24     id INT(11) NOT NULL AUTO_INCREMENT,
25     name VARCHAR(30) NOT NULL,
26     film_id INT(11) NOT NULL,
27     INDEX (film_id)
28     PRIMARY KEY (id)
29 );
30 -----
31 --On remplit la table des films
32 --et on la relie à celle des réalisateurs
33 -----
34 INSERT INTO directors (title, film_id)
35     VALUES ('Steven Spielberg', 1);
36 INSERT INTO directors (title, film_id)
37     VALUES ('Martin Campbell', 2);
38 INSERT INTO directors (title, film_id)
39     VALUES ('William Wyler', 4);
40 INSERT INTO directors (title, film_id)
41     VALUES ('Terence Young', 3);
  
```

Modéliser son projet

- Un document XML?
- Une base de données relationnelle?

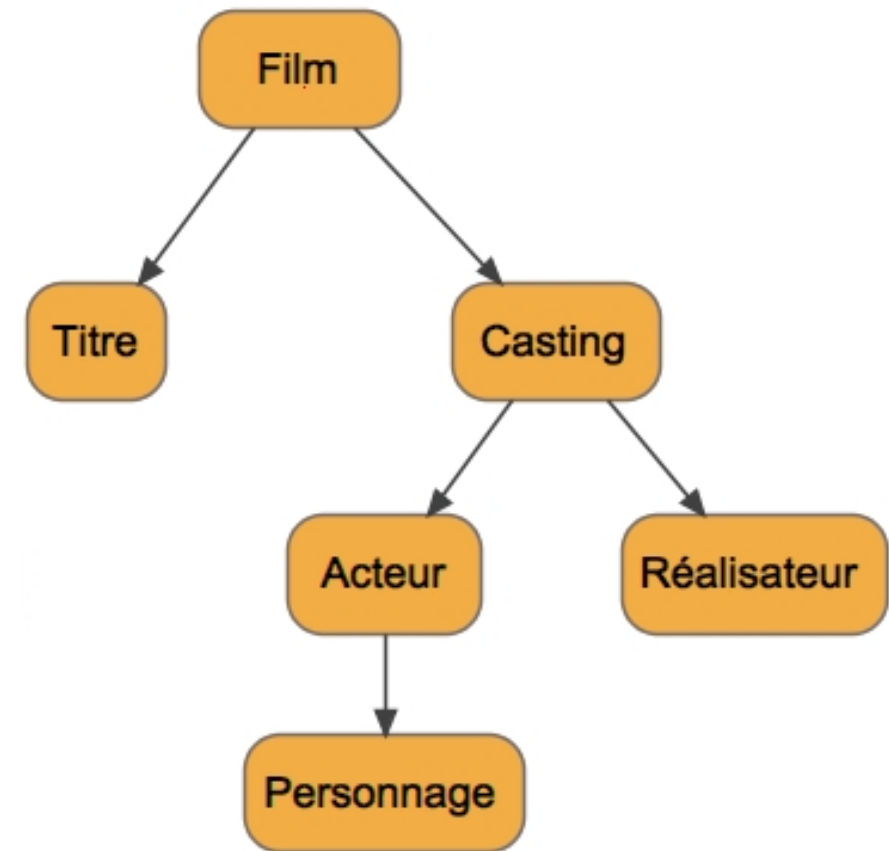
```
<film>
  <title lang="en">Raiders of the Lost Ark</title>
  <title lang="fr">Aventuriers de l'Arche</title>
  <casting>
    <director>Steven Spielberg</director>
    <actor gender="m">
      Harrison Ford
    <character>Indiana Jones</character>
  </actor>
    <actor gender="f">
      Karen Allen
    <character>M. Ravenwood</character>
  </actor>
</casting>
</film>
```

```
1  -----
2  --CREATION DE LA TABLE DES FILMS
3  -----
4  CREATE TABLE films (
5      id INT(11) NOT NULL AUTO_INCREMENT,
6      title VARCHAR(30) NOT NULL,
7      PRIMARY KEY (id)
8  );
9  -----
10 --On remplit la table des films
11 -----
12 INSERT INTO films (title)
13     VALUES ('Indiana Jones');
14 INSERT INTO films (title)
15     VALUES ('Goldeneye');
16 INSERT INTO films (title)
17     VALUES ('Dr No');
18 INSERT INTO films (title)
19     VALUES ('Vacances romaines');
20 -----
21 --CREATION DE LA TABLE DES REALISATEURS
22 -----
23 CREATE TABLE directors (
24     id INT(11) NOT NULL AUTO_INCREMENT,
25     name VARCHAR(30) NOT NULL,
26     film_id INT(11) NOT NULL,
27     INDEX (film_id)
28     PRIMARY KEY (id)
29 );
30 -----
31 --On remplit la table des films
32 --et on la relie à celle des réalisateurs
33 -----
34 INSERT INTO directors (title, film_id)
35     VALUES ('Steven Spielberg', 1);
36 INSERT INTO directors (title, film_id)
37     VALUES ('Martin Campbell', 2);
38 INSERT INTO directors (title, film_id)
39     VALUES ('William Wyler', 4);
40 INSERT INTO directors (title, film_id)
41     VALUES ('Terence Young', 3);
```

Modéliser son projet

- Un document XML?
- Une base de données relationnelle?

```
<film>
  <title lang="en">Raiders of the Lost Ark</title>
  <title lang="fr">Aventuriers de l'Arche</title>
  <casting>
    <director>Steven Spielberg</director>
    <actor gender="m">
      Harrison Ford
      <character>Indiana Jones</character>
    </actor>
    <actor gender="f">
      Karen Allen
      <character>M. Ravenwood</character>
    </actor>
  </casting>
</film>
```



Modéliser son projet

- Modèle des données
- Vocabulaire et grammaire
- Éléments : `<film>`, `<actor>`...
- Attributs : `<actor>` a un `@gender` "f" ou "m"
- Relations :

```
<film>
  <title lang="en">Raiders of the Lost Ark</title>
  <title lang="fr">Aventuriers de l'Arche</title>
  <casting>
    <director>Steven Spielberg</director>
    <actor gender="m">
      Harrison Ford
      <character>Indiana Jones</character>
    </actor>
    <actor gender="f">
      Karen Allen
      <character>M. Ravenwood</character>
    </actor>
  </casting>
</film>
```

- `<actor>` \subset `<casting>`
- `<title>` $\not\subset$ `<casting>`
- `<casting>` $\not\subset$ `<title>`
- `<title>` et `<casting>` \subset `<film>`

→ C'est le schéma. Comment informatiser cette description ?

Modéliser son projet

- Modèle des données
 - Vocabulaire et grammaire
 - Mise en œuvre
- La grammaire et le vocabulaire d'un document XML sont décrits dans un langage de définition de données (*Data definition language*) qui peut être
 - un schéma
 - une DTD (*Document Type Definition*)
 - Un ODD (*One Document Does it all*), spécifique à la TEI

DTD

- La DTD
 - Elle se trouve
 - Soit dans fichier séparé
 - Soit dans une partie d'un document XML (ou SGML)

DTD

- La DTD est
 - externe privée

- En accès en local (donc restreint) sur mon ordinateur ou sur un serveur

```
1. <?xml version="1.0" encoding="UTF-8"
    standalone="no"?>
2. <!DOCTYPE nomÉlémentRacine SYSTEM
    "madtd.dtd">
```

- Remarquez que le nom de l'élément racine se trouve dans la déclaration, et pas dans la DTD

DTD

- La DTD est
- externe privée

- Exemple

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE film SYSTEM "film.dtd">
<film>
  <title>Raiders of the Lost Ark</title>
  <casting>
    <director>Steven Spielberg</director>
    <actor> Harrison Ford</actor>
    <actor>Karen Allen</actor>
  </casting>
</film>
```

- Renvoi à la DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT film (title,casting)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT casting (director,actor+)>
<!ELEMENT director (firstName,lastName)>
<!ELEMENT firstName (#PCDATA)>
<!ELEMENT lastName (#PCDATA)>
<!ELEMENT actor (#PCDATA)>
```

DTD

- La DTD est
 - externe privée
 - externe publique

- accessible publiquement

- via un ISBN

```
<!DOCTYPE ead PUBLIC" +//ISBN  
1-931666-00-8//DTD ead.dtd (Encoded  
Archival Description (EAD) Version 2002)//  
EN" "../shared/ead/ead.dtd"
```

- via une URI

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML  
1.0 Strict//EN" "http://www.w3.org/TR/  
xhtml1/DTD/xhtml1-strict.dtd">
```

DTD

- La DTD est
 - externe privée
 - externe publique
 - interne

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<!DOCTYPE nomÉlémentRacine [
  <!ELEMENT nomÉlémentRacine (#PCDATA)>
  Définition des autres éléments...
```

- Exemple

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<!DOCTYPE film [
  <!ELEMENT film (title,casting)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT casting (director,actor+)>
  <!ELEMENT director (firstName,lastName)>
  <!ELEMENT firstName (#PCDATA)>
  <!ELEMENT lastName (#PCDATA)>
  <!ELEMENT actor (#PCDATA)>
]>
<film>
  <title>Raiders of the Lost Ark</title>
  <casting>
    <director>Steven Spielberg</director>
    <actor> Harrison Ford</actor>
    <actor>Karen Allen</actor>
  </casting>
</film>
```

DTD

- La DTD est
 - externe privée
 - externe publique
 - interne
- Externe avec sous-ensemble interne

```
<!DOCTYPE document SYSTEM "madtd.dtd" [  
    <!ELEMENT elementNonDecritDansmadtd.dtd  
    (#PCDATA)>  

```

- Exemple

```
<?xml version="1.0" encoding="UTF-8"  
standalone="no"?>  
<!DOCTYPE film SYSTEM "film.dtd" [  
<!ELEMENT actor (#PCDATA)>  
<film>  
    <title>Raiders of the Lost Ark</title>  
    <casting>  
        <director>Steven Spielberg</director>  
        <actor> Harrison Ford</actor>  
        <actor>Karen Allen</actor>  
    </casting>  
</film>
```

- Complète la DTD

```
<?xml version="1.0" encoding="UTF-8"?>  
<!ELEMENT film (title,casting)>  
<!ELEMENT title (#PCDATA)>  
<!ELEMENT casting (director,actor+)>  
<!ELEMENT director (firstName,lastName)>  
<!ELEMENT firstName (#PCDATA)>  
<!ELEMENT lastName (#PCDATA)>
```

DTD

- La DTD
- Déclaration d'élément
 - Vide

- EMPTY

```
<!ELEMENT film EMPTY>
```

DTD

- La DTD
- Déclaration d'élément
 - Vide
 - Données textuelles

- (#PCDATA)

```
<!ELEMENT film (#PCDATA)>
```

DTD

- La DTD
- Déclaration d'élément
 - Vide
 - Données textuelles
 - Élément(s) imbriqué(s)

- Un element obligatoire

```
<!ELEMENT film (title)>
```

- Un element+ obligatoire
répété une ou plusieurs fois

```
<!ELEMENT film (title+)>
```

- Un element* optionnel répété
zéro ou plusieurs fois

```
<!ELEMENT film (title*)>
```

- Un element? optionnel répété
zéro ou une fois

```
<!ELEMENT film (title?)>
```


DTD

- La DTD
- Déclaration d'élément
 - Vide
 - Données textuelles
 - Élément(s) imbriqué(s)
 - Opérateurs

- Choix entre deux element | element

```
<!ELEMENT film (title|casting)>
```

- Plusieurs element1, element2 (dans cet ordre)

```
<!ELEMENT film (title,casting)>
```

DTD

- La DTD
- Déclaration d'élément
 - Vide
 - Données textuelles
 - Élément(s) imbriqué(s)
 - Opérateurs
 - Contenu mixte
- Choix entre un `element` ou du texte parsable (`#PCDATA`) avec cette syntaxe obligatoirement

```
<!ELEMENT film (#PCDATA | title)*>
```

DTD

- La DTD
- Déclaration d'élément
- Déclaration d'attribut

- Pour ajouter un @attribut

```
<title lang="en">Raiders of the Lost Ark</title>
```

- La DTD doit le spécifier avec un
<!ATTLIST>.

```
<!ELEMENT title (#PCDATA)>  
<!ATTLIST title  
  lang CDATA #IMPLIED>
```

DTD

- La DTD
- Déclaration d'élément
- Déclaration d'attribut
 - Optionnel, requis et fixe

- Un `@attribut` peut être `#IMPLIED` (optionnel)

```
<!ELEMENT title (#PCDATA)>  
<!ATTLIST title  
  lang CDATA #IMPLIED>
```

- Un `@attribut` peut être `#REQUIRED` (requis)

```
<!ELEMENT title (#PCDATA)>  
<!ATTLIST title  
  lang CDATA #REQUIRED>
```

- Un `@attribut` peut être `#FIXED` (fixe) avec une `"valeur"` par défaut

```
<!ELEMENT title (#PCDATA)>  
<!ATTLIST title  
  lang CDATA #FIXED "en">
```

DTD

- La DTD
- Déclaration d'élément
- Déclaration d'attribut
 - Optionnel, requis et fixe
 - Il peut avoir une valeur par défaut

- La "valeur" par défaut de l'@attribut est indiquée entre guillemets

```
<!ELEMENT title (#PCDATA)>  
<!ATTLIST title  
  lang CDATA "en">
```

→ Si @attribut n'est pas renseigné, alors sa "valeur" est "en"

```
<title lang="en">Raiders of the Lost Ark</title>
```

=

```
<title>Raiders of the Lost Ark</title>
```

DTD

- La DTD
- Déclaration d'élément
- Déclaration d'attribut
 - Optionnel, requis et fixe
 - Il peut avoir une valeur par défaut
 - Type CDATA

- Si la **valeur** de l'**@attribut** est une simple chaîne de caractères on peut utiliser CDATA

```
<!ATTLIST title  
  lang CDATA #IMPLIED>
```

→ Toutes les chaînes de caractères sont possibles comme "**valeur**" de **@lang**

```
<title lang="en">Star Wars</title>  
<title lang="english">Star Wars</title>  
<title lang="anglais">Star Wars</title>  
<title lang="arabe">Star Wars</title>
```

DTD

- La DTD
- Déclaration d'élément
- Déclaration d'attribut
 - Optionnel, requis et fixe
 - Il peut avoir une valeur par défaut
 - Type CDATA
 - Type énuméré

- Si la "valeur" de l'@attribut est restreint, on peut énumérer les options

```
<!ATTLIST title  
  lang (en | fr) #REQUIRED>
```

→ La "valeur" de @lang ne peut être que "en" ou "fr"

```
<title lang="en">Star wars</title>  
<title lang="fr">La Guerre des étoiles</title>
```

Mais pas

```
<film cote="es">La Guerre des étoiles</film>
```

- En l'absence de choix, on peut fixer une "valeur" par défaut

```
<!ATTLIST title  
  lang rating (en | fr) "en">
```

DTD

- La DTD
- Déclaration d'élément
- Déclaration d'attribut
 - Optionnel, requis et fixe
 - Il peut avoir une valeur par défaut
- Type CDATA
- Type énuméré
- Type ID

- Si la "valeur" de l'`@attribut` doit être unique

```
<!ATTLIST film
    cote ID #REQUIRED>
```

→ La "valeur" de `@cote` ne peut être la même pour deux `<film>`

```
<filmographie>
  <film cote="1">Raiders of the Lost Ark</film>
  <film cote="2">Return of the Jedi</film>
</filmographie>
```

Mais pas

```
<filmographie>
  <film cote="1">Raiders of the Lost Ark</film>
  <film cote="1">Return of the Jedi</film>
</filmographie>
```

- **IDREF** permet de référer à un élément identifié par la valeur de son attribut de type ID

```
<!ATTLIST affiche
    cote IDREF #REQUIRED>
```

```
<film cote="2">Return of the Jedi</film>
<affiche cote="2">Return of the Jedi</film>
```


DTD

- La DTD
- Déclaration d'élément
- Déclaration d'attribut
 - Optionnel, requis et fixe
 - Il peut avoir une valeur par défaut
 - Type CDATA
 - Type énuméré
 - Type ID
- Déclaration de plusieurs attributs

- L'<element> peut porter plusieurs @attributs

```
<!ATTLIST film
  cote ID #REQUIRED
  lang (en | fr) #REQUIRED
>
```

→ Le <film> a ici une @cote et une @lang

```
<film cote="1" cote="en">Star Wars</film>
```

DTD

- La DTD
- Déclaration d'élément
- Déclaration d'attribut
- Entité
 - interne

- On peut créer ses propres entités, en plus de celles qui sont génériques (e.g. `&` pour l'esperluette/«&»)

```
<!ENTITY UNI "Université">
```

→ `&UNI;` sera développé
«Université»

DTD

- La DTD
- Déclaration d'élément
- Déclaration d'attribut
- Entité
 - interne
 - externe parsée

- On peut aller chercher un autre document

```
<!ENTITY docXML SYSTEM "desktop/  
film2.xml">
```

→ À l'appel de `&docXML;` le fichier film2.xml du bureau sera inséré

DTD

- La DTD
- Déclaration d'élément
- Déclaration d'attribut
- Entité
 - interne
 - externe parsée
 - externe non-parsée

- On peut aller chercher un autre document

```
<!NOTATION jpeg SYSTEM "image/jpeg">  
<!ENTITY affiche SYSTEM "starWars.jpg"  
        NDATA jpeg>  
<!ATTLIST film  
        src ENTITY #REQUIRED>
```

→ Permet l'utilisation d'une image
comme @affiche

```
<film affiche="starWars">Star Wars</film>
```

=

```
<film affiche="starWars.jpg">Star Wars</film>
```

DTD

- La DTD
- Déclaration d'élément
- Déclaration d'attribut
- Entité
 - interne
 - externe parsée
 - externe non-parsée
 - paramètre

- On peut créer des raccourcis dans la DTD qui évitent les répétitions (et les erreurs)

```
<!ENTITY acteur (prenom,nom)>  
<!ENTITY realisateur (prenom,nom)>
```

=

```
<!ENTITY % noms "prenom,nom">  
<!ENTITY acteur (%noms)>  
<!ENTITY realisateur (%noms)>
```

→ Dans les deux cas on permet

```
<realisateur>  
  <prenom>Steven</prenom>  
  <nom>Spielberg</nom>  
</realisateur>  
<acteur>  
  <prenom>Harrison</prenom>  
  <nom>Ford</nom>  
</acteur>
```

DTD

- La DTD
- Déclaration d'élément
- Déclaration d'attribut
- Entité
- DTD vs schéma
- pas de typage précis du contenu des éléments (chaîne de caractères de texte, nombre entier, etc.) ou de leur sens (date, heure, nom, etc.)
- peu de contraintes sur les éléments (nombre d'occurrences, élément racine...)
- Et enfin, les DTD en elles-mêmes ne sont pas écrites en XML