

Glossaire

Accessoire polymorphe

Cette locution est relative au modèle SOAP utilisé dans les services web. Elle désigne une manière de transmettre, sous forme XML, des tableaux à n dimensions contenant un nombre variable de types de données. Ce mécanisme original n'est supporté naturellement ni par XML ni par XML Schema.

Adressage

On désigne par adressage tout ce qui concourt à connaître et exploiter le lieu de stockage d'une donnée ou d'une ressource. Le plus souvent, il s'agit d'un chemin d'accès, mais cela peut prendre des formes plus sophistiquées en XML, par exemple avec Xpointer ; technique qui permet de combiner un chemin d'accès dans un système de fichiers avec un chemin d'accès à l'intérieur d'un document XML.

Agrégation

Terme issu du monde UML, une agrégation est un type d'association entre deux classes exprimant un rapport entre un tout et ses parties. La classe formant le tout est appelée « agrégat », l'autre classe désigne les parties. Les agrégations indiquent ainsi un rapport structurel entre les classes.

Par exemple, on considérera une famille comme un tout et les membres de la famille comme les parties.

Voir composition.

Ambiguïté

L'ambiguïté est l'incapacité à dire si un document XML est valide ou non. Le plus souvent, les validateurs, ou parseurs, pourraient lever l'ambiguïté en allant voir plus en avant ce que le document XML contient ; cela s'appelle le *lookahead*, mais c'est interdit par l'ensemble des normes et recommandations SGML et XML.

Les expressions « modèles ambigus » de SGML, « modèle de contenu non déterministe » de XML 1.0 et « règle de la particule unique » désignent le même problème d'ambiguïté dont XML Schema donne la définition suivante :

« Un modèle de contenu doit être formé de telle manière que, pendant la validation d'une séquence d'unités d'information de type élément, la particule, contenue directement, indirectement ou implicitement, avec laquelle on essaie de valider l'une après l'autre chaque unité de la séquence, doit pouvoir être identifiée sans avoir à examiner le contenu ou les attributs de chaque unité et sans avoir besoin de faire appel aux informations relatives aux autres unités du reste de la séquence. »

Appel d'entité

Cette locution appartient aux mondes SGML et XML. Un appel d'entité désigne l'utilisation d'un nom d'entité dans un document XML. Par exemple, soit une entité à laquelle on donne le nom `texte`, alors l'appel d'entité correspondant sera la chaîne de caractère `&texte;`.

```
<!DOCTYPE sample [  
<!ENTITY texte "Ceci est un paragraphe utilisant un appel d'entité.">  
.....  
<root>  
<p>&texte;</p>  
</root>
```

Applicabilité

L'applicabilité est l'identification de l'objet auquel s'applique une information. La gestion de l'applicabilité peut prendre des formes simples, tel un simple nom de système, ou complexes lorsque ledit système possède des variantes techniques (« applicable pour les outils de la marque untel... »), temporelles (« applicable du tant ou tant... »), calendaires (« applicable à partir de telle date... ») et géographiques (« applicable aux systèmes produits en Espagne... »). Ces critères d'applicabilité peuvent être combinés entre eux.

Arbre, arborescence

Le concept de classification de l'information sous la forme d'arbre n'est pas nouveau. Tous les systèmes mécaniques ou logiques sont tôt ou tard, dans leurs phases de conception, décomposés en objets élémentaires prenant la forme d'une arborescence.

En XML, l'organisation arborescente de l'information est évidente, mais dans les représentations UML des modèles de données, une telle représentation arborescente n'est pas un postulat de base. En effet, une représentation arborescente n'est suffisante ni pour représenter la manière dont les données interagissent les unes avec les autres, ni les liens entre ces mêmes données.

Architecture

Par analogie avec l'art de construire des édifices, l'architecture en informatique désigne les principes et techniques d'organisation des systèmes informatiques ; par exemple, l'architecture en trois tiers, l'architecture de service, etc.

Articulation

Par analogie avec le corps humain, l'articulation est un mécanisme qui donne de la souplesse à un modèle :

- Le modèle peut intégrer facilement d'autres modèles.
- Le modèle peut valider des documents XML de plus petite taille que le modèle lui-même.

Attributs flottants

Un attribut est dit flottant quand on peut le mettre n'importe où dans la structure sans déclaration préalable. Actuellement, XML Schema permet d'y recourir seulement pour les attributs de l'espace de noms XML Schema Instance, reconnu par le préfixe `xsi` qui lui est souvent associé. Les trois attributs les plus connus de XML Schema Instance sont `xsi:type`, `xsi:noNamespaceSchemaLocation` et `xsi:schemaLocation`. Un autre attribut flottant connu est `xmlns` qui définit un préfixe d'espace de noms.

Cela devrait également devenir le cas à terme avec les attributs de XLink.

Backbone

Encore appelé structure d'assemblage ou squelette, un backbone est un document XML dont la seule vocation est de définir un assemblage (ordonné ou non) de documents XML. Il s'agit donc essentiellement d'un document XML qui contient des pointeurs vers d'autres documents XML.

Base de données

Le nom complet en est *Système de gestion de base de données*, ou SGBD. Expression trop longue à dire et écrire, l'habitude a été prise d'utiliser la forme raccourcie *base de données* ; en revanche, le sigle SGBD est, quant à lui, resté.

Tenant compte de ce contexte, une base de données est un logiciel capable de gérer un volume, en général important, de données. Gérer signifie ici : contrôler les imports et exports de données, l'accès aux données, leur mise à jour, leur cohérence, et offrir des outils de calcul capables de s'appuyer sur la structure logique des données pour réaliser des opérations (calculs arithmétiques, booléens, recherche par motifs lexicaux ou valeurs...).

Base de données XML

Il s'agit d'une base de données dédiée au modèle XML. Cela signifie que les données sont stockées dans des documents XML, que les fonctions d'extraction et de mise à jour sont celles définies pour les données XML (XQuery, Xupdate, DOM, XSLT, SAX), que les méthodes d'accès sont compatibles avec un protocole tel que SOAP et que les standards définis par le W3C ou l'ISO sont tout ou partie mis en œuvre.

Il existe :

- des bases « natives XML » développées sur le seul modèle XML ;
- des bases relationnelles dans lesquelles le XML est projeté (on dit « mappé »), ces bases étant restrictives car les modèles XML et relationnels ne sont pas égaux ;
- des bases XML qui utilisent un moteur objet ;
- des bases XML qui s'appuient sur des bases hiérarchiques.

Cardinalité

On appelle cardinalité le nombre d'éléments d'un ensemble. En UML et XML, cela représente le nombre de fois qu'un objet peut être utilisé relativement à un autre.

Carte ou cartographie de sujets, de topiques

Expressions utilisées pour traduire les termes Topics Map. Elles désignent un modèle XML bien particulier dont la vocation est de définir les relations qui unissent des ressources informatiques, physiques ou virtuelles, ainsi que les descripteurs de ces dernières.

Cible

Une cible est la ressource informatique, physique ou logique pointée par un lien, une relation ou tout mécanisme de référencement ou de liaison.

Classe

Une classe est un ensemble concret ou logique qui contient tous les objets d'un même type, voire d'un type dérivé ; on parle alors de classes et de sous-classes.

La classe est concrète lorsqu'elle est définie une bonne fois pour toutes par programme, ou elle peut être logique lorsqu'on fabrique, à la volée, des ensembles d'objets ayant des caractéristiques communes.

Clé

Dans une base de données, la même information est parfois dupliquée ou croisée. La notion de clé est dès lors un mécanisme permettant de connaître la donnée de référence et de faire la différence avec celles qui sont simplement des copies de cette donnée de référence. Une clé peut être composée de plusieurs données qui, réunies, forment un identifiant unique d'une chose. Le mécanisme des clés existe tant pour les bases de données relationnelles que XML (on peut définir des clés avec XML Schema).

Clé artificielle

Clé ou élément de clé n'étant pas une donnée propre de l'application mais un identifiant quelconque attribué de manière mécanique à un ensemble de données. Par exemple, un simple numéro incrémental permet d'identifier les rangées d'un tableau, et ainsi de les différencier.

Clé étrangère

Une clé étrangère est tout simplement une information servant de clé ou partie de clé, mais reprenant la valeur d'une clé stockée ailleurs. Le principe des clés étrangères existe également dans XML Schema.

Composant ou constructeur

Les composants sont les représentations concrètes des fonctions définies dans XML Schema : il s'agit des éléments du vocabulaire de XML Schema utilisés par un auteur qui écrit un schéma XML. Remarquez que le terme « composant » ne désigne pas les attributs de ces éléments mais seulement les éléments.

Composition

Terme issu du monde UML, une composition est une association de type « agrégation » où l'existence des objets formant les parties est liée à celle de l'objet formant le tout. La suppression de l'objet parent – le tout – entraîne celle des objets reliés par l'association de composition – les parties.

Conceptuel

On utilise le mot conceptuel dans les expressions *modèle conceptuel*, *modèle conceptuel de données*. Il désigne la représentation idéale de la chose que l'on cherche à maîtriser, l'idée que l'on cherche à transmettre, les données que l'on souhaite organiser, et les traitements et programmes qui feront les contrôles et manipulation des données. En informatique, le modèle conceptuel des données est à un programme ce que le plan est à un immeuble.

Contrôle lexical

Le contrôle lexical consiste à vérifier si les caractères significatifs du langage XML sont utilisés correctement. Par exemple, une valeur d'attribut s'écrit entre deux guillemets droits " et non entre chevrons «.

Contrôler que les caractères utilisés dans les chaînes de caractères des noms, valeurs et données du document XML sont conformes aux définitions lexicales de leurs types fait partie du contrôle lexical. Par exemple, un nom d'élément doit commencer par une lettre, un souligné ou un deux-points, et tout autre caractère est interdit. Ainsi, des noms d'éléments lexicalement valides sont `para`, `_para` et `:para`, et des exemples de noms invalides sont `1para`, `#para`, etc. Vous pouvez remarquer qu'il est inutile de disposer d'un schéma pour contrôler la validité lexicale de la syntaxe d'un document XML. En revanche, le schéma est impératif pour contrôler la validité lexicale des contenus d'éléments et d'attributs.

Contrôle grammatical

Le contrôle grammatical porte sur la bonne utilisation du vocabulaire : dans l'ordre et à bon escient.

On distingue deux niveaux de contrôle grammatical :

- L'un consiste à contrôler que le schéma est conforme au langage d'écriture de schémas utilisé. Par exemple, en XML Schema, le contrôle grammatical doit vérifier qu'une définition d'élément (constructeur `xs:element`) n'utilise pas simultanément les attributs `name` et `ref`.
- L'autre consiste à contrôler que les éléments et attributs d'un document XML respectent les règles définies par le schéma dont dépend le document XML. Par exemple, l'élément suivant un titre doit être un `para` qui doit lui-même être qualifié par un attribut `date`.

Contrôle syntaxique

La syntaxe désigne l'ensemble des caractères et mots significatifs du langage. La racine grecque du mot syntaxe, *suntaxis*, veut dire « ranger ensemble » (*sun*,

ensemble, et *tassein*, ranger). La syntaxe est donc la partie de la grammaire qui traite du rôle et de la disposition des mots et propositions dans une phrase.

De même qu'en français, le point, la virgule et les autres signes de ponctuation jouent indéniablement un rôle dans l'expression écrite – ils sont remplacés par les intonations dans l'expression orale –, les documents XML et leurs schémas utilisent des caractères significatifs qui leur sont propres. Cela se remarque immédiatement en observant les extraits de ces langages représentant une même définition de modèle :

En XML 1.0 :

```
<!ELEMENT schema ((%include; | %import; | %redefine; | %annotation;)*,
(%simpleType; | %complexType; | %element; | %attribute; |
%attributeGroup; | %group; | %notation; ),
(%annotation;)*)* )>
<!ATTLIST schema targetNamespace CDATA #IMPLIED >
```

En XML Schema :

```
<xs:element name="schema">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:openAttrs">
        <xs:sequence>
          <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element ref="xs:include"/>
            <xs:element ref="xs:import"/>
            ...
          </xs:choice>
          <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:group ref="xs:schemaTop"/>
            <xs:element ref="xs:annotation" minOccurs="0"
              maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:sequence>
      </xs:extension>
      <xs:attribute name="targetNamespace" type="xs:anyURI"/>
      ...
    </xs:complexContent>
  </xs:complexType>
  ...
</xs:element>
```

Les DTD comportent un grand nombre de caractères syntaxiques que les schémas XML remplacent par du vocabulaire. Les caractères significatifs des DTD, notam-

ment les connecteurs, tels que parenthèses, virgules, barres verticales, etc. et les symboles d'occurrences (point d'interrogation, astérisque), sont remplacés dans les schémas par des noms d'éléments. Ces changements résultent de la volonté d'utiliser dans XML Schema la syntaxe XML. Un schéma s'écrit comme n'importe quel document XML tandis qu'une DTD a son propre langage.

Crochet

Un élément crochet est un élément XML conçu de telle sorte qu'il peut servir de point de raccordement d'un schéma XML à un autre (voir aussi anneau).

Data-module code

C'est le code d'identification d'un module de données. Cette codification obéit à des règles strictes qui permettent de donner un sens métier à ce code.

Dérivation

La dérivation est une opération au moyen de laquelle on crée un type à partir d'un autre. Il s'agit d'un terme utilisé dans le monde des schémas XML. Comme des objets, les types dérivés les uns des autres forment un chaînage qui possède ses propres contraintes de fabrication et d'utilisation.

Diagramme de classes

Le diagramme de classes fait partie de la terminologie UML et désigne la représentation graphique des classes d'un système de données et les relations (et cardinalités) qui les relient les unes aux autres. Le diagramme des classes est celui qui se rapproche le plus des besoins de représentation des schémas XML. Il n'est malheureusement pas parfait pour cet usage.

Document électronique

On appelle document électronique tout objet informatique pouvant être apparenté à un document, au sens traditionnel du terme : un document papier, une photo, une vidéo, une séquence sonore, peuvent être des documents à partir du moment où ils sont porteurs de sens.

L'informatique offre désormais des zones d'ombre : un courrier électronique, un fichier PréAO (Présentation assistée par ordinateur), un objet multimédia, un fichier CAO (Conception assistée par ordinateur), un échange par messagerie instantanée, sont-ils des documents ? La réponse est affirmative si on considère que ces objets prennent du sens par rapport à un événement particulier. Par exemple, on peut considérer que le courriel historique de John Bosak du 22 novembre 1996 à 14 h 06'58" annonçant la naissance de XML est un document. On peut également considérer

que tout fichier qui ne contient pas des données (au sens d'un alignement de valeurs atomiques) est un document.

Document papier

Un document papier a la caractéristique d'avoir une dimension physique finie : par exemple, une feuille de papier A4 et des contraintes de mise en page (les en-têtes et pieds de page par exemple). Aussi les deux caractéristiques principales d'un document papier sont-elles sa dimension et la typographie qui y est utilisée, parfois de manière très sophistiquée.

Que le document soit papier, ou « papier électronique » comme PDF, ne change pas le problème : pour que l'information soit compréhensible, claire et facile à lire, il faut qu'elle soit composée. Cela fait une grande différence avec les documents dématérialisés tels que des pages HTML pour lesquelles une qualité très moyenne est souvent suffisante (on parle ici du cas de l'information technique et non des pages HTML à caractère marketing où le graphisme est souvent très élaboré).

Document structuré vs non structuré

Autrefois, tout document était qualifié de « *non structuré* » par opposition aux données, souvent stockées dans des tables, qui, de leur côté, étaient qualifiées de « *structurées* ».

Avec XML, ce distinguo n'a plus lieu d'être : un document textuel est parfaitement structuré.

Document textuel vs données

Une donnée est une valeur atomique qui n'a pas de sens en soi. Un tableau de chiffres (des dates par exemple) n'a jamais donné de sens à ces données. En revanche, il est possible de manipuler ces données (faire des calculs, des extractions, des tris, etc.).

À l'opposé, un document est a priori un objet permettant de donner de la valeur aux données. Ce faisant, et avant que XML n'arrive, il n'était plus possible de manipuler les données.

Un document XML a dès lors l'avantage de pouvoir contenir indifféremment des données et/ou des textes sur lesquels les manipulations et transformations sont toujours possibles. On est alors amené à parler de « données données » et « données textuelles ».

Document XML

Est baptisé document XML tout fichier contenant des données balisées XML respectant les règles de bonne forme. Le mot document n'est donc pas ici à prendre dans son sens commun.

Documentation modulaire structurée

L'expression de documentation modulaire structurée désigne une méthode de conception de la structure d'un document (ou de tout autre objet) par laquelle des modules représentent des composants textuels ou graphiques élémentaires. Les modules contiennent des données balisées et ils sont assemblés entre eux selon un ordonnancement logique, lui-même défini au moyen d'un modèle XML.

DTD

Le mot DTD est l'acronyme de deux choses différentes : d'une part *définition de type de document* et d'autre part *déclaration de type de document*.

Remarquez que, dans les deux cas, il faut dire « la » DTD et non « le » DTD comme on le voit trop souvent écrit.

La définition de type de document est le fichier qui contient des définitions d'éléments, d'attributs, d'entités et de notations. C'est en général ce fichier qui est désigné quand on parle de « *la DTD du document XML* ».

La déclaration de type de document est la carte que l'on trouve dans le prologue d'un document XML et qui s'écrit : `<!DOCTYPE toto SYSTEM "toto.dtd">`. Cette déclaration spécifie l'élément racine du document et lui associe la « DTD » qui convient.

Effectivité

L'effectivité complète les notions d'applicabilité. C'est une information qui indique le niveau d'obsolescence d'une autre information, en particulier d'une publication. Par exemple, un module d'information peut avoir une applicabilité aux systèmes A, B et C, mais peut faire partie d'une publication dédiée, quant à elle, au seul système B. Le uplet {A,B,C} représente l'applicabilité du module tandis que le singleton {B} représente l'effectivité de la publication.

Élément de données

Un élément de données ne contient que des données (numériques ou textuelles peu importe) ou un modèle qui peut être assimilé à de la donnée (cas d'un élément mixte par exemple).

Élément de données global

Un élément de données global contient une collection d'éléments de données, potentiellement mêlés à des données textuelles, voire des éléments purement structurels.

Élément flottant

Un élément est dit flottant quand on peut le mettre n'importe où dans une structure. C'était possible avec SGML grâce au mécanisme des exceptions ; cela ne l'est plus avec la version actuelle de XML. Le mouvement s'inversera peut-être un jour.

Élément global

Il s'agit d'une terminologie XML Schema qui désigne la qualité d'un élément à être utilisé comme élément racine d'un document XML et, par référence, dans plusieurs autres éléments XML (voir également élément local ci-après).

Élément local

Il s'agit d'une terminologie XML Schema désignant un élément qui ne peut être utilisé ni comme racine d'un document XML ni par référence à l'intérieur de plusieurs autres éléments. Un élément XML local n'appartient qu'à un seul élément XML parent (voir également élément global ci-avant).

Élément mixte

Un élément mixte contient, sur le même niveau hiérarchique, un contenu textuel et des sous-éléments. Un tel élément est typique de documents XML représentant des documents papier.

Élément purement structurel

Un élément purement structurel ne contient aucun élément de données comme enfant direct.

Élément racine

Un élément racine est le premier élément d'un document XML. Un fichier renfermant un document XML ne peut contenir qu'un seul élément racine.

Ensemble d'informations

Un ensemble d'informations est tout simplement un document XML. Cette locution, qui est la traduction de l'anglais *infoset*, est en train de rentrer dans le vocabulaire courant de XML et permet avantageusement d'éviter d'utiliser le mot *document*, parfois trop proche des notions de document papier.

Ensemble d'informations bien formé

Un ensemble d'informations bien formé, ou *well-formed infoset*, est un ensemble d'informations qui vérifie les critères de bonne forme de XML.

Entité

Dans le vocabulaire de XML, une entité est une ressource informatique que l'on peut spécifier à l'aide d'une déclaration d'entité. Une entité est typiquement soit une chaîne de caractères, soit un fichier XML, ASCII ou binaire.

Espaces de noms, espaces de nommage

Les espaces de noms, ou encore *de nommage*, servent à définir les limites de validité des vocabulaires définis par les schémas XML. Ainsi, une même balise `<prix>` peut avoir deux significations différentes selon qu'elle appartient à un espace de noms A ou B, et on écrira pour les distinguer `<a:prix>` et `<b:prix>`.

Feuille

Une feuille est un élément terminal d'une arborescence de données XML, ou arbre XML. À pas confondre avec une feuille de papier !

Forme

La forme est l'allure que prend un ensemble de données lorsqu'il est écrit en utilisant une syntaxe particulière. La forme XML se reconnaît aisément grâce à ses balises et attributs emboîtés les uns dans les autres.

Forme canonique

La forme canonique d'un document XML enlève les ambiguïtés résultant des différentes formes lexicales que peut avoir un document XML, et ce afin de pouvoir simplifier les applications de traitement des documents XML et même comparer de tels documents deux à deux. Les parseurs et les validateurs sont censés être capables de produire les formes canoniques des documents qu'on leur soumet.

Voici les caractéristiques d'une forme canonique :

- utilisation du seul codage UTF-8 ;
- normalisation des valeurs d'attributs et repositionnement des attributs dans l'ordre alphabétique de l'initiale de leur nom ;
- suppression du prologue du document ;
- changement de notation des éléments vides, écriture de `<gi></gi>` au lieu de `<gi/>` ;
- intégration des sous-documents : les appels d'entités externes analysées sont remplacés par les entités désignées ;
- homogénéisation des déclarations de préfixes d'espaces de noms ;
- suppression des préfixes d'espaces de noms des éléments et attributs où ils sont inutiles ;
- inclusion des attributs par défaut.

Toutes les questions relatives à la canonisation ne sont pas réglées et ce sujet reste en discussion.

Forme lexicale et forme lexicale canonique

La forme lexicale est celle que l'on peut voir en consultant un document XML avec un éditeur ASCII, donc tel qu'il est enregistré sur disque, sans interprétation aucune des balises. La forme lexicale du chiffre 3 peut être "03.00" ou "3" ou " 3 ", cela ne change pas la valeur numérique de ce contenu. XML Schema spécifie des formes lexicales canoniques pour chaque type (dates, durées, nombres entiers, flottants, chaînes de caractères, etc.).

Gestion de configuration

La gestion de configuration consiste à gérer la version de chaque composant entrant dans la conception d'un système. Dans le cas de publications, il s'agit d'identifier chaque objet entrant dans la composition du document, qu'il s'agisse d'éléments textuels ou graphiques.

Gestion de contenu

Cela concerne la gestion des composants entrant dans la composition des pages web. Un outil de gestion de contenu permet de modéliser les pages d'un site web, produire les contenus textuels, gérer les liens entre les pages et avec les illustrations, et surtout les règles d'affichage en fonction de profils d'utilisateurs ou d'événements autres.

Gestion de l'applicabilité

Voir applicabilité.

Gestion de versions

Cela consiste à gérer des variantes d'une version de base. Cela peut se faire par recopie des fichiers ou par modification à l'intérieur du document XML. Un système doit permettre de conserver l'historique des liens avec le fichier d'origine, et éventuellement de provoquer des alertes quand le fichier d'origine est modifié.

Gestion des révisions

La gestion des révisions consiste à stocker l'historique des fichiers. La gestion des révisions comprend toujours deux dimensions : d'une part une gestion des éditions (un numéro s'incrémente à chaque fois qu'une modification mineure est effectuée) et d'autre part la gestion des révisions (après une série d'éditions, une version finale est créée qui consolide toutes les éditions).

Gestion des connaissances

La gestion des connaissances (*Knowledge Management* en anglais) consiste à gérer des documents, des informations sur ces documents et une partie de l'activité qui accompagne un domaine particulier. Ainsi, si des forums de discussion sont ouverts sur des thèmes particuliers, il est intéressant d'archiver les propos échangés pour les retrouver un jour, et parfois même très longtemps après. Les savoir-faire, associés à des profils d'utilisateurs, sont rangés dans des classeurs thématiques virtuels auxquels on accède de plusieurs manières.

Gestion électronique de documents

La gestion électronique des documents concerne les systèmes dédiés au contrôle des cycles de production des documents. Plus largement, la GED concerne tout système destiné à contrôler un référentiel de documents. Pour cela, les documents sont accompagnés de métadonnées qui servent à les classer, les identifier, et fournir un résumé ou des mots-clés relatifs à leur contenu.

Grammaire

Une grammaire est constituée des règles de positionnement des noms d'éléments et d'attributs les uns par rapport aux autres. Un schéma XML définit un vocabulaire (ensemble des noms des éléments et d'attributs) et une grammaire.

Comme dans une langue naturelle, en XML la grammaire sert à définir la logique d'assemblage des mots. Chaque schéma définit une logique d'assemblage des éléments et des attributs.

Héritage

L'héritage est un concept qui vient du monde des objets. Ces derniers font partie de classes qui sont toujours créées à partir de classes parents. Ainsi, les propriétés et méthodes d'une classe sont pour partie héritées de celles de leurs parents.

Héritage et dérivation

Quand on veut modifier les caractéristiques, ou les méthodes, héritées d'une classe parent, on opère ce qui s'appelle une dérivation.

Identifiant, identificateur

L'identifiant d'un élément est ce qui permet de l'identifier de manière unique : il s'agit d'une clé simple composée d'une seule chaîne de caractères. Avec XML 1.0, le seul type d'identifiant autorisé était le type ID. XML Schema permet de définir, quant à lui, de vraies clés qui sont des up]ets prenant la forme {A,B,C...} (voir le mot *clé* dans ce glossaire).

Infrastructure UML 2.0

Il s'agit du premier livre de la spécification UML 2.0. Il définit les principes fondateurs servant de base à la création de tous les modèles de UML 2.0.

Voir superstructure UML 2.0.

Illustration code number

Comme le code d'identification des modules (voir *data-module code*), celui destiné aux illustrations est un identificateur unique pour une application métier. À la différence d'un module, ce code ne peut pas être inscrit dans le fichier lui-même et doit obligatoirement correspondre à un nom de fichier ; il ne peut donc s'agir que d'une chaîne de caractères.

Infoset

Un infoset, littéralement *ensemble d'informations*, fait l'objet de la recommandation du W3C XML Infoset datée du 24 octobre 2001. Il s'agit de la nouvelle terminologie utilisée pour désigner un document XML. Cette recommandation a un rôle très important. Grâce à elle, les spécifications du modèle de documents XML et des schémas qui servent à les contrôler sont, pour la première fois, radicalement séparées. En effet, la norme ISO 8879 (SGML) et la recommandation XML 1.0 du W3C mêlaient, jusqu'alors, la définition du modèle de document XML et celle du langage d'écriture de leurs schémas : les DTD.

XML Infoset ne définit aucun langage d'écriture de schéma. Cette recommandation spécifie qu'un document XML est un ensemble d'informations composé d'unités d'information et de contenus textuels.

Nous avons choisi dans cet ouvrage de ne pas utiliser les expressions *infoset* et *ensemble d'information*, mais celle de *document XML*.

Si l'on sait habituellement qu'un document XML est composé des unités d'information de type élément et attribut, on sait moins en revanche qu'il peut contenir jusqu'à neuf autres types d'unités d'information, soit onze au total. L'objectif de XML Infoset est de donner une définition précise de chacun d'eux.

Ces onze types d'unités d'information sont les suivants :

- la déclaration de type de document ;
- le document, c'est-à-dire l'élément racine du document XML ;
- l'élément ;
- l'attribut ;
- l'espace de noms ;
- l'instruction de traitement ;

- le commentaire ;
- la référence à une entité externe ;
- l'entité externe ;
- le caractère ;
- la notation.

Les logiciels et textes normatifs qui se disent conformes à XML Infoset doivent prendre en charge la totalité de ces onze types d'unités d'information.

Les langages d'écriture de schémas de documents XML ne concernent qu'une partie de ces unités d'information : celles qui font la structure et le vocabulaire d'un document XML – les éléments, les attributs, les espaces de noms, et dans une moindre mesure les notations.

Sur le seul point de vue logique, référencer les entités externes utilisées par un document XML à partir de son prologue, c'est-à-dire dans un complément à la DTD, est une anomalie de conception de ces langages que les nouvelles approches, XML Schema et RelaxNG, corrigent avec XLink et XInclude.

A contrario, des informations définies dans les schémas n'apparaissent pas en tant qu'unités d'information dans XML Infoset. Il s'agit par exemple des types de données. Le fait qu'une chaîne de caractères soit un nombre entier, une date ou une forme lexicale particulière n'est pas une information censée être contenue dans le document XML lui-même.

Ainsi, il est aujourd'hui flagrant qu'un document XML et son schéma sont deux sources d'information indépendantes et complémentaires.

Voilà pourquoi le concept de PSVI, *Post Schema Validation Infoset*, a été inventé. Il s'agit d'un document XML qui contient l'infoset de départ, canonisé et augmenté des informations issues de son schéma. Ainsi, le PSVI contient les informations sur le type des données, les modèles auxquels appartiennent les éléments, la nature des attributs (imposé, obligatoire, par défaut...). Le PSVI peut être considéré comme une sorte de fichier XML exporté dans lequel on aurait à la fois le document XML d'origine et la totalité des informations issues de son schéma et le concernant. Le PSVI est un cas particulier d'ensemble d'informations.

Instance

Une instance est un objet créé à partir d'un modèle et conforme à ce dernier. Dans le cas de SGML, les instances sont les documents SGML conformes à une DTD mais, en XML, le mot instance a disparu au profit des expressions *document XML valide* et *ensemble d'informations XML valide*. En ce qui concerne UML, et le monde des objets plus généralement, l'instance (d'une classe) est un objet qui possède les comportements de la classe à laquelle il appartient. Cet objet est alors le composant

effectif des programmes (c'est lui qui calcule) alors que la classe est plutôt une définition ou une spécification de l'ensemble des instances à venir.

À notre grand étonnement, nous avons appris que ce mot a été refusé à plusieurs reprises par l'Académie française, malgré son usage répandu parmi les informaticiens et sa présence dans le grand dictionnaire terminologique québécois de la langue française (<http://www.olf.gouv.qc.ca/ressources/gdt.html>).

Langage de liaison

Un langage de liaison sert à écrire (établir) des liens ; il s'agit typiquement de XLink.

Lien entre documents vs intra-document

Un lien entre documents part d'un document XML pour aller dans un autre tandis qu'un lien intra-document reste limité au document XML où se trouve la source du lien. Dans ce deuxième cas, un simple identifiant suffit, tandis que dans le premier il faut ajouter, au minimum, le chemin d'accès du deuxième document. On notera enfin que dans le cas du lien entre documents, le document dans lequel se trouve la cible du lien devient subordonné au premier : on ne peut le déplacer sans intervenir sur celui où se trouve la source.

Liste des pages effectives

Une liste effective des pages est, en quelque sorte, une table des matières des pages d'une publication. Concept réservé aux documents techniques sensibles, la liste effective des pages permet de lister chaque page de l'ouvrage avec, en regard, son indice de révision et son statut (effective ou détruite). Une telle liste est l'un des moyens de contrôle des révisions d'une publication.

Métadonnées

Une métadonnée est une donnée sur une donnée : par exemple, spécifier que le nombre 1590575118235 est un numéro de sécurité sociale nécessite de faire appel à une métadonnée (en l'occurrence celle qui précisera le type de ce nombre). Dans le monde des documents, les métadonnées sont toutes les informations nécessaires à la gestion administrative dudit document (numéro ISBN, noms des auteurs, dates de parution, etc.).

Métalangage

Un métalangage est un langage qui sert à écrire d'autres langages. Si on considère que le vocabulaire (noms des éléments, des attributs et de leurs valeurs possibles) défini par une DTD ou un schéma XML est un langage, alors le langage utilisé pour écrire de tels modèles est un métalangage. La syntaxe des DTD pouvant être modi-

fiée, SGML les définissait à l'aide d'une syntaxe abstraite ; on avait vraiment là un métalangage.

Métamodèle

Un métamodèle est le père d'un modèle, on dit aussi le modèle d'un modèle. Le terme de méta fait référence au niveau d'abstraction supplémentaire induit naturellement par la position de « père de modèle ». Pour XML, il existe deux branches parentes : celle de la syntaxe et celle des schémas.

Dans la branche de la syntaxe, le modèle père d'un document XML est celui qui définit la syntaxe concrète d'écriture des documents XML (en l'occurrence la recommandation Infoset), tandis que le métamodèle (le grand-père) est celui qui définit les relations parent-enfant, le fait que les éléments aient des propriétés (les attributs), la notion de prologue, etc. Ce métamodèle est visible dans la norme SGML ISO 8879:1986.

Dans la branche des schémas, il existe plusieurs modèles pères (XML Schema, RelaxNG et Schematron) et plusieurs métamodèles qui sont les modèles conceptuels ayant servi à définir les vocabulaires (et les sémantiques associées) de chacun de ces langages de modélisation. XML Schema et RelaxNG ont le même métamodèle : celui des langages à base de grammaire. Schematron est le résultat d'un métamodèle différent : celui des langages à base de règles.

En valeur absolue, tout modèle servant à produire d'autres modèles est un métamodèle.

Modèle

Le mot modèle a deux sens. Il peut désigner le moule qui permet de reproduire à l'identique une même chose comme il peut désigner une représentation simplifiée d'un ensemble de choses. Par exemple : un cylindre marron, des racines, une boule verte et des points rouges à l'intérieur représentent tous les pommiers de la terre ; un cercle jaune et des rayons : c'est le soleil.

Modèle et schéma conceptuels

Il existe deux types de modèles conceptuels. L'un concerne les langages d'écriture de schémas, l'autre les données.

En ce qui concerne les langages, un modèle conceptuel est la définition théorique du langage, soit le préalable nécessaire à la définition d'une forme concrète du langage. Un langage – par exemple le langage d'écriture de schémas XML du W3C – doit être défini de manière théorique avant d'exister sous une forme concrète, ou syntaxique. XML Schema est défini d'une part par des règles de production et d'autre part par des explications en langage naturel. Quant à lui, RelaxNG est formellement défini par des formules mathématiques dites règles d'inférence.

En ce qui concerne les données, un modèle conceptuel permet de les représenter ainsi que les relations qui les unissent indépendamment de la manière selon laquelle elles seront physiquement organisées, et notamment stockées. Un schéma conceptuel est dès lors une application d'un modèle conceptuel de données pour une application particulière. UML propose un modèle conceptuel de représentation des données. Un diagramme de classes UML réalisé pour une application particulière représentera le schéma conceptuel des données de cette application.

Modèle logique

Un modèle logique représente la manière selon laquelle les données seront réellement mises en œuvre pour une application. Un schéma XML, une DTD, sont des modèles logiques : ils représentent la manière dont seront réellement organisées les données, et notamment la manière dont elles seront stockées les unes par rapport aux autres.

Par exemple, on peut avoir un modèle conceptuel représentant des familles composées de parents et d'enfants : la présence d'une relation parent-enfant est *de facto* évidente. Ce modèle conceptuel ne signifie pas pour autant que les éléments XML correspondants seront <famille>, <père>, <mère>, <fille> et <fils>, et que ces derniers seront des enfants des éléments <père> et <mère>. Le modèle logique retenu *in fine* pour représenter les familles pourra être totalement différent des sous-entendus du modèle conceptuel. On pourrait même très bien avoir des éléments <individus> et <relation> permettant de lier les individus entre eux selon une logique de relations familiales.

Modèle vs schéma, langage de modélisation

Rick Jelliffe, auteur de la spécification Schematron, fait une distinction entre les termes « schémas » et « modèles ». Selon lui, un schéma définit de manière exhaustive et complète une réalité tandis qu'un modèle n'en est que la meilleure représentation possible.

Nous soutenons cette approche que nous expliquons par une analogie : quand un homme sert de *modèle* aux autres, il est un point de repère, un guide, un exemple à suivre, mais quand on affirme suivre le *schéma* de pensée d'une personne, c'est qu'on déroule à l'identique un même scénario intellectuel. De même que le sculpteur dispose d'un modèle et d'une technique pour réaliser son œuvre, les modèles et les schémas sont deux notions nécessaires et complémentaires pour produire un document XML.

Dans le domaine technique, on peut dire d'une voiture qu'elle est de type break quand on peut la comparer à un objet de référence qui a ce type : un autre break montré en exemple. Cet objet de référence est le modèle qui nous sert à reconnaître

le type d'une autre voiture. La définition exacte de chaque voiture est cependant constituée des dessins techniques qui ont servi à la fabriquer : les plans, ou schémas.

Le mot schéma est donc à prendre dans le sens que lui donnent les Américains, celui de plan très précis d'un objet. Nous avons suivi ce *distinguo* dans cet ouvrage et avons choisi d'utiliser la terminologie suivante :

- Le mot *schéma* est utilisé pour désigner l'ensemble des règles définissant une classe de document. Un schéma est typiquement une DTD ou un fichier contenant un document XML Schema ou RelaxNG.
- Le mot *modèle* désigne la chose, ou l'idée, que l'on veut représenter. Cela est très différent du schéma. Il en est ainsi des idées ayant prévalu à la définition concrète des schémas et des documents XML. En effet, tout document XML peut avoir un modèle, à savoir un document XML exemple montrant ce qu'est la syntaxe XML et mettant en scène des éléments, des attributs et d'autres unités d'informations. Toutefois, seul un schéma permettra de savoir si un document XML en particulier est bon ou pas pour une application spécifique. Il existe toutefois des exceptions « historiques », telle l'expression « modèle de contenu d'un élément » qui désigne, dans une DTD, la définition d'un élément. Dans un schéma XML, le vocabulaire est plus précis, et on préférera parler du type de l'élément.
- L'expression *langage de modélisation* n'est pas utilisée ; nous lui préférons *langage d'écriture de schémas*. De la même manière, l'expression *langage de schématisation* n'est pas utilisée : elle choque sans correspondre à la réalité qu'on voudrait lui donner. En ce qui concerne les modèles, nous ne parlerons que de représentation des modèles.
- Le mot *métamodèle* est utilisé pour désigner le modèle du modèle, il désigne bien souvent les concepts purement intellectuels ayant prévalu à la définition d'un modèle. Il existe par exemple un métamodèle pour XML qui est constitué du concept d'objet (les types), de propriétés (les attributs) et d'héritage (les relations parent-enfant créées par les emboîtements des éléments forment des lignées d'héritages sémantiques par le seul jeu des noms d'éléments).

Module

Dans le présent ouvrage, un module est un document XML destiné à faire partie d'un ensemble plus vaste. Le module n'a pas de taille prédéfinie, il peut s'agir d'un paragraphe comme d'un chapitre entier : c'est son auteur qui la détermine. Un module est géré à l'aide de métadonnées.

Nœud

Un nœud est, dans un arbre représentant une structure XML, un embranchement.

Objet métier

Dans cet ouvrage, un objet métier est un élément XML dont le nom est directement interprété par les programmes de traitement de l'information. Par exemple, si l'élément `<dateDeValeur>` est un objet métier, il sera associé à un programme de même nom.

Occurrence et nombre d'occurrences

L'occurrence est la répétition d'une même chose. Aussi, le *nombre d'occurrences* est le nombre de fois qu'une chose apparaît. En XML, la notion d'occurrence intervient dans les modèles (DTD et schémas) car c'est là que le concepteur précise combien de fois un élément est autorisé ; on parle alors du *nombre d'occurrences*. En UML, ce *nombre d'occurrences* est représenté par la multiplicité.

Ontologie et thésaurus

Le dictionnaire Larousse donne de ces mots les définitions suivantes :

Ontologie : (gr. *On, ontos*, être, et *logos*, science). Discours issu de la logique mathématique et de la linguistique, qui traite des termes utilisés pour désigner les êtres constitutifs de la réalité.

Thésaurus : (gr. *Thesaurus*, trésor). Dictionnaire destiné à aider les recherches dans certaines disciplines et contenant, pour chaque mot-clé, les termes similaires ou synonymes.

Une ontologie est un dictionnaire de mots doublés de définitions de relations entre eux. À la différence d'un dictionnaire normal dont le seul sens de lecture est le sens alphabétique, les relations permettent à une ontologie d'avoir plusieurs sens de lecture.

De son côté, un thésaurus est comparable à une ontologie réduite à un seul domaine et un seul type de relation entre les mots : la relation (hiérarchique) d'appartenance à un domaine. Un thésaurus est donc un dictionnaire augmenté d'une seule structure additionnelle : une classification hiérarchique des mots par domaine.

Parseur (voir validateur)

PSVI (voir la section Infoset)

Persistence

La persistance s'applique aux données qui doivent être stockées pour « persister » au-delà du temps d'exécution d'un programme. Dans les applications non standardisées, la persistance pose simultanément le problème du format de stockage des données (des champs de fichiers ASCII séparés par des tabulations ? des champs de base de données ?, etc.) et de leur lieu de stockage. Cependant, avec XML, la question du format de stockage ne se pose plus (car les normes relatives à XML sont très précises

sur la question). Ne subsiste que la question du lieu de stockage du document XML (dans des arborescences de fichiers, au sein d'une base de données XML, etc.), et éventuellement du vocabulaire qui sera utilisé.

Polymorphe

Littéralement, se dit d'un objet qui présente plusieurs formes. Dans le monde XML, un objet est dit polymorphe lorsque le sens de la donnée qu'il contient dépend du contexte dans lequel l'objet est baigné. Par exemple, le fragment `<date>2004-08-27</date>` prend un sens différent selon que l'élément parent est `<Paiement>` ou `<Facturation>`, par exemple. Aussi, dans la conception de schémas XML, le choix des noms attribués aux éléments XML peut ne pas être anodin. Un élément qui s'appellerait `<dateDePaiement>` serait moins polymorphe car il garderait une bonne partie de son sens même isolé de son contexte.

Polymorphisme

Le polymorphisme concerne les messages envoyés aux objets : un même message peut être envoyé à différents objets qui, en fonction de leur type, rendront des réponses différentes. En XML, le polymorphisme concerne les éléments ayant un même nom mais des types différents (c'est possible avec XML Schema).

Publication module code

Le code d'un module de publication est l'identifiant unique attribué à la structure d'assemblage d'une publication. En général, cet identifiant est également visible dans la publication finale.

Référence

Une référence est un identifiant permettant de connaître la cible d'un lien.

Référence d'entité

En XML, une référence d'entité (on devrait dire *référence à une entité*) est un identifiant renvoyant à une déclaration d'entité.

Réification

La réification est un concept relatif aux cartes de topiques. C'est l'instanciation physique, concrète, d'une abstraction, en particulier un sujet. Quand on met un sujet dans un certain contexte, il devient un topique : on dit que le sujet est réifié.

Ressource

Une ressource est toute information électronique accessible à partir d'une adresse URL. À ne pas confondre avec les mémoires d'ordinateur, imprimantes et autres systèmes physiques également appelés ressources dans le monde informatique.

Schéma conceptuel

Voir modèle conceptuel.

Schéma à base de grammaire

Un schéma à base de grammaire définit la syntaxe d'un vocabulaire XML en totalité : les noms des éléments, des attributs, leur type, parfois leur contenu imposé. Contrairement aux schémas à base de règles, ces schémas ne permettent pas aux documents XML d'être partiellement valides.

Voir aussi la rubrique *Métamodèle*.

Schéma à base de règles

Un schéma à base de règles définit des règles que le document XML doit vérifier et qui peuvent tout aussi bien porter sur la présence (un certain nombre de fois) d'un élément que sur une corrélation entre la valeur d'un attribut et le contenu d'un élément. Contrairement aux schémas à base de grammaire, ces schémas permettent d'avoir des documents XML « partiellement » valides. Expliquons ce que l'on entend par partiellement valides : certes, toutes les règles édictées par le schéma doivent être vérifiées, mais il n'est pas obligé que ces règles portent sur tous les éléments du document XML.

Voir aussi rubrique *Métamodèle*.

Schémas

Voir la rubrique *Modèle vs schéma, langage de modélisation*.

Schematron

Langage d'écriture de modèle XML à base de règles. Les règles de Schematron sont écrites principalement en s'appuyant sur le langage Xpath.

Sémantique

La sémantique est le sens profond d'un nom, par rapport à son contexte d'utilisation. La sémantique des éléments, de leurs attributs, représente le sens qu'ils ont par rapport à l'application qui les utilise. Plus la sémantique est précise et moins l'élément est polymorphe.

Sérialisation

Le mot sérialisation désigne la transformation qui consiste à produire un fichier XML physique à partir de sa représentation en mémoire. XML est tel qu'il n'existe aujourd'hui pas une seule représentation possible mais plusieurs. La sérialisation consiste à reproduire physiquement dans un fichier le sens de lecture d'un document XML en conservant les codes permettant de comprendre sa structure arborescente, ainsi que toutes les autres unités d'information autorisées.

La sérialisation est donc la traduction d'un sens de parcours de l'arbre. Nous allons expliquer ce propos à l'aide de trois figures inspirées du livre de Ronald Bourret, *Mapping DTDs to Databases*, paru en 2000 aux éditions O'Reilly.

Il existe quatre manières d'envisager le sens de parcours d'un arbre :

- l'ordre défini par le ciblage des éléments ;
- l'ordre hiérarchique ;
- l'ordre du document ;
- l'ordre des classes d'objets.

Pour les illustrer, nous utiliserons le fragment suivant :

```
<para>IDENTIFICATION BY CUSTOMER<custname>Air Canada</
custname><cus>ACN</cus><xref refid="ibc001"></para>
```

L'ordre de ciblage correspond à la notion de fratrie, qui se définit comme étant l'ensemble des éléments d'un même niveau hiérarchique. Pour le fragment précédent, cet ordre est illustré à la figure E-1.

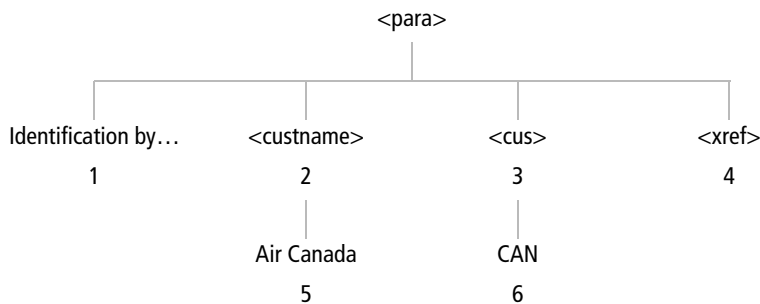


Figure E-1 Représentation graphique de l'ordre des éléments selon la règle du ciblage

L'ordre hiérarchique correspond à la profondeur des éléments dans l'arborescence définie par la structure depuis la racine. Cet ordre est illustré à la figure E-2.

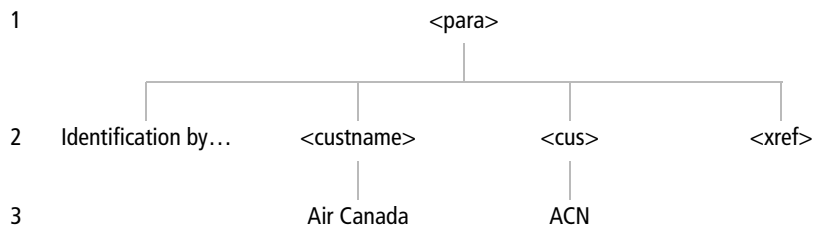


Figure E-2 Représentation graphique de l'ordre des éléments selon la règle hiérarchique

L'ordre du document est celui qui représente l'ordre des éléments dans le sens de la lecture. Cet ordre est illustré à la figure E-3.

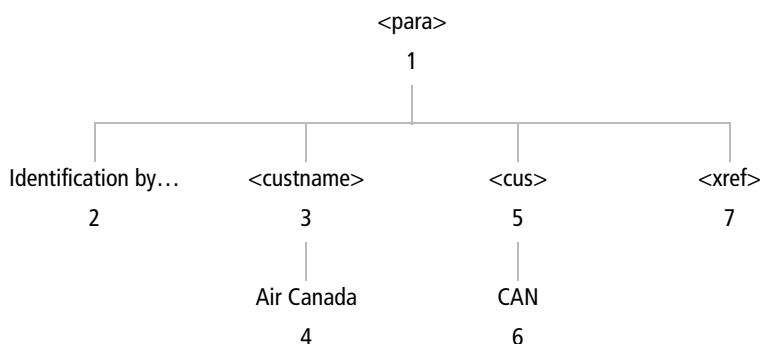


Figure E-3 Représentation graphique de l'ordre des éléments selon la règle du document

Dans les applications orientées objet telles que DOM (Document Object Model), la quatrième manière de parcourir une arborescence d'objets, c'est de cibler les objets d'une même classe. Si les balises <para>, <custname> et <cus> représentent trois sous-classes de la classe des éléments et si Identification by ..., Air Canada et CAN représentent trois nœuds de l'ensemble des nœuds textuels, on peut envisager de balayer cet ensemble d'objets en sautant de paragraphe en paragraphe, de lien en lien, de nœud textuel en nœud textuel, etc. Ce mode de lecture est notamment l'une des caractéristiques de XPath. Ce mode de parcours ne fait pas ici l'objet d'une illustration car il ouvre la voie à autant de possibilités qu'il existe de classes d'objets dans un document XML, et car il n'est jamais utilisé dans le but de stocker un document XML.

Sérialisé

Un document XML est sérialisé lorsqu'il a été enregistré physiquement. Quand les données sont stockées en mémoire sous la forme d'un DOM (Document Object Model), la sérialisation va de soi mais quand il s'agit d'objets, la sérialisation est le résultat de la transformation des champs et propriétés publics d'un objet, ou les paramètres et valeurs de retour des méthodes, en un flux XML conforme à un schéma XML spécifique. C'est typiquement ce qui se passe lors des opérations de sérialisation de SOAP.

Serveur de ressources

La notion de serveur de ressource intervient lorsque les liens entre ressources sont définis logiquement, sans utilisation d'adresses physiques. Au moment de la résolution du lien, il faut bien transposer l'identifiant logique en adresse physique. On y procède par programme ou par un serveur de ressources qui a la capacité de recevoir un identifiant logique et de retourner l'URL physique de la ressource correspondante.

Source

Dans notre ouvrage, la source désigne toujours la source d'un lien ; à ne pas confondre donc avec un code source. La source d'un lien est la ressource qui contient le point de départ de ce lien.

Squelette de publication

Un squelette de publication est une ossature écrite en XML qui spécifie les composants d'une publication, leur ordre d'apparition, ainsi que les numéros de révision des composants utilisés.

Structure d'assemblage

Voir squelette de publication.

Superstructure UML 2.0

Il s'agit du deuxième livre de la spécification UML 2.0. Sur les bases des principes fondateurs définis dans le livre *infrastructure UML 2.0*, il définit l'ensemble des concepts de modélisation proposés par UML 2.0.

Voir infrastructure UML 2.0

Syntaxe abstraite

On appelle syntaxe abstraite une syntaxe dans laquelle les caractères spécifiques sont remplacés par des variables. Ainsi, une telle syntaxe laisse la porte ouverte à plusieurs

implémentations. Quand les variables sont remplacées par les caractères définitifs, on obtient une syntaxe concrète.

Syntaxe concrète

Voir syntaxe abstraite.

Système d'information

Un système d'information représente l'ensemble des moyens humains et technique mis en œuvre pour gérer les informations nécessaires au fonctionnement d'une entreprise. Il couvre les grandes fonctions de création, archivage, transformation et diffusion de l'information. À ce titre, il fait appel aux techniques informatiques de gestion des données, des traitements et des communications, ainsi qu'aux méthodes de mise en œuvre de ces techniques.

Taxinomie

Étude théorique des bases, lois, règles, principes d'une classification. Le terme est utilisé pour désigner des classifications de mots.

Synonyme de taxonomie.

Taxonomie

Voir taxinomie.

Topics Map

Voir la rubrique *carte de topiques*.

Topique

Voir la rubrique *carte de topiques*.

Type d'un élément

Le type d'un élément est la définition de la nature de son contenu. En XML, le type peut être simple, l'élément ne contenant qu'une chaîne de caractères, ou complexe quand l'élément renferme des sous éléments et/ou des attributs.

Les chaînes de caractères peuvent elles-mêmes être de plusieurs types : dates, nombres, durées, booléens, etc.

Parmi les types complexes, on trouve les types mixtes (texte et sous-éléments mêlés) et vide (l'élément ne contient rien).

Unité d'information

Une unité d'information est un module.

À ne pas confondre avec une *unité documentaire*, expression parfois utilisée pour désigner un ensemble de modules intermédiaire entre un module et une publication.

Voir la rubrique *module*.

Urbanisation

L'urbanisation est une discipline de gestion des systèmes d'information visant à aligner les fonctions requises par les divisions opérationnelles de l'entreprise avec l'architecture logicielle et technique des systèmes informatiques. L'objectif en est de définir les règles d'équilibre entre la dynamique requise par les processus toujours mouvants de l'entreprise et la nécessaire structuration des données et des programmes propre aux techniques de développements informatiques.

Validation de schéma

La validation de schéma est une opération par laquelle un document XML est contrôlé par rapport au schéma qu'il est censé respecter. Quand le contrôle est positif, il est possible de produire un PSVI, ou Post Schema Validation Infoset, qui est un ensemble d'informations XML contenant la totalité de l'information : le document XML et son schéma associé (attention, un PSVI n'est pas qu'une simple concaténation des deux mais une fusion).

Valideur et parseur

Un valideur est un programme dont le rôle est de contrôler la conformité d'un document XML par rapport à un schéma.

Un parseur – en français analyseur lexico-syntaxique – est un programme qui a pour rôle de vérifier la validité lexicale et syntaxique d'un document XML.

À l'époque où seules les DTD existaient, le parseur contrôlait le vocabulaire, les caractères et les structures utilisés dans un document XML. La tendance actuelle est d'avoir plusieurs types de validation : conformité d'un document XML par rapport à la recommandation XML Infoset, conformité par rapport à un ou plusieurs schéma(s)...

Le « parsing » ou analyse lexico-syntaxique, est le contrôle de la validité lexicale et syntaxique d'un document XML : les caractères utilisés dans le document XML sont-ils autorisés ? la syntaxe XML est-elle respectée ? le document est-il bien formé ?

La validation, c'est le contrôle de la conformité d'un document XML par rapport à un schéma. La validation peut ajouter des informations dans le document XML – par exemple, ajout des valeurs d'attributs par défaut – et en modifier le contenu – par exemple, écriture homogène des nombres et suppression des blancs anormaux.

Validation sémantique

La validation sémantique consiste à contrôler que des éléments, des attributs ou des valeurs de contenus sont utilisés correctement par rapport à leur sens.

La validation sémantique s'applique aux documents XML comme aux langages d'écriture de schémas.

La sémantique peut être définie formellement – c'est le cas de RelaxNG *via* des règles d'inférence –, ou par des explications en langage naturel – c'est le cas des schémas XML du W3C.

Concernant les données, la sémantique peut être contrôlée par rapport à des thésaurs, des ontologies, voire des types.

Vocabulaire (voir aussi grammaire)

Un vocabulaire est l'ensemble des noms d'éléments et d'attributs définis par un schéma XML.

Web sémantique

Le concept de Web sémantique vise à faire prendre conscience de l'importance de la question du sens dans les pages web, et ce afin de pouvoir les retrouver et de donner une représentation de ce sens qui soit exploitable par des agents logiciels. La simple recherche en plein texte ne suffit pas. Les pages web doivent être classées, triées par thème, afin que les mots qu'elles contiennent aient du sens pour des requêtes intelligentes. Quand on cherche à « contacter un avocat sur Paris », on n'est pas vraiment satisfait de tomber sur tous les marchands de primeur de la capitale !