XML: Plan

- I. Origines et base
- II. DTD et Schéma
- III. XSL et présentation

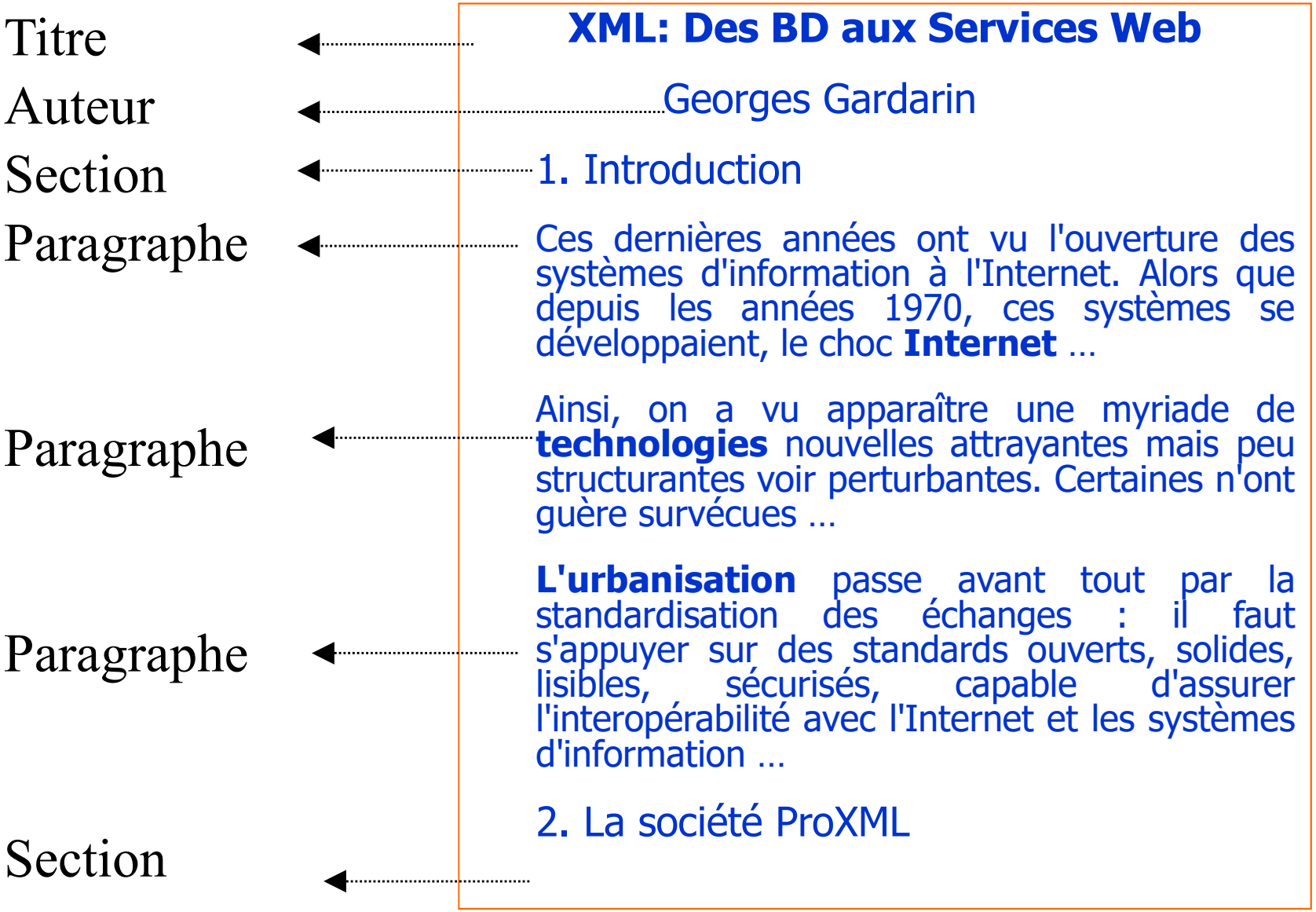
I. Origine et Concepts de base

1. Origines et Objectifs
2. SGML et HTML
3. Introduction à XML
4. XML pour quoi faire ?

1. Origine et objectifs

- XML est issu de la Gestion de Documents (GED)
- Séparation du fond de la forme.
  - Forme = présentation à partir de la structure (style)
  - Fond = structure + données (contenu)
- Multiples précurseurs dont les plus connues :
  - SGML pour la structuration
  - HTML pour la présentation
  - Approches mélangeant parfois le fond et la forme !

Présentation et Structuration



Vue Balisée

<Livre>
<Titre> XML : Des BD aux Services Web</Titre>
<Auteur>Georges Gardarin</Auteur>
<Section titre = "Introduction">
<Paragraphe>Ces dernières années ont vu l'ouverture des systèmes d'information à l'Internet. Alors que depuis les années 1970, ces systèmes se développaient, le choc Internet … </Paragraphe>
<Paragraphe>Ainsi, on a vu apparaître une myriade de technologies nouvelles attrayantes mais peu structurantes voir perturbantes. Certaines n'ont guère survécues … </Paragraphe>
<Paragraphe>L'urbanisation passe avant tout par la standardisation des échanges : il faut s'appuyer sur des standards ouverts, solides, lisibles, sécurisés, capable d'assurer l'interopérabilité avec l'Internet et les systèmes d'information … </Paragraphe>
</Section>
<Section titre= "La Société ProXML"> …
</Section>
</Livre>

Les balises (tags) peuvent porter plus ou moins de sémantique

World Wide Web Consortium

- W3C - Fondé en 1994
- Consortium industriel international accueilli par différents sites
  - MIT/LCS aux Etats-Unis
  - INRIA en Europe
  - Keio University au Japon
- 452 membres industriels en 2021
- Président: Tim Berners-Lee, inventeur du web

W3C : dans quel but ?

- Accroître le potentiel du Web
- Développer des protocoles communs
- Assurer l'inter-opérabilité sur le Web entre les différents systèmes
- Stock d'informations sur les standards et les normes pour développeurs et utilisateurs
  - Code référence pour présenter et promouvoir les différents standards
  - Prototypes variés et exemples d'applications

XML : objectifs

- XML= un nouveau langage d'échange basé sur le balisage
- XML= plus simple que SGML
- XML= plus ouvert que HTML englobé (XHTML)
- XML = développé par XML Working Group dirigé par le W3C (depuis 1996)
- XML 1.0 = recommandation officielle du W3C depuis le 10 février 1998

les 10 objectifs de conception:

- XML doit pouvoir être utilisé sans difficulté sur Internet
- XML doit soutenir une grande variété d'applications
- XML doit être compatible avec SGML et HTML
- Il doit être facile d'écrire des programmes traitant les documents XML
- Le nombre d'options dans XML doit être réduit au minimum, idéalement à aucune

les 10 objectifs de conception

- Les documents XML doivent être lisibles par l'homme et raisonnablement clairs
- La spécification de XML doit être disponible rapidement
- La conception de XML doit être formelle et concise
- Il doit être facile de créer des documents XML
- La concision dans le balisage de XML est peu importante

Forces de XML

- Séparation du contenu et de la présentation
- Moins confus que HTML
- Plus simple que SGML
- Idéal pour l'échange de données semi-structurées
- Utilisable entre applications hétérogènes

2. SGML et HTML

- 1969 C. Goldfarb, E. Mosher, R. Lorie inventent GML chez IBM
- GML a été créé pour éditer des documents, les mettre en page et les partager au sein de systèmes de gestion éditoriaux
- 1978 Goldfarb prend la tête d'un comité « Computer Language for the Processing of Text » au sein de l'American National Standards Institute (ANSI).

Qu'est-ce que SGML ?

- Une norme internationale :
  - Standard Generalized Markup Language
  - ISO 8879 - 1989
- Un métalangage de balisage de documents
  - lisible par l'être humain et traitable par une machine
  - permet de définir des langages de balisage
- Les documents sont balisés conformément à la DTD
  - document instances de DTD
  - permet un balisage sémantique du fond
- Syntaxe plus permissive que XML
  - Analyseurs (parsers) plus complexes

SGML : critiques

- Très lourd et complexe pour la mise en œuvre de documents respectant ce format
- Une grande rigueur est demandée à l'entrée des documents
- Standard complexe et complet pour le traitement des documents
- Liens hypertextes possibles mais complexes

HTML : présentation

- Proposé par le W3C comme format de documents sur le Web
- Langage avec des balises fixes standardisées permettant la mise en forme d'un texte
- Standard reconnu par tous les navigateurs, très populaire sur le Web
- Langage remplacé progressivement par XHTML 1.1, HTML épuré conforme aux règles XML
- Nouvelle version en cours de standardisation XHTML 2.0, pour le web 2.0

```
<HTML>
<HEAD>
<TITLE> Exemple </TITLE>
</HEAD>
<BODY>
<H1>Contenu du document</H1>
<A HREF = "http://www.server.fr/Info
    /dir/test.html"> une référence
    externe
</A>
</BODY>
</HTML>
```

HTML : inconvénients

- Normalisation des différentes balises difficile :
  - les constructeurs ont eu tendance à définir leurs propres balises pour répondre à leurs besoins (incompatibilité)
  - HTML 4.0
    - boutons, tables, applets, objects, graphiques, maths, ...
    - styles, frames, protections, ...
- Mises à jour et extractions difficiles :
  - données utiles et mises en forme mixées ;
  - remise en forme de l'ensemble des pages d'un site fastidieuse.
- Mélange le fond et la forme :
  - méta-données avec la présentation
  - pages conçues pour un type de terminal

SGML et HTML : Résumé

- SGML
  - langage de la GED plutôt complexe
  - très utilisé dans l'industrie
- HTML
  - spécialisation de SGML
  - adapté à la présentation autonome ou en conjonction avec CSS
  - inadapté à l'échange entre programmes

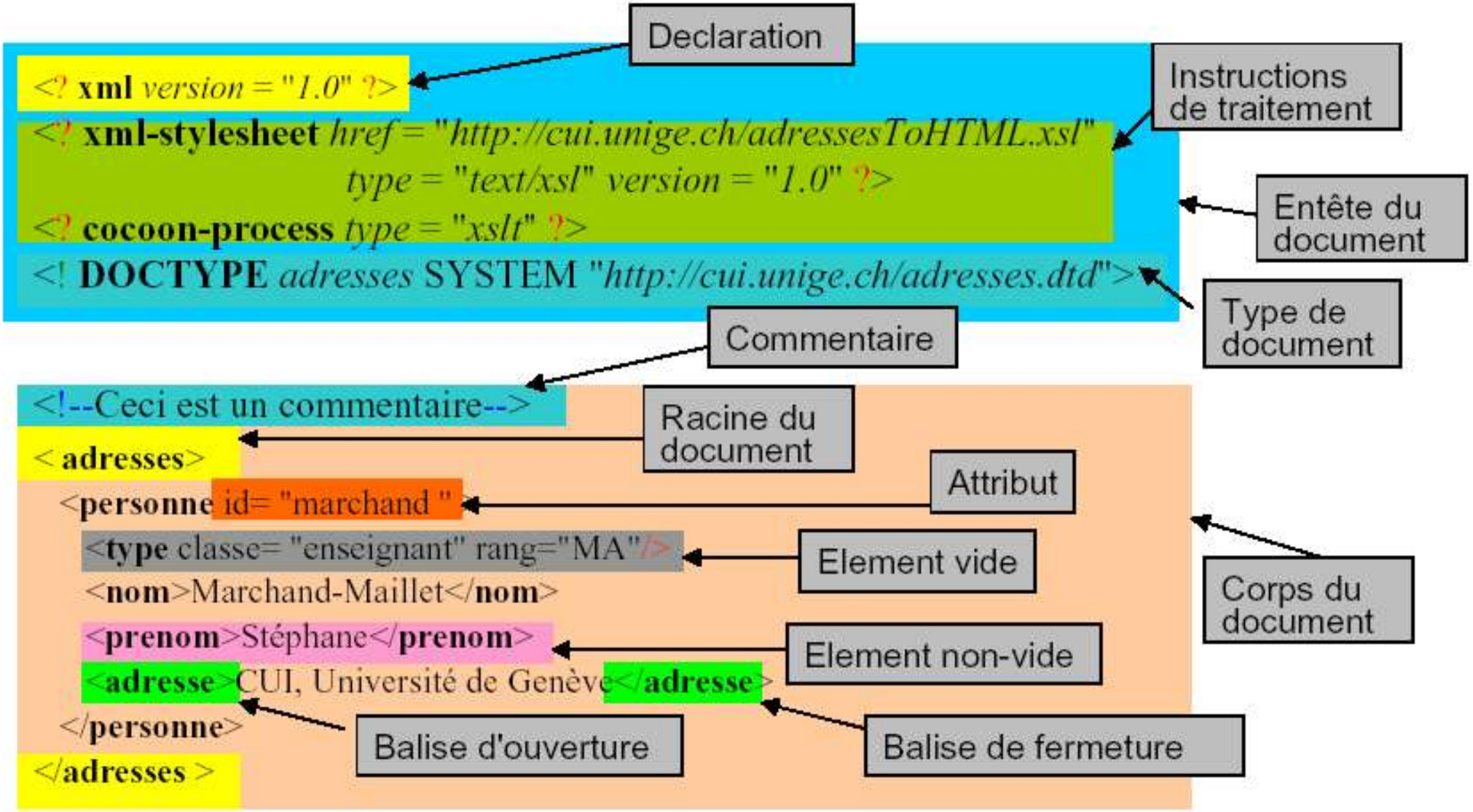
3. XML: définitions de base

- XML est un méta-langage universel pour représenter les données échangées sur le Web qui permet au développeur de délivrer du contenu depuis les applications à d'autres applications ou aux navigateurs
- XML standardise la manière dont l'information est :
  - échangée
  - présentée
  - archivée
  - retrouvée
  - transformée
  - ...

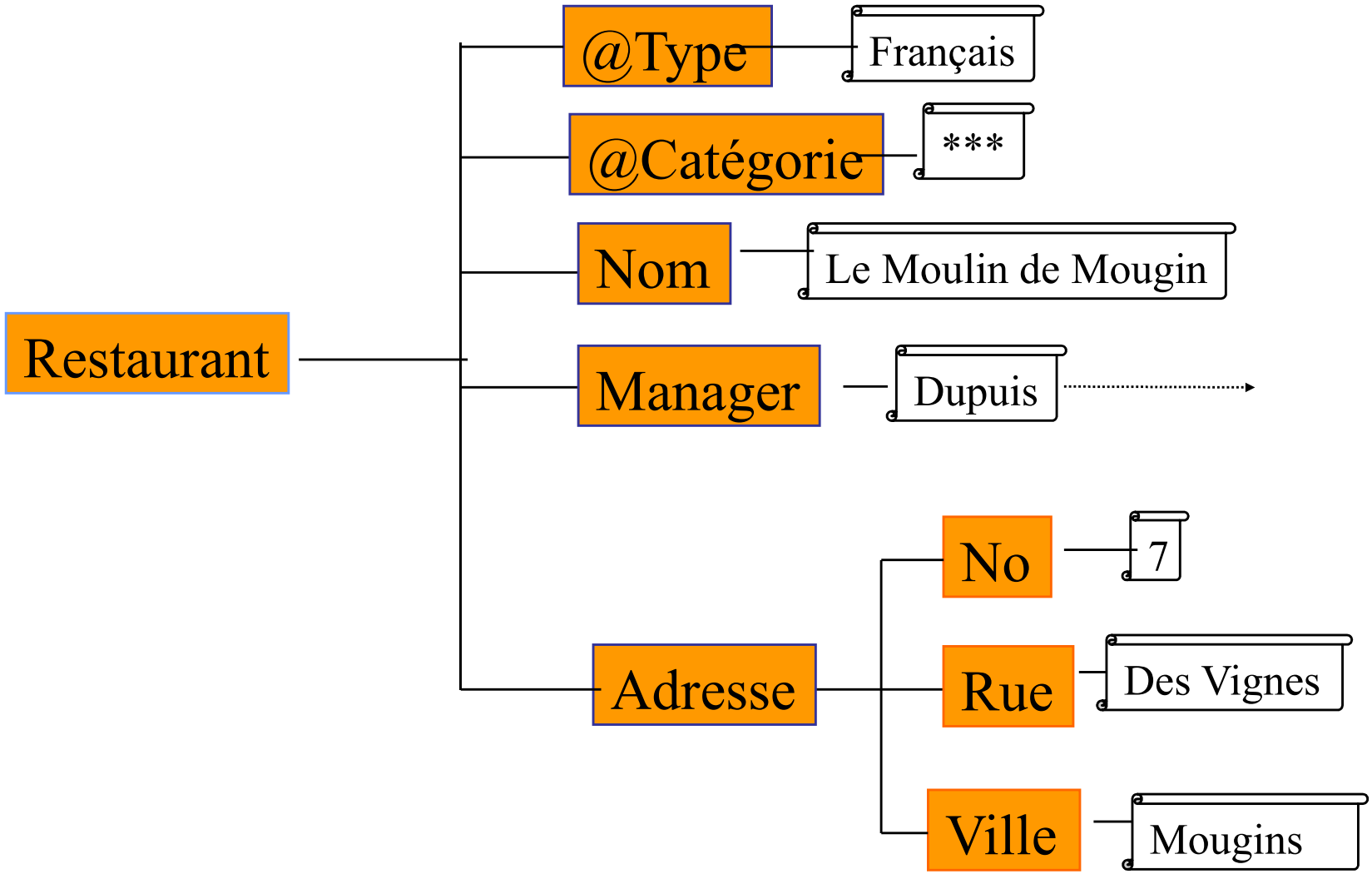
Concepts du modèle

- Balise (ou tag ou label)
  - Marque de début et fin permettant de repérer un élément textuel
  - Forme: <balise> de début, </balise> de fin
- Elément de données
  - Texte encadré par une balise de début et une de fin
  - Les éléments de données peuvent être imbriquées
    - <producteur>
    - <adresse>
    - <rue>A. Briand</rue>
    - <ville>Dijon</ville>
    - </adresse>
    - </producteur>
- Attribut
  - Doublet nom="valeur" qualifiant une balise
    - <producteur no="160017" region="Bourgogne">

Exemple complet



Modèle arborescent



Présentation textuelle

```
<Restaurant Type="Français" Categorie="***">
<Nom>Le Moulin de Mougins</Nom>
<Manager ... >Dupuis</Manager>
<Adresse>
    <No> 7</No>
    <Rue>Des Vignes</Rue>
    <Ville>Mougins</Ville>
</Adresse>
</Restaurant>
```

Arbre sérialisé

Espaces de noms (1)

- Permet de distinguer des tags de même nom
- Mécanisme intéressant pour l'intégration de contenus

préfixe = balise

Le préfixe permet de retrouver le langage

```
<fus:Guide xmlns:fus= "http://www.myGuide.com"
   xmlns:mic="http://www.michelin.com/2001/Guide",
   xmlns:pj="http://www.pagejaune.com/2001/Annuaire">
  <mic:Adresse><ville>Mougins</ville><zip>06212</zip></mic:Adresse>
  <pj:Adresse>Le Moulin 06 Mougins</pj:Adresse>
</fus:Guide>
```

Espaces de noms (2)

- Identifié par une URI (Uniform Resource Identifier)
  - L'URL n'a pas nécessité d'exister
  - Référence souvent le schéma
- Un namespace déclaré sans préfixe devient le namespace par défaut de tous les éléments descendants
- Les éléments non qualifiés appartiennent au namespace le plus interne

```
<Livre xmlns="www.bookstuff.org/bookinfo">
    <Titre>Tout sur XML</Titre>
    <Auteur>Joyeux l'analyste</Auteur>
    <Editeur xmlns:pub="urn:publishers:publinfo">
        <Nom>Microsoft Press</Nom>
    </Editeur>
</Livre>
```

Les langages techniques XML (1)

XML	**Extensible Markup Language**	**Defines XML documents**
Infoset	**Information Set**	**Abstract model of XML data; definition of terms**
DTD	**Document Type Definition**	**Non-XML schema**
XSD	**XML Schema**	**XML-based schema language**
XDR	**XML Data Reduced**	**An earlier XML schema**
CSS	**Cascading Style Sheets**	**Allows you to specify styles**
XSL	**Extensible Stylesheet Language**	**Language for expressing stylesheets; consists of XSLT and XSL-FO**
XSLT	**XSL Transformations**	**Language for transforming XML documents**
XSL-FO	**XSL Formatting Objects**	**Language to describe precise layout of text on a page**

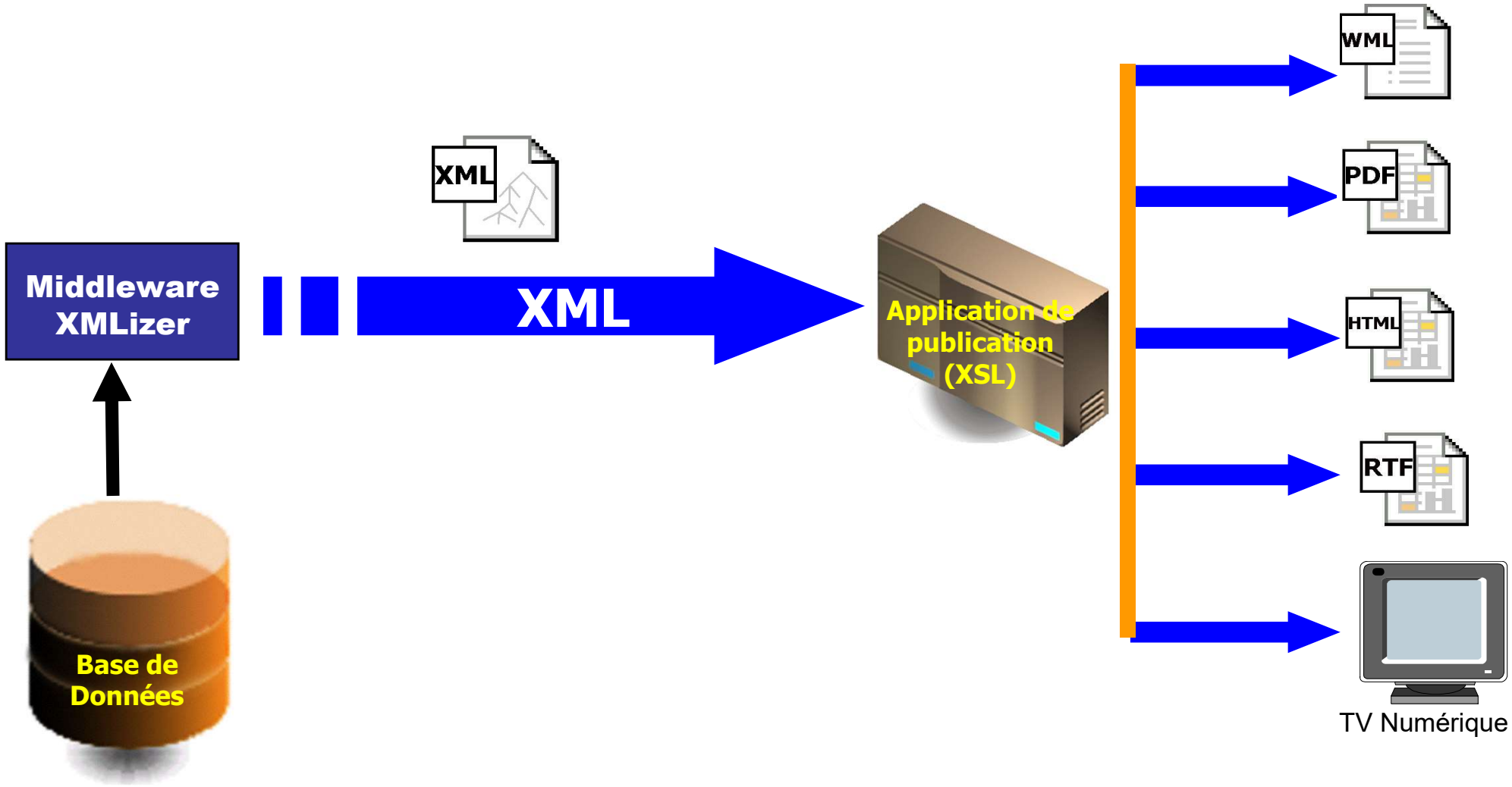
Les langages techniques XML (2)

XPath	**XML Path Language**	**A language for addressing parts of an XML document, designed to be used by both XSLT and XPointer**
XPointer	**XML Pointer Language**	**Supports addressing into the internal structures of XML documents**
XLink	**XML Linking Language**	**Describes links between XML documents**
XQuery	**XML Query Language (draft)**	**Flexible mechanism for querying XML data as if it were a database**
DOM	**Document Object Model**	**API to read, create and edit XML documents; creates in-memory object model**
SAX	**Simple API for XML**	**API to parse XML documents; event-driven**
Data Island	**XML data embedded in a HTML page**	
Data Binding	**Automatic population of HTML elements from XML data**	

4. Pourquoi XML?

- Définir vos propres langages d'échange
  - Commande, facture, bordereau de livraison, etc.
- Modéliser des documents et des messages
  - Modèle logique de données
  - Eléments typés agrégés (DTD, XML Schema)
  - Passerelle avec Unified Modelling Language (UML)
- Publier des informations
  - Neutre du point de vue format
  - Mise en forme avec des feuilles de style
- Archiver des données
  - Auto-description des archives

Publication multi-supports



Forces et faiblesses de XML

- Une technologie structurante
- Clarifie tous les échanges
- Des standards internes et externes
- Transversale à l'entreprise
  - Échanges de données
  - Bureautique
  - GED
  - Sites Web
  - Bases de données
  - Intégration e-business
  - …

- Une syntaxe bavarde
- Un méta-langage, mais de nombreux langages
- Coûteux en CPU
  - Parsing
- Coûteux en mémoire
  - Instanciation
- Format compressé à l'étude
  - XML Binaire

Un choix stratégique de direction

II. Conception et Manipulation des Documents

1. Introduction
2. DTD
3. Schéma
4. Outils

1. Introduction

- Un document XML peut être associé à :
  - une DTD ou un schéma pour décrire les balises
  - Une feuille de style pour présenter les données
- DTD ou/et schéma permettent de définir son propre langage basé sur XML
  - Vocabulaire (balises)
  - Grammaire (imbrications)

Validité des documents

- Document bien formé (Well Formed Document)
  - balises correctement imbriquées
  - analysable (parsable) et manipulable
  - pas nécessairement valide
- Document valide (Valid Document)
  - bien formé + conforme à la DTD ou au schéma

2. DTD

- Permet de définir le «vocabulaire» et la structure qui seront utilisés dans le document XML
- Grammaire du langage dont les phrases sont des documents XML (instances)
- Peut être mise dans un fichier (DTD externe) et être appelée dans le document XML

Déclaration d'élément simple

- <! ELEMENT balise (définition) >

  - Le paramètre définition représente soit un type de donnée prédéfini, soit un élément de données composé, constitué lui même d'éléments
  - Types prédéfinis
    - ANY : L'élément peut contenir tout type de donnée
    - EMPTY : L'élément ne contient pas de données spécifiques
    - #PCDATA : L'élément doit contenir une chaîne de caractère
  - Exemple
    - <! ELEMENT Nom (#PCDATA)>
    - <Nom>Victor Hugo</Nom>

Déclaration d'élément composé

- Définit une séquence ou un choix d'éléments
- Syntaxe spécifique avec opérateurs de composition d'éléments :

<! ELEMENT balise (*composition*) >

Opérateur	Signification	Exemple
+	L'élément doit être présent au minimum une fois	A+
*	L'élément peut être présent plusieurs fois (ou aucune)	A*
?	L'élément peut être optionnellement présent	A?
|	L'élément A **ou** B peuvent être présents (pas les deux)	A|B
,	L'élément A doit être présent et suivi de l'élément B	A,B
()	Les parenthèses permettent de regrouper des éléments afin de leur appliquer les autres opérateurs	(A,B)+

Exemple d'élément composé

- <!ELEMENT personne (nom, prenom+, tel?, adresse) >
  - <!ELEMENT nom (#PCDATA) >
  - <!ELEMENT prenom (#PCDATA) >
  - <!ELEMENT tel(#PCDATA) >
  - <!ELEMENT email (#PCDATA) >
  - <!ELEMENT adresse (ANY) >
- <personne>
  <nom>Hugo</nom>
  <prenom>Victor</prenom>
  <prenom>Charles</prenom>
  <tel>01120243</tel>
  <adresse><rue></rue><ville>Paris</ville></adresse>
  </personne>

Déclaration d'attributs

- <! ATTLIST *balise Attribut Type Mode* >
- *balise* spécifie l'élément auquel est attaché l'attribut
- *Attribut* est le nom de l'attribut déclaré
- *Type* définit le type de donnée de l'attribut choisi parmi:
  - CDATA
    - Chaînes de caractères entre guillemets ("aa") non analysées
  - Enumération
    - Liste de valeurs séparées par |
    - <! ATTLIST *balise Attribut* (Valeur1 | Valeur2 | ...) >
  - ID et IDREF
    - Clé et référence à clé
- *Mode* précise le caractère obligatoire ou non de l'attribut
  - #REQUIRED, #IMPLIED ou #FIXED

Exemple d'attributs

```
<! ATTLIST personne
                num ID,
                age CDATA,
                genre (Masculin | Feminin ) >

<!ELEMENT auteur (#PCDATA) >
<!ELEMENT editeur (#PCDATA) >
<!ATTLIST auteur
                genre (Masculin | Feminin ) #REQUIRED
                ville CDATA #IMPLIED>
<!ATTLIST editeur
                 ville CDATA #FIXED "Paris">
```

Exemple de DTD

```
<!DOCTYPE Restaurant [
<!ELEMENT Restaurant (Nom, Adresse, (Telephone | Manager), Menu?) >
<!ATTLIST Restaurant
        categorie CDATA #REQUIRED
        type CDATA #FIXED "francais" >
<!ELEMENT Nom (#PCDATA)>
<!ELEMENT Adresse (No, Rue, Ville)>
<!ELEMENT No (#PCDATA)>
<!ELEMENT Rue (#PCDATA)>
<!ELEMENT Ville (#PCDATA)>
<!ELEMENT Telephone (#PCDATA)>
<!ELEMENT Manager (#PCDATA)>

<!ELEMENT Menu EMPTY>
<!ATTLIST Menu
        Nom CDATA #REQUIRED>

]
```

Entité paramètre

- Permet la définition d'un groupe d'éléments sous un nom (macro)
  - <!ENTITY %nom "definition">
- Réutilisable dans une DTD par simple appel :
  - %nom;
- Exemple :
  - <!ENTITY %genres "(homme | femme)">
    <!ATTLIST auteur genre %genres; #REQUIRED>
- Peuvent être externes :
  - <!ENTITY %mpeg PUBLIC "http://www.myweb.fr/fic.dtd">

Insuffisance des DTD

- Pas de types de données
  - difficile à interpréter par le récepteur
  - difficile à traduire en schéma objets
- Pas en XML
  - langage spécifique
- Propositions de compléments
  - XML-data de Microsoft
  - XML-schema du W3C