Tirème Sarl

DTD, Définition de type de document

Rédaction : **Pierre Attar**

Recommandation(s) liée(s) : DSDL - RELAX NG - Schema - XML - XMI

La notion de DTD est partie intégrante de la spécification XML, héritière de SGML, mais délestée de beaucoup de fonctionnalités aujourd'hui superflues.

Pour des *parsers* "validants", la notion de DTD dans XML permet de définir un modèle de données qui servira ensuite à valider, de façon électronique, toutes les instances supposées être conformes au modèle. Pour des *parsers* "non validants", la notion de DTD sert simplement à fournir un réservoir permettant de décrire où sont des fichiers qui doivent être inclus dans les documents XML au travers d'entités générales.

Le modèle lui-même permet de spécifier une hiérarchisation d'objets typés et *valués* selon des attributs. Il permet encore, au travers des entités paramètres, de modulariser l'écriture des DTD. Ce modèle est largement implémenté par tous les outils de création et de modification de documents issus du monde SGML.

Objectifs

L'objectif d'une DTD est de pouvoir définir un modèle de données formel, sous forme électronique, afin que les outils puissent valider, de façon automatisée, la conformité d'une classe de documents XML à ce modèle.

Cette activité est nécessaire pour tous les processus de traitement de l'information XML, qui doivent pouvoir présupposer de ce qui est contenu dans un document. Elle est aussi extrêmement utile pour les processus d'édition interactive de documents XML qui, s'auto-adaptant à une DTD, pourront aider l'utilisateur dans ses activités de création et de modification de structure et dans ses activités de validation.

Par ailleurs, l'objectif d'une DTD est de pouvoir définir tout ce qui est nécessaire comme ressources à tous les documents d'une classe donnée. Si un document XML doit inclure des fragments externes, ceux-ci seront définis et identifiés de façon formelle au sein de la DTD, afin de pouvoir ensuite être appelés au sein des documents (mécanisme des entités *parsées* générales).

Si le modèle devait recenser tous les fragments appelables dans un document, le fichier issu de ce modèle deviendrait extrêmement imposant et surtout très difficile à maintenir. Du coup, les concepteurs de la recommandation autorisent l'utilisation de DTD, complétées de déclarations n'affectant qu'un seul document : on parle alors des entités *parsées* générales de l'*Internal Entity* .

Principes

Modèles de documents

Une DTD permet de définir une organisation d'éléments entre eux et des typages de ces éléments via des notions d'attribut (de *valuation*).

Plus précisément, une DTD définit un modèle d'organisation hiérarchique de documents XML en utilisant :

- *des éléments*, organisés hiérarchiquement. Pour ce faire, un modèle de contenu est défini pour chaque élément. Le modèle de contenu peut faire appel aux notions d'organisation d'éléments entre eux (séquence, choix ou agrégat), en utilisant des indicateurs d'occurrence (optionabilité, répétabilité). Il peut se réduire aussi à de simples caractères. Lorsqu'il définit un regroupement de caractères et d'éléments, on parle alors de contenus mixtes (*mixed contents*). Dans ce dernier cas, des règles très strictes sont définies ;
- *des attributs*, qui permettent de valuer un élément. Les attributs sont typés ou énumérés. Ils peuvent avoir une valeur fixe par défaut ou laissée au libre choix des outils de traitement ;
- d'autres informations qui peuvent être ajoutées aux éléments et aux attributs, telles les *notations*, les *entités paramètres ou générales*, etc.

La syntaxe utilisée pour écrire des DTD est une syntaxe *ad hoc*. Par exemple, déclarer un élément `article` contenant une en-tête et un corps s'écrira :

```
<!ELEMENT article (tete, corps) >
```

Cette syntaxe est concise et simple, mais présente le grand désavantage de ne pas être celle utilisée pour écrire des documents XML et, donc, de ne pas pouvoir être analysée par des outils XML standard. C'est une des raisons qui amène à la définition des nouveaux modèles Schema.

DTD et entités

Les DTD permettent aussi de définir des entités qui implémentent le mécanisme d'inclusion cher à tous les langages de programmation. *Générale et parsée*, l'entité définit des fragments XML que le *parser* doit inclure lors de la validation d'un document (que ce soit pour vérifier s'il est "bien formé" ou s'il est valide au regard de son modèle) ; le contenu du fragment appelé doit lui-même être un arbre bien formé. Si elle est *paramètre*, l'entité définit des fragments de DTD à intégrer lors du *parsing* de la DTD elle-même.

L'entité peut être externe au document. Par exemple :

```
<!ENTITY fichier "-//MUTU-XML//Un fichier externe//FR" "http://
www.monsite.fr/fragment.xml">
```

définit une entité générale (un fragment XML bien formé), qui se nomme `-//MUTU-XML//Un fichier externe//FR` et qui se trouve à l'adresse `http://www.monsite.fr/fragment.xml`.

L'entité peut être interne. Par exemple :

```
<!ENTITY xml "eXtended Markup Language">
```

définit un fragment XML directement dans la déclaration interne liée au document.

Ce mécanisme d'inclusion est extrêmement utile pour toutes les opérations de modularisation et de réutilisation. Il souffre cependant d'une lacune, qui est l'obligation de définir dans le document, d'une part, un nom logique et, d'autre part, une localisation physique. Cette lacune existait aussi dans les premiers temps de SGML et était fort contraignante, dès lors que des documents étaient échangés et que les organisations de fichiers n'étaient pas les mêmes chez l'émetteur et le récepteur d'un document. (Par exemple, un développeur de site Web utilisera souvent des adresses basées sur ses propres disques locaux, voire basées sur son site de test. Lorsqu'il publiera l'information, les adresses seront alors basées sur le site réel.)

Recommandations(s)

Extensible Markup Language (XML) 1.1 (Second Edition)
Recommandation, version 1.1, du 16-08-2006
Document sur http://www.w3.org/TR/xml11#sec-prolog-dtd