Chapitre I

XML, un langage de balisage extensible

Le langage XQuery auquel est consacré ce livre est destiné à la manipulation de données décrites en XML[1]. Il est donc nécessaire, avant d'aborder l'étude du langage XQuery lui-même, d'avoir une bonne compréhension des données XML : leur syntaxe, leur structure logique et leur typage. C'est l'objectif de ce chapitre.

À l'origine, XML ou plutôt SGML[2] dont XML est une variante, a été développé pour décrire des documents textuels en marquant la structure d'un document à l'intérieur du document lui-même. Mais il est vite apparu que le champ d'application de XML était beaucoup plus large que celui des données documentaires. En effet, XML est un langage très flexible particulièrement bien adapté à la description de données arborescentes qui sont présentes dans un grand nombre d'applications. XML est aussi un bon format d'échange de données entre applications hétérogènes, celles qui sont accessibles sur le Web notamment. Cet élargissement du champ d'application de XML a conduit (i) au développement par le W3C du langage XML Schéma pour typer les données XML de façon beaucoup plus précise que ne le permettait initialement XML, (ii) à l'extension des SGBD relationnels et de SQL à la prise en compte de données XML et (iii) au développement de systèmes de gestion de bases de données *XML natif* intégrant XQuery pour leur interrogation.

Dans la suite de ce chapitre, on étudiera :

- la syntaxe d'un document XML (paragraphe I.1) et la définition de sa structure sous la forme d'une grammaire non contextuelle appelée Déclaration de Type de Document (DTD) (paragraphe I.2) ;
- les espaces de noms dont l'utilisation est indispensable pour éviter les conflits de nommage dans les applications qui manipulent des données XML issues de plusieurs sources (paragraphe I.3) ;
- le langage XML Schéma pour le typage de données XML (paragraphe I.4).

I.1 Syntaxe d'un document XML

Le qualificatif de *langage de balisage* attribué au langage XML provient de son objectif initial : marquer, au sein du document lui-même, les différentes parties d'un document (sections, paragraphes, expressions à mettre en valeur...) par des paires de balises catégorisées

[1] *Extensible Markup Language (XML) 1.0* (`http://www.w3.org/TR/xml/`)

[2] Standard Generalized Markup Language (ISO 8879), inventé par Charles Goldfarb

et imbriquées hiérarchiquement. Les portions d'un document ainsi balisées sont appelées éléments. La balise ouvrante d'un élément peut être enrichie par des attributs apportant des informations supplémentaires non obligatoirement présentes dans le texte du document, mais pouvant être utiles pour certaines utilisations de ce document. Le qualificatif de *langage extensible* provient, lui, du fait que les catégories de balises utilisables ne sont pas imposées, comme c'est le cas en HTML, mais peuvent être librement choisies par le concepteur d'un document ou d'une famille de documents.

Un document XML a une structure arborescente. Il est composé d'un élément : l'élément du document et d'une suite, éventuellement vide, d'instructions de traitement ou de commentaires. Un élément est composé d'un nom, d'un ensemble d'attributs et d'un contenu formé d'une suite de caractères dans laquelle peuvent être insérés des éléments, des instructions de traitement ou des commentaires. Un attribut a un nom et une valeur qui est une suite de caractères.

La structure d'un document XML peut être déclarée dans une DTD (Déclaration de Type de Document) qui se présente sous la forme d'une grammaire non contextuelle. Un document XML est dit bien formé si sa description est syntaxiquement correcte. Un document XML bien formé est dit valide si sa description est conforme à une DTD.

Un document XML est destiné à être traité par un processeur XML qui est mis en œuvre par une application. Un processeur XML lit le document, vérifie s'il est bien formé et valide dans le cas où sa DTD est fournie, et le traduit sous une forme interne adaptée à son traitement par cette application.

C'est un guide de randonnées à ski, *Ski de randonnée en Queyras*, élaboré pour la circonstance, qui servira de support aux exemples proposés dans ce livre. Ce guide décrit une trentaine d'itinéraires à ski de difficulté modérée dans le massif du Queyras. Ces itinéraires sont regroupés par secteur géographique. Le guide est composé de son titre, de son auteur, de son éditeur, de son année d'édition et des secteurs dans lesquels des itinéraires à ski sont proposés. Chaque secteur est composé d'un identificateur, d'un nom, d'une courte description et des itinéraires à ski proposés dans ce secteur. Chaque itinéraire est composé d'un identificateur, du nom et de l'altitude du point (sommet ou col) d'arrivée, de la cotation de sa difficulté (1 à 3 étoiles) et d'une courte description qui comporte le lieu de départ, les lieux traversés et peut comporter des renvois à un autre itinéraire, des avertissements sur des dangers éventuels (corniches, plaques à vent, coulées...) ou des recommandations de matériel (corde, crampons, piolet). L'identificateur d'un secteur ou d'un itinéraire l'identifie de façon unique dans le guide. La description d'un secteur ou d'un itinéraire est composée d'un ou plusieurs paragraphes. La figure I.1 présente un extrait d'une page de ce guide, la figure I.2 présente une carte de deux itinéraires décrits sur cette page. Ce guide est enregistré, sous la forme d'un document XML, dans le fichier `queyras.xml` dont le contenu est fourni en annexe A. La figure I.3 présente un extrait de ce document.

I.1.1 Caractères et noms

Un document XML est une suite de caractères parmi lesquels il faut distinguer les caractères de donnée et les caractères de balisage. Par exemple, si XML est utilisé pour décrire le contenu d'un livre, les caractères de donnée sont ceux du texte de ce livre (titres, paragraphes, noms des auteurs...) et les caractères de balisage sont ceux des balises qui marquent les différents composants de ce livre.

Secteur de St-Véran

Les itinéraires de ce secteur permettent d'atteindre des cols et des sommets sur les crêtes qui dominent l'Aigue Blanche et l'Aigue Agnelle.
En hiver, la route est dégagée jusqu'à St-Véran (2050 m) et jusqu'au Pont de Lariane (2024 m) sur la route du Col Agnel.

…

4. Col de Longet (2701 m) *

Du Pont de Lariane (2024 m), traverser le pont, longer sur 200 m la rive gauche de l'Aigue Agnelle puis monter au travers de la forêt vers le sud-est pour rejoindre le Vallon de Longet. Le suivre jusqu'au Col de Longet (2701 m). Ne pas longer de trop près la face nord-est du Pic de Château Renard : risque de coulées.

5. Pic Traversier (2882 m) **

Suivre l'itinéraire n° 4 jusque sous le Col de Longet. Remonter vers l'est un petit vallon qui se redresse jusqu'au sommet du Pic Traversier (2882 m).

…

Figure I.1. *Extrait d'une page du guide Ski de randonnée en Queyras*

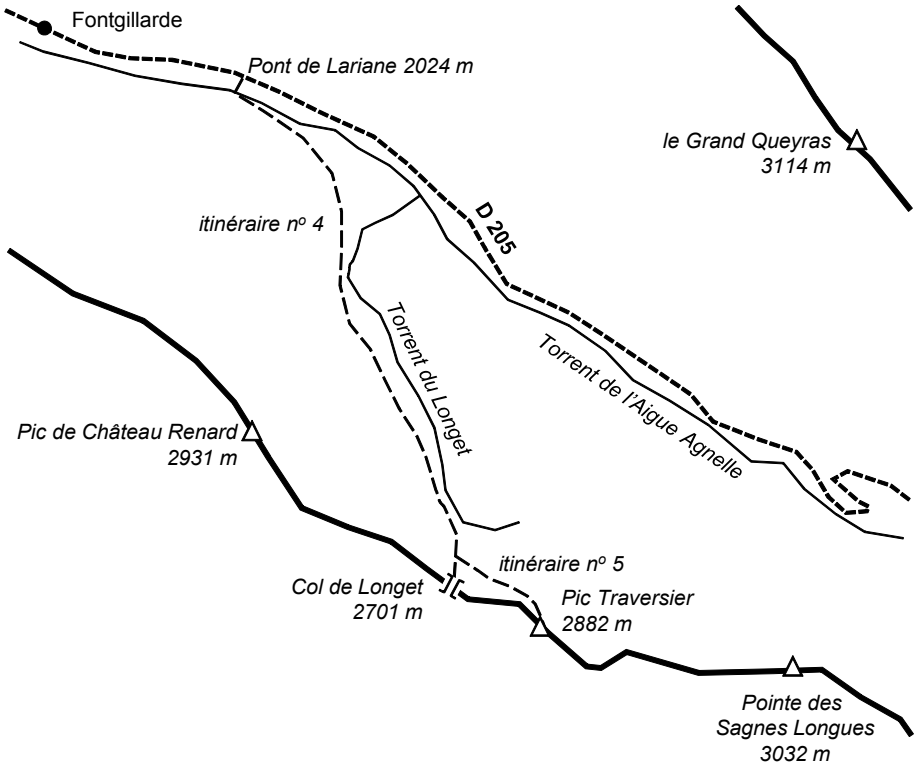

Figure I.2. *Carte des itinéraires n° 4 et n° 5 du secteur de St-Véran*

```
<guide>
<titre>Ski de randonnée en Queyras</titre>
<auteur>Jacques Le Maitre</auteur>
<éditeur>inédit</éditeur>
<année>2013</année>
...
<secteur id="S2">
<nom>Secteur de St-Véran</nom>
<para>Les itinéraires de ce secteur permettent d'atteindre des cols et
   des sommets sur les crêtes qui dominent l'Aigue Blanche et l'Aigue
   Agnelle.</para>
<para>En hiver, la route est dégagée jusqu'à St-Véran (2050 m) et
   jusqu'au Pont de Lariane (2024 m) sur la route du Col Agnel.</para>
...
<itinéraire id="I2.4">
<nom>Col du Longet</nom>
<alt>2701</alt>
<cotation>*</cotation>
<para>Du Pont de Lariane (2024 m), traverser le pont, longer sur 200 m
   la rive gauche de l'Aigue Agnelle puis monter au travers de la forêt
   vers le sud-est pour rejoindre le Vallon de Longet. Le suivre
   jusqu'au Col de Longet (2701 m). <note type="prudence">Ne pas longer
   de trop près la face nord-ouest du Pic de Château Renard : risque de
   coulées</note>.</para>
</itinéraire>
<itinéraire id="I2.5">
<nom>Pic Traversier</nom>
<alt>2882</alt>
<cotation>**</cotation>
<para>Suivre l'<renvoi cible="I2.4"/> jusque sous le Col de Longet.
   Remonter vers l'est un petit vallon qui se redresse jusqu'au sommet
   du Pic Traversier (2822 m).</para>
</itinéraire>
...
</secteur>
</guide>
```

Figure I.3. *Extrait du document XML* `queyras.xml`

Les caractères utilisables dans un document XML sont ceux de l'UCS (*Universal Character Set*) qui couvre pratiquement toutes les langues du monde ainsi que les caractères mathématiques ou graphiques, ou l'un de ses sous-ensembles tel que le jeu de caractères de l'ISO/CEI 8859-1 qui couvre les langues européennes occidentales (allemand, anglais, espagnol, français, italien...).

Un caractère de nom est soit une lettre ou un idéogramme, soit un chiffre, soit un point, soit un tiret, soit une espace soulignée, soit un deux-points.

Un caractère blanc est soit une espace, soit une tabulation, soit un retour chariot (CR), soit un saut de ligne (LF).

Un nom XML est une suite de un ou plusieurs caractères de nom dont le premier est soit une lettre ou un idéogramme, soit une espace soulignée, soit un deux-points et les suivants sont des caractères de nom. Le caractère deux-points est réservé aux noms préfixés (voir

paragraphe I.3). Par exemple :

▷ `alt`, `année`, `titre-1` et `g:cotation` sont des noms XML.

Un lexème nominal est une suite de caractères de nom.

I.1.2 Document

Un document XML bien formé est composé d'un prologue facultatif, de l'élément du document et d'une suite éventuellement vide de commentaires ou d'instructions de traitement.

Le prologue d'un document XML indique notamment la version de XML et l'encodage des caractères. L'encodage par défaut est l'UTF-8 qui est un codage en nombre d'octets variable (1 à 4) des caractères de l'UCS et qui inclut le code ASCII de base. Par exemple :

▷ `<?xml version="1.0"?>`

est le prologue d'un document XML conforme à la version 1.0 et dont les caractères sont codés en UTF-8 ;

▷ `<?xml version="1.0" encoding="iso-8859-1"?>`

est le prologue d'un document XML conforme à la version 1.0 et dont les caractères sont codés sur 1 octet selon la norme ISO-8859-1 qui est un sur-ensemble de l'ISO/CEI 8859-1 (les caractères ajoutés sont des caractères de contrôle).

Une instruction de traitement est une information à l'usage de l'application qui manipule le document. Par exemple :

▷ `<?robots index="no" follow="no"?>`

est une instruction de traitement dans laquelle `robots` est la cible de l'instruction (ici, le robot d'un moteur de recherche parcourant le web) et `index="yes" follow="no"` est le contenu de cette instruction (ici, la demande au robot de ne pas indexer ce document, ni d'en suivre les liens hypertexte qu'il contient)[3].

Un commentaire apporte des informations sur le contenu d'un document ou met en valeur son organisation. Par exemple :

▷ `<!--Liste des secteurs-->`

est un commentaire dont le contenu est `Liste des secteurs`.

Un commentaire peut contenir n'importe quelle suite de caractères excepté `--`.

I.1.3 Éléments

Un élément est composé d'une balise ouvrante, d'un contenu et d'une balise fermante. La balise ouvrante contient le nom de l'élément, un nom XML, et un ensemble éventuellement vide d'attributs. Par exemple :

[3] exemple extrait de *XML in a Nutshell*, Second Edition, O'Reilly

▷ `<note type="prudence">risque de coulées</note>`

est un élément dans lequel :

- `<note type="prudence">` est la balise ouvrante ;
- `note` est le nom de l'élément ;
- `type="prudence"` est un attribut ;
- `risque de coulées` est le contenu ;
- `</note>` est la balise fermante.

Un attribut est un couple *n*`:"`*v*`"` où *n* est le nom de l'attribut, un nom XML, et *v* est la valeur de l'attribut, une suite, éventuellement vide, de caractères, d'appels de caractère ou d'appels d'entité. Par exemple :

▷ `type="matériel"`

est un attribut dont le nom est `type` et la valeur est `matériel`.

Si une valeur d'attribut peut contenir le caractère de donnée `"` (guillemets), on pourra la placer entre apostrophes (`'`) au lieu de la placer entre guillemets.

Le contenu d'un élément est une suite de constituants dont chacun peut être un caractère de donnée, un élément, une instruction de traitement, un commentaire, une section CDATA, un appel de caractère ou un appel d'entité. Les éléments, les instructions ou les commentaires qui apparaissent dans le contenu d'un élément sont les enfants de cet élément. Par exemple :

```
▷ <para>Suivre l'<renvoi cible="I2.4"/> jusque sous le Col de Longet.
     Remonter vers l'est un petit vallon qui se redresse jusqu'au
     sommet du Pic Traversier (2822 m).</para>
```

est un élément dont le contenu est constitué :

- de la suite de caractères de donnée `Suivre l'` ;
- de l'élément `<renvoi cible="I2.4"/>` qui est un enfant de l'élément `para` ;
- de la suite de caractères de donnée `jusque sous le Col de Longet...`

On distingue trois types de contenu : contenu vide, contenu uniquement composé d'éléments et contenu mixte constitué d'une suite non vide de caractères de donnée dans laquelle peuvent être insérés des éléments, des instructions de traitement ou des commentaires. Par exemple :

▷ `<renvoi cible="I2.4"/></renvoi>`

qui peut aussi s'écrire

`<renvoi cible="I2.4"/>`

est un élément dont le contenu est vide ;

```
▷ <itinéraire id="I2.4">
  <nom>Col du Longet</nom>
  <alt>2701</alt>
  <cotation>*</cotation>
  <para>Du Pont de Lariane (2024 m), traverser...</para>
  </itinéraire>
```

est un élément dont le contenu est uniquement composé d'éléments ;

▷ `<para>Suivre l'<renvoi cible="I2.4"/> jusque...</para>`

est un élément dont le contenu est mixte.

I.1.4 Appel de caractère, appel d'entité caractère et section CDATA

Certains caractères de donnée ne peuvent pas être saisis au clavier. Ils peuvent être inclus dans un document XML par un appel de caractère. Si *c* est un caractère de l'UCS dont la valeur décimale du code est *d* et la valeur hexadécimale est *h* :

– `&#`*d*`;` est un appel de caractère qui remplace le caractère de donnée *c* ;

– `&#x`*h*`;` est un appel de caractère qui remplace le caractère de donnée *c*.

Ces deux appels sont reconnus par un processeur XML comme le caractère de donnée *c*. Par exemple :

▷ `&#38;`

est un appel de caractère qui est reconnu comme le caractère de donnée `&` ;

▷ `&#x03A6;`

est un appel de caractère qui est reconnu comme le caractère de donnée Φ.

Dans le contenu d'un élément, les caractères `<` et `&` sont réservés car ils marquent le début d'une balise (balise ouvrante ou fermante d'élément, instruction de traitement, commentaire, section CDATA, appel de caractère ou d'entité). De même dans une valeur d'attribut, le caractère `"` ou le caractère `'` qui en a marqué le début est réservé car il en marque aussi la fin. Lorsqu'un caractère de donnée est l'un de ces quatre caractères, il est nécessaire de ne pas le confondre avec un début de balise ou une marque de fin d'attribut. Deux solutions sont disponibles :

– dans le contenu d'un élément ou dans une valeur d'attribut, remplacer ce caractère de donnée par un appel d'entité caractère ;

– dans le contenu d'un élément, inclure une suite de caractères de donnée contenant des caractères `<` ou `&` dans une section CDATA.

Cinq appels d'entités caractère sont prédéfinis : `&`, `<`, `>`, `&apos;` et `&quot;`. Ils sont reconnus par un processeur XML comme les caractères de donnée `&`, `<`, `>`, `'` et `"`. Par exemple :

▷ la condition : «altitude < 3000 m», peut être représentée par l'élément

```
<condition>altitude &lt; 3000 m</condition>
```

dans lequel l'appel d'entité caractère `<` est reconnu comme le caractère de donnée `<`.

Une section CDATA a la forme suivante : `<![CDATA[`*s*`]]>`, où *s* est son contenu. Le contenu d'une section CDATA est une suite de caractères ne contenant pas la suite de caractères `]]>`, qui est reconnue par un processeur XML comme une suite de caractères de donnée. Par exemple :

▷ la phrase «<alt>2882</alt> est un élément XML.» peut être représentée par l'élément

```
<phrase>"<![CDATA[<alt>2882</alt>]]>" est un élément XML.</phrase>
```

dans lequel la suite de caractères `<alt>2882</alt>` est reconnue comme une suite de caractères de donnée et non comme l'élément `<alt>2882</alt>`.

I.1.5 Appel d'entité

Dans le texte source d'un document XML ou d'une DTD, un appel d'entité peut remplacer un morceau de document XML qui apparaît fréquemment, qui serait trop long à saisir ou qui est enregistré dans un autre fichier.

Une entité est une association, déclarée dans une DTD (voir paragraphe I.2), entre un nom, le nom de l'entité, et une suite de caractères, le contenu de l'entité. On distingue : les entités générales et les entités paramètres. Le contenu d'une entité générale est une suite de caractères qui doit pouvoir constituer tout ou partie du contenu d'un élément ou d'une valeur d'attribut. Le contenu d'une entité paramètre est une suite de caractères qui doit pouvoir constituer tout ou partie d'une déclaration d'élément, d'attribut, d'entité ou de notation.

Si *n* est le nom d'une entité déclarée dans la DTD d'un document XML et *s* est son contenu.

– `&`*n*`;` est un appel d'entité générale qui peut remplacer *s* dans le contenu d'un élément ou dans une valeur d'attribut de ce document ;

– `%`*n*`;` est un appel d'entité paramètre qui peut remplacer *s* dans cette DTD.

C'est le processeur XML qui remplace les appels d'entité par leur contenu : on dit que ces appels sont résolus. Cette résolution est un processus récursif puisqu'un contenu d'entité peut lui-même contenir des appels d'entités. Ce n'est qu'une fois les appels d'entité résolus que le processeur vérifie si le document est bien formé et valide.

Par exemple :

▷ si `crampons-utiles` est une entité dont le contenu est

```
<note type="matériel">Crampons utiles</note>
```

l'élément :

```
<para>...(&crampons-utiles;).</para>
```

sera transformé en :

```
<para>...(<note type="matériel">Crampons utiles</note>).</para>
```

après résolution de l'appel d'entité `&crampons-utiles;`.