

TIRÈME SARL

XInclude, XML Inclusions

Rédaction : Pierre Attar

Recommandation(s) liée(s) : [Infoset](#)

Beaucoup de langages de programmation fournissent un mécanisme d'inclusion, afin de faciliter la modularité. Cette proposition présente un mécanisme générique pour fusionner des documents [XML](#) dans un même et unique document [XML](#).

Objectifs

L'objectif de [XInclude](#) est de permettre de modulariser l'écriture de documents [XML](#) et d'atteindre des objectifs de réutilisation de fragments partagés.

Utiliser [XInclude](#) relève d'un processus basé sur la forme *désérialisée*, en [Infoset](#), de documents [XML](#). Plus précisément, *réaliser l'inclusion n'est en aucun cas le rôle d'un parser*, mais plutôt celui d'un processus, déclenché, si nécessaire, une fois générées les [Infoset](#) du document contenant les inclusions. Le résultat de ce processus sera un nouveau jeu d'[Infoset](#), résultant de la fusion du jeu primaire et de tous les jeux des objets à inclure. En ce sens, tous les processus applicatifs pourront décider quelle vue ils souhaitent obtenir du document primaire : une vue avec inclusions identifiées ou une vue avec inclusions réalisées.

Principes

Le mécanisme est différent de celui proposé par [XLink](#), car indépendant des applications. En effet, dans [XLink](#), ce que doit générer, en termes de comportement, un processeur de l'attribut `show= 'embed'`, n'est pas spécifié de façon obligatoire.

Le mécanisme est aussi différent de celui utilisé par les [DTD](#) pour les entités externes, pour deux raisons principales :

- ce sont les parseurs qui font la résolution des entités. En ce sens, l'information disponible dans le modèle abstrait de données est soit inexistante, soit extrêmement minimale. Cette information est, pourtant, par moment, intéressante, lors de la mise à jour de documents dont les différents morceaux sont à stocker dans différents documents partagés ;
- les notions d'entités générales nécessitent l'utilisation, même minimale, de [DTD](#), alors qu'actuellement, beaucoup d'utilisations de [XML](#) se font sans [DTD](#) et qu'il est à prévoir que ce processus s'amplifiera, dès lors que les [Schema](#) seront standardisés.

D'un point de vue technique, pour utiliser ce mécanisme d'inclusion, il faut se situer dans l'espace de noms <http://www.w3.org/1999/XML/xinclude>. Ceci fait, deux attributs assurent l'inclusion, étant posés sur n'importe quel élément :

- `xinclude:href` : permet de définir l'objet à inclure. Cela peut être un document (`something.xml` ou une portion de document (`#xpointer(x/myinclude[1])`) ;
- `xinclude:parse` : définit la méthode d'inclusion pour savoir s'il faut *parser* (valeur `xml`) ou non (valeur `text`) l'**InfoSet** inclus.

Pour conclure, **XInclude** souffre de la même faiblesse que le système de gestion des entités externes dans les **DTD** : le mécanisme de désignation des fichiers est contenu dans les documents **XML** eux-mêmes. On suivra alors avec intérêt la façon dont **XInclude** évoluera, dès lors que les propositions de catalogue externe (**XML Catalogs**) seront mises en place.

Recommandations(s)

■ ■ *Les inclusions XML (XInclude)*

Recommandation, version 1.0, du 20-12-2004

Document sur <http://www.yoyodesign.org/doc/w3c/xinclude/#>

🇺🇸 *XML Inclusions (XInclude)*

Recommandation, version 1.0 (Second Edition), du 15-11-2006

Document sur <http://www.w3.org/TR/xinclude/>