

Vocabulaires et domaines d'utilisation

Vocabulaires horizontaux

- CML (p. 3)- MathML (p. 4)- OASIS Tables (p. 18)- SVG (p. 22)- VoiceXML (p. 24)

Domaines verticaux

Métadonnées

- BiblioML (p. 25)- Dublin Core (p. 26)- EAD (p. 28)- HyTime (p. 29)- MADS (p. 31) - MODS (p. 32)- OAI (p. 33)- RDF (p. 35)- RDF-S (p. 39)- OWL (p. 42)
- RDFa (p. 44) - Topic Maps (p. 46) - XTM (p. 48)

Multimédia

- SMIL (p. 49)

Administration

Administration française

Industrie

Documentation technique après-vente

Industrie de l'informatique et de l'électronique

Documentation technique

- DITA (p. 50) - DocBook (p. 53)

Interfaces graphiques

- UIML (p. 57) - XUL (p. 58)

Industrie automobile

- J2008 (p. 59)

Défense

- S1000D (p. 61) - FOSI (p. 64)

Commerce électronique

- ebXML (p. 65) - BizTalk (p. 66) - EDI (p. 68)

Finance

- XBRL (p. 69)

Statistique

- SDMX (p. 70) - GESMES (p. 71)

Publications

Documents

- ISO 12083 EMS (p. 72)- NewsML (p. 73)- Open Office XML (p. 74)- Open XML (p. 75) - TEI (p. 77)

· Livres électroniques

- OEB (p. 78) - ePub (p. 80)

· Méta modèles de modélisation

- UML (p. 81) - XMI (p. 82)

TIRÈME SARL

CML, Chemical Markup Language

This DTD is fully described in Journal of Chemical Information and Computer Science, Vol xxx, 1999.

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

CML

Recommandation, version 1.01, du 05-2001

Document sur <http://sourceforge.net/projects/cml/dtdschema/index.html>

Design Considerations for CML

Expression des besoins, version 0, du 2002??

Document sur <http://sourceforge.net/projects/cml/information/design.html>

TIRÈME SARL

MathML, Mathematical Markup Language

Rédaction : Ulric Genièvre, Yannick Litaiz, Wojciech Machocki,

Ludovic Maurillon, Benoît Roger, Sébastien Vallée, **Pierre Attar**

MathML est un langage de représentation des mathématiques. Plus précisément, **MathML** permet de coder une forme sémantique et/ou typographique d'un objet mathématique. En ce sens, **MathML** est un compromis entre les défenseurs du "tout sémantique" (le groupe *OpenMath*, par exemple) et le "tout typographique" (la communauté *TeX* et *LaTeX*).

La partie typographique de **MathML** se justifie par la spécificité de la notation typographique mathématique où prime l'ambiguïté, par souci pédagogique aussi bien que par souci de densification de l'information. Ainsi, un article de mathématique parlant de matrices n'aurait aucun intérêt à toujours coder la sémantique sous-jacente ! En effet, quel effort de saisie faudrait-il alors mobiliser pour seulement parler de "*la matrice M précédemment définie*" ! Pourtant, M est bien un objet mathématique ou, plutôt, un objet typographique mathématique.

L'enjeu de **MathML** est sa reconnaissance ! Dans sa version 2, la spécification semble stable et pouvoir coder bon nombre de besoins... il reste alors à comprendre comment sera réalisée son implémentation, particulièrement sur le Web. En effet, si, depuis Knuth et son moteur de composition *TeX*, "écrire sur papier" des mathématiques ne pose aucun problème, il n'en va pas de même sur Internet. Avec **MathML**, un standard existe maintenant. Pour qu'il puisse être largement utilisé, il faut qu'il soit implémenté dans tous les logiciels de consultation, de manière presque native, du fait de l'étroite inter-relation entre la typographie mathématique et le texte.

Les logiciels *Open Source* peuvent aussi aider au développement de **MathML**. À ce titre, toutes les explications données ci-après sont extraites d'un mémoire d'étudiants en 5^e année de Génie mathématique (*INSA de Rouen*) [*sur internet : ./images/rapportMathML.pdf*], qui ont pris l'initiative de proposer à **Pierre Attar** la libre disposition de leur mémoire, dans le but de participer à une plus large connaissance de **MathML** et de ses outils.

Objectifs

MathML est un format de définition de mathématique codé en **XML**, dont le but est d'identifier à la fois la typographie et/ou le sens mathématique. **MathML** doit permettre aux mathématiques d'être échangées, reçues et traitées sur le Web, comme **HTML** l'a permis pour le texte. Parallèlement, il doit permettre à des logiciels de calculs formels comme

Maple ou *Matlab*, ainsi qu'à des applications classiques développées dans des langages standard (C, Java, LISP, etc.) d'échanger et d'interpréter des objets mathématiques complexes.

Pour ce faire, les objectifs de **MathML** sont :

- encoder de la substance mathématique adaptée à la communication scientifique et mathématique de tout niveau ;
- encoder la notation typographique et/ou la sémantique sous-jacente ;
- faciliter la conversion de (et vers) d'autres formats de présentation et/ou de sémantique mathématique.

Les formats et applications utilisant **MathML** doivent aussi permettre de *tirer* de ce langage :

- des représentations graphiques ;
- des synthèses vocales ;
- des données traitables par les systèmes algébriques ;
- d'autres représentations dans d'autres langages mathématiques (e.g. TeX) ;
- des représentations "plain text" ;
- des affichages divers, y compris le braille.

Les formats et applications utilisant **MathML** doivent encore permettre :

- de supporter les longues expressions à la navigation ;
- le passage d'information voulue pour des "rendeurs" spécifiques et des applications ;
- d'être étendu.

Enfin, les formats et applications utilisant **MathML** doivent être :

- adaptés aux *templates* et autres techniques d'édition ;
- "humainement" lisibles ;
- faciles à gérer et à traiter par les logiciels.

Le W3C Math Working Group [*sur internet* : www.w3.org/Math/] a identifié des objectifs supplémentaires d'implémentation, dont le but est de décrire rapidement les fonctionnalités élémentaires que les moteurs de "rendu" et les logiciels de traitement **MathML** devraient fournir :

- les équations dans les pages **HTML** doivent avoir un rendu correct dans les navigateurs, en accord avec les préférences d'affichage des auteurs et des lecteurs, et avec une qualité optimale au vu des capacités de chaque plate-forme ;
- les pages **HTML** contenant des équations **MathML** doivent être imprimées correctement, à la résolution optimale de l'imprimante (*idem* pour le braille, les synthétiseurs vocaux, etc.) ;
- les équations **MathML** dans une page Web doivent être capables de réagir au comportement de l'utilisateur (clic d'un souris par exemple) et de gérer la communication avec d'autres applications, à travers le navigateur ;
- des éditeurs d'équation et des "convertisseurs" doivent être développés pour faciliter la création de pages Web contenant des équations **MathML**.

Il est reconnu que la conversion de (ou vers) d'autres systèmes de notation ou médias peut entraîner une perte d'information.

En outre, comme nous l'avons dit, l'un des objectifs de **MathML** est la diffusion d'information mathématique sur Internet. Par conséquent, **MathML** se doit d'être en relation avec les autres technologies Internet connues. Il faut plus particulièrement que l'on puisse être en mesure de :

- convertir les langages de balises mathématiques existants vers **MathML** ;
- ajouter du **MathML** dans les pages **HTML**, afin que les applications Web qui manipulent, dès aujourd'hui, du **HTML** puissent, dans le futur, manipuler facilement du **MathML** ;

"rendre" du [MathML](#) dans les navigateurs existants de quelque façon que ce soit (e.g. par images), même si cela est loin d'être idéal.

Note. Rappel : cette partie consacrée aux objectifs de [MathML](#) est issue du travail rédigé par des étudiants de 5^e année de Génie mathématique, à l'INSA de Rouen : "Conversion TEX et rendu MathML" [sur internet : [./images/rapportMathML.pdf](#)]. Merci à Ulric Genièvre, Yannick Litaize, Wojciech Machocki, Ludovic Maurillon, Benoît Roger, Sébastien Vallée d'avoir décidé de façon volontaire et active de participer au développement de l'information sur le glossaire de Tirème.

Principes

Syntaxe et grammaire

[MathML](#) est une application [XML](#). Par conséquent, sa syntaxe est en partie dictée par la syntaxe [XML](#) et sa grammaire est spécifiée par une [DTD](#). Ainsi, nous retrouvons les éléments classiques de la syntaxe [XML](#), à savoir :

- les caractères [Unicode XML](#) (incluant les caractères ASCII ordinaires) ;
- les attributs ;
- les entités ;
- la structure de ses éléments ;
- la casse des chaînes de caractères.

Cependant, le [W3C](#) a trouvé, du fait de sa volonté d'encourager la modularisation des applications construites avec [XML](#), que la forme actuelle d'une [DTD](#) ne convenait pas exactement : un groupe fut créé afin de développer des spécifications pour les [Schema](#). [MathML](#) étant conçu pour que les mathématiques puissent tirer avantage des dernières évolutions technologiques en matière de Web, il doit donc y avoir un [Schema](#) pour [MathML](#), ce qui est le cas.

Par ailleurs, [MathML](#) définit aussi certaines règles de syntaxe et de grammaire qui lui sont propres, lui permettant d'encoder davantage d'information, sans pour autant introduire beaucoup plus d'éléments et sans utiliser une [DTD](#) ou un [XML](#) plus complexes. Evidemment, l'un des inconvénients de ces règles spécifiques à [MathML](#) est qu'elles sont totalement transparentes pour un processeur [XML](#) classique. Ces règles sont de deux types :

- des critères sur la valeur des attributs ;
- des restrictions sur les *éléments fils* (des règles d'ordre, par exemple).

L'ordonnancement des *éléments fils* est très pratique (voire indispensable) dans bon nombre de cas, car il évite des problèmes de compréhension sans surcharger la syntaxe [XML](#).

Exemple : Une fraction est constituée de deux éléments principaux : le numérateur et le dénominateur. [MathML](#) fournit donc une balise `mfrac` possédant seulement deux fils pouvant être de tous types, le premier étant obligatoirement le *numérateur*, le deuxième étant obligatoirement le *dénominateur*.

$$\frac{a}{b}$$

s'écrit donc :

```
<mfrac>
  <mi>
```

```

      a
    </mi>
  <mi>
    b
  </mi>
</mfrac>

```

Les informations supplémentaires sur la valeur des attributs résident essentiellement sur le typage de ces valeurs. Ce problème peut être géré en préférant les [Schema](#) aux [DTD](#). Il peut aussi être traité lors du parsing, en demandant à l'analyseur de vérifier la validité des données passées en valeur d'attribut. Le non-respect de ces règles entraîne une erreur [MathML](#), bien qu'il ne gêne par le parsing [XML](#).

Les notations utilisées pour décrire les types de valeurs sont les suivantes :

number	entier ou rationnel, pouvant commencer par l'opérateur binaire "-"
unsigned-number	entier ou rationnel non signé
integer	entier, pouvant commencer par l'opérateur binaire "-"
positive-integer	chaîne de caractères quelconques (toujours la totalité de valeur de l'attribut)
string	chaîne de caractères quelconques (toujours la totalité de valeur de l'attribut)
character	caractère seul, quelconque (excepté l'espace), ou une entité.
#rrggb	code RVB d'une couleur
h-unit	unité de longueur (la liste des unités est donnée ci-après)
v-unit	unité de hauteur (la liste des unités est donnée ci-après)
css-fontfamily	unité CSS
css-colorname	unité CSS
form+	une ou plusieurs instances de « form »
form*	zéro ou plusieurs instances de « form »
f1 f2 ... fn	une instance de chaque fi, séparée ou non par des espaces
f1 f2... fn	n'importe lequel des fi spécifiés
[form]	zéro ou une instance de « form »
(form)	1 instance de « form »
mot « plain text »	ce mot, littéralement présent dans la valeur de l'attribut
quoted symbol (e.g. '+' ou "+")	ce symbole, littéralement présent dans la valeur de l'attribut

L'ordre de priorité des opérateurs de cette syntaxe est la suivante (du plus fort au plus faible) :

form+ ou form* ,

`f1 f2 ... fn` (séquence),
`f1|f2...|fn` (alternative).

Dans la valeur des attributs, les mots-clés et les nombres doivent être séparés par des espaces, exceptées les "unités" (`h-unit`, `v-unit`) qui suivent les nombres. Pour les valeurs des attributs, l'espace n'est pas particulièrement requis, mais il est autorisé dans toutes les expressions exposées précédemment, excepté devant une unité (pour une comptabilité avec [CSS1](#)), entre le signe "-" et la valeur du nombre négatif, et entre # et `rrggbb`.

Le signe "+" n'est pas autorisé dans la syntaxe d'un nombre entier ou rationnel positif, sauf s'il est explicitement demandé.

Comme nous l'avons dit dans le tableau précédent ci-dessus, quelques attributs acceptent comme valeur des hauteurs et des largeurs (`h-unit` et `v-unit`), qui sont généralement formés d'un nombre suivi d'une unité. `h-unit` et `v-unit` se réfèrent donc à une unité pour exprimer respectivement la hauteur et la largeur lors du rendu des balises. Les unités possibles sont :

<code>em</code>	relatif à la largeur de la police courante
<code>ex</code>	relatif à la hauteur de la police courante
<code>px</code>	pixels (1 pixel = 2.54 cm)
<code>in</code>	inches
<code>cm</code>	centimètres
<code>mm</code>	millimètres
<code>pt</code>	points (1 point = 1/72 inch)
<code>pc</code>	picas (1 pica = 12 pts)
<code>%</code>	pourcentage de la valeur par défaut

Ces unités sont tirées des spécifications de [CSS](#). Cependant, la syntaxe [MathML](#) reste différente sur ce point de la syntaxe [CSS](#), puisque les nombres dans les feuilles de styles [CSS](#) ne peuvent être rationnels et peuvent être précédés du signe "+".

Lorsqu'il n'y a pas d'unité, le nombre donné sert de multiplicateur à la valeur par défaut.

Exemple :

```
<mo maxsize = "2"> ... </mo>
```

équivalent à

```
<mo maxsize = "200%"> ... </mo>
```

Cependant, pour rester compatibles avec [CSS](#), il est rare que les unités de longueur soit optionnelles. Toujours par compatibilité avec [CSS](#), 0 n'a pas besoin d'être suivie d'une unité, même si la syntaxe en spécifie une.

Pour faciliter la comptabilité avec [CSS1](#), tous les éléments [MathML](#) acceptent les attributs `class`, `style` et `id`, en plus des attributs définis pour chaque élément. Les moteurs de rendu ne supportant pas [CSS](#) se doivent d'ignorer ces attributs. La valeur de ces attributs est définie comme étant une chaîne de caractères.

Tous les éléments [MathML](#) acceptent aussi l'attribut `other`, qui permet de passer des attributs non standard sans violer la DTD. Cet attribut peut s'avérer pratique, mais il est fortement déconseillé de l'utiliser, car, dans la plupart des cas, il est possible de trouver dans [MathML](#) d'autres façons de transmettre ces informations.

Les éléments MathML

Tous les éléments [MathML](#) appartiennent à l'une des catégories suivantes :

- les éléments de présentation,
- les éléments de contenu,
- les éléments d'interface.

Les éléments de présentation

Cette classe, composée d'environ 30 éléments acceptant un peu plus de 50 attributs, regroupe l'ensemble des éléments censés représenter des notations typographiques mathématiques. De manière générale, chaque élément de présentation correspond à plusieurs types de structure de notation, comme des intégrales, des exposants, des indices, des matrices, etc. Toutes les formules mathématiques seront mises en forme grâce à ces balises emboîtées les unes dans les autres, avec, aux extrémités de l'arbre résultant, les éléments simples tels que des symboles, des chiffres, des caractères...

Les balises de présentation sont de deux types :

Les token elements : ce sont les symboles, les noms, les nombres, etc. En général, ils n'ont que des caractères et des éléments `mchar` comme fils. Tous les identificateurs, les nombres, les opérateurs, doivent être présentés à l'aide de ces éléments. Ils permettent aussi de représenter du texte ou des espaces, qui ont davantage un intérêt esthétique que réellement significatif, et pour représenter des chaînes de caractères afin de les rendre compatibles avec des systèmes algébriques.

Le Layout schemata : permet de construire des expressions. Par conséquent, on comprendra aisément que l'on nomme ces éléments des "constructeurs d'expression". Ces expressions spécifient dans quel ordre les sous-expressions sont construites et intégrées (d'où l'importance du nombre et de la disposition de ses *enfants*). On appelle aussi un *enfant* de ces constructeurs un *argument*.

Les constructeurs peuvent eux-mêmes être regroupés en plusieurs classes :

- les éléments généraux : `mrow`, `mstyle`...
- les scripts : `msup`, `munder`...
- les tableaux : `htable`, `mtr`...
- `maction`.

Un exemple simple permet de comprendre la syntaxe de ces éléments. Prenons les racines d'un polynôme de second degré. On obtient :

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

, ce qui en [MathML](#) s'écrit :

```
<mrow>
  <mi>x</mi>
  <mo>=</mo>
  <mfrac>
    <mrow>
      <mo>-</mo>
      <mi>b</mi>
      <mo>±</mo>
      <msqrt>
        <mi>b</mi>
        <mo>²</mo>
        <mo>-</mo>
        <mi>4</mi>
        <mi>a</mi>
        <mi>c</mi>
      </msqrt>
    </mrow>
    <mi>2</mi>
    <mi>a</mi>
  </mfrac>
</mrow>
```

```

        <mo>-</mo>
        <mi>b</mi>
    </mrow>
    <mo><mchar name="PlusMinus" /></mo>
    <msqrt>
        <mrow>
            <msup>
                <mi>b</mi>
                <mn>2</mn>
            </msup>
            <mo>-</mo>
            <mrow>
                <mn>4</mn>
                <mo><mchar name="InvisibleTims" /></mo>
                <mi>a</mi>
                <mo><mchar name="InvisibleTimes" /></mo>
                <mi>c</mi>
            </mrow>
        </mrow>
    </msqrt>
    <mrow>
        <mn>2</mn>
        <mo><mchar name="InvisibleTimes" /></mo>
        <mi>a</mi>
    </mrow>
</mfrac>
</mrow>

```

L'élément mrow

Cet élément est utilisé pour regrouper un nombre quelconque de sous-expressions, généralement conçues d'un ou plusieurs opérateurs (de type `mo`) agissant eux-mêmes sur différentes expressions. La plupart des éléments agissent par défaut comme si leurs arguments étaient contenus dans un élément `mrow`. Cet élément accepte seulement les attributs communs à l'ensemble des balises [MathML](#), à savoir `id`, `xref`, `class` et `style`.

Les éléments `mrow` sont rendus, comme leur nom l'indique, comme étant une rangée horizontale dans laquelle sont insérées des expressions (de gauche à droite) et par les synthétiseurs vocaux, comme une séquence d'expression. Mais cet élément n'est pas responsable du rendu des espaces entre les opérateurs et les sous-expressions. En effet, les opérateurs `mo` se chargent de fournir cette information. De même, cet élément n'indique pas directement où le système de formatage peut "couper" les expressions et revenir à la ligne (par exemple, lorsque l'expression est trop longue pour être visualisée sur une seule ligne), mais la présence des éléments `mrow` donne des points de repères sémantiques à l'interpréteur pour qu'il puisse déterminer ou séparer l'expression.

Si les balises `mrow` ne contiennent qu'un seul argument, tout se passe comme si l'élément était seul, sans être encapsulé dans des balises `mrow`. Cette équivalence est présente pour simplifier le travail des développeurs d'outils [MathML](#). De même, si la balise `mrow` redéfinit des attributs (l'attribut `style` par exemple), il n'est pas demandé, dans la spécification [MathML](#), d'exercer des règles de rendu particulières, mais les nouvelles directives d'affichage induites devront se répercuter sur l'argument.

Il n'existe aucune règle grammaticale pour l'insertion et l'utilisation des balises `mrow`. En fait, tout est laissé à l'appréciation de l'auteur. Cependant, il existe quelques recommandations pour l'utilisation de cet élément. L'intérêt de ces règles est triple :

- améliorer l'affichage en associant des espacements aux `mrow` lors du rendu ;
- favoriser les sauts de lignes lorsque cela est nécessaire au sein d'une même équation ;
- améliorer la compréhension sémantique lorsque l'on transmet le document [MathML](#) à un interpréteur algébrique ou à des synthétiseurs vocaux.

Généralement, on regroupera un opérateur avec l'ensemble de ses arguments, de manière à rendre les choses plus lisibles et donc plus compréhensibles. Mais voici les règles précises que l'on doit suivre.

Deux opérateurs adjacents, séparés ou non par des expressions (i.e. tout ce qui n'est pas un opérateur), appartiennent au même élément `mrow` si et seulement si les deux opérateurs ont la même priorité et si ces opérateurs appartiennent au dictionnaire des opérateurs fourni dans la recommandation [MathML](#) (appendice F). Dans tous les autres cas, les opérateurs seront dans des `mrow` distincts.

On entend par "opérateur" l'ensemble des opérateurs algébriques, mais aussi des parenthèses et des séparateurs.

Exemple : En respectant cette règle, l'exemple $2x + y - z$ s'écrit :

```
<mrow>
  <mrow>
    <mn>2</mn>
    <mo>&InvisibleTimes</mo>
    <mi>x</mi>
  </mrow>
  <mo>+</mo>
  <mi>y</mi>
  <mo>-</mo>
  <mi>z</mi>
</mrow>
```

de même, (x, y) s'écrit :

```
<mrow>
  <mo>( </mo>
  <mrow>
    <mn>x</mn>
    <mo>,</mo>
    <mi>y</mi>
  </mrow>
  <mo>)</mo>
</mrow>
```

L'élément `maction`

L'élément `maction` permet d'associer une action particulière à une expression. Cela permet d'ajouter une certaine interactivité au document. L'élément `maction` peut avoir un nombre quelconque d'arguments et il possède les deux attributs suivants :

- **actiontype** : cet attribut est indispensable. Son absence génère une erreur lors de l'analyse du document. Il n'y a pas de valeur par défaut, nous allons lister ci-après les valeurs possibles ;
- **selection** : de type entier, cet attribut définit le nombre de sous-expressions concernées par l'action liée. Par défaut, cette valeur est 1.

La valeur de l'attribut `actiontype` n'est pas formalisée, elle est fortement dépendante du type d'application utilisée. Pour indication, des valeurs possibles sont :

- **toggle** : permet d'afficher à tour de rôle les différentes sous-expressions, en changeant l'affichage à chaque clic de l'utilisateur ;
- **highlight** : l'expression est surlignée lorsqu'on la sélectionne avec la souris ;
- **menu** : fournit un pop-up menu des sous-expressions.

Les balises de contenu

Cette classe regroupe une centaine d'éléments acceptant à peu près une douzaine d'attributs. Les éléments de contenu sont soumis aux mêmes règles d'ordre que les éléments de présentation.

Ces éléments, qui correspondent à une large gamme d'opérateurs, de relations et de fonctions, véhiculent essentiellement de la sémantique, des entités mathématiques au sens propre, censées être dégagés de toute notion de notation. Le but de ces éléments est de structurer une information afin de la retransmettre à des interpréteurs algébriques. Ainsi, l'ensemble des entités mathématiques étant conséquent, il existe un grand nombre de balises de contenu. De plus, il existe peu d'attributs contre un grand nombre d'éléments, le W3C Math Working Group [sur internet : www.w3.org/Math/] préférant lever le plus d'ambiguïté possible en introduisant de nouveaux éléments plutôt que de nouveaux attributs.

Un bon nombre de fonctions et d'opérations nécessitent des "quantificateurs" pour être bien définies. Par exemple, une intégrale doit spécifier les bornes d'intégration, ainsi que la variable sur laquelle nous intégrons. Il y a donc un certain nombre de "**qualifier schemata**" comme `bvar` ou `lowlimit`, utilisés avec des opérateurs comme `diff` (différentiel) ou `int` (intégrale).

Présentons à présent deux des éléments les plus importants des balises de contenu : les balises `apply` et `declare`.

L'élément `declare`

L'élément `declare` est particulièrement important pour les éléments de contenu qui peuvent être évalués par des systèmes algébriques. Cet élément fournit un simple mécanisme d'assignation, où une variable peut être déclarée comme étant d'un certain type, avec une certaine valeur.

L'élément `apply`

L'élément `apply` est utilisée pour appliquer des opérations à des expressions, ainsi que pour créer de nouveaux objets mathématiques à partir d'objets existants. Il permet par exemple d'appliquer une fonction à un ensemble d'arguments. Une fois de plus, l'ordre des arguments à une importance capitale : le premier fils correspond à la fonction à appliquer, les fils restants correspondant aux arguments de la fonction. En ce qui concerne les opérations algébriques, il faut respecter, comme en LISP, la notation polonaise inversée ; l'opération $a - b$ s'écrit :

```
<apply>
  <minus/>
  <ci>a</ci>
  <ci>b</ci>
</apply>
```

Reprenons l'exemple

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

. Selon sa sémantique, il s'écrira :

```
<apply>
  <eq/>
  <ci>x</ci>
  <apply>
    <divide/>
    <apply>
      <fn><mo><mchar name='PlusMinus' /></mo></fn>
      <apply>
        <minus/>
        <ci>b</ci>
      </apply>
    </apply>
  </apply>
```

```

<root/>
<apply>
  <minus/>
  <apply>
    <power/>
    <ci>b</ci>
    <cn>2</cn>
  </apply>
  <apply>
    <times/>
    <cn>4</cn>
    <ci>a</ci>
    <ci>c</ci>
  </apply>
</apply>
<cn>2</cn>
</apply>
</apply>
<apply>
  <times/>
  <cn>2</cn>
  <ci>a</ci>
</apply>
</apply>
</apply>

```

Ce code est beaucoup plus long que celui utilisé pour la notation et l'élément `apply` y joue un rôle primordial.

Notons ici que les éléments de contenu et de présentation peuvent être mélangés. La seule règle à respecter dans ce cas est que le contenu "panaché" n'est autorisé à apparaître que s'il veut vraiment dire quelque chose. Par exemple, pour l'équation

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

, le caractère "±" n'existe pas en balise de contenu. On utilise donc `fn` pour déclarer que cet opérateur agit en tant qu'élément de contenu.

Une autre option est l'utilisation de la balise `semantics`. Cet élément est utilisé pour lier des expressions `MathML` avec d'autres types de notations. Une utilisation courante de la balise `semantics` est de joindre des éléments de contenu à des éléments de présentation, en tant qu'annotation sémantique. Ainsi, l'auteur peut spécifier qu'une notation non standard est utilisée lors du rendu d'une expression utilisant seulement des éléments de contenu. Prenons l'exemple de l'intégrale de 0 à t de 1/x, codée en balises de contenu. Le rendu par défaut serait sans doute

$$\int_0^t (1/x) dx$$

. Nous allons donc fournir une expression de présentation qui servira d'annotation sémantique afin d'obtenir l'affichage

$$\int_0^t \frac{dx}{x}$$

Il faut cependant savoir que l'utilisation présentée ci-dessus des annotations ne fait pas partie des spécifications [MathML](#). L'utilisation qui en sera faite sera donc liée au moteur de rendu choisie.

Note. Rappel : cette partie consacrée à la syntaxe et à la grammaire de [MathML](#) est issue du travail rédigé par des étudiants de 5^e année de Génie mathématique, à l'INSA de Rouen "Conversion TEX et rendu MathML" [sur internet : [./images/rapportMathML.pdf](#)]. Merci à Ulric Genièvre, Yannick Litaize, Wojciech Machocki, Ludovic Maurillon, Benoît Roger, Sébastien Vallée d'avoir décidé de façon volontaire et active de participer au développement de l'information sur le Glossaire XML.

Informations connexes

Mais finalement, c'est quoi la notation typographique mathématique ? Pour mieux comprendre la problématique, voici un extrait d'une étude réalisée par **Pierre Attar**.

La notation typographique mathématique (extrait d'un article publié dans "Documents Numériques", Vol. 1, N° 2, Editions Hermes)

Le champ de cet article s'inscrit dans la nécessité d'adapter les primitives graphiques du discours écrit pour lui permettre de s'intéresser à l'écrit scientifique : il s'agit de comprendre la nature, l'utilité et les évolutions de la notation typographique mathématique, comme outil supportant la spécificité de la communication scientifique.

En effet, s'il n'utilisait que les possibilités de l'écrit, le discours mathématique serait bavard : comprendre une phrase–texte décrivant un objet mathématique est beaucoup plus difficile que d'en comprendre une vision plus synthétique représentée par son "dessin". Par exemple, le symbolisme

$$\sqrt[3]{4}$$

est plus rapidement appréhendé – faisant partie d'une culture mathématique partagée – que le texte « *la racine cubique de la valeur entière quatre* ».

Ainsi, pour atteindre ses objectifs de communication, la communauté scientifique a été obligée d'enrichir l'expression écrite traditionnelle d'un ensemble de fonctionnalités. De nouveaux symboles et arrangements graphiques apparaissent, de nouvelles polices de caractères sont utilisées ; toutes choses qui donnent une réalité à part entière à la typographie mathématique.

Cette typographie est une extension de la typographie traditionnelle. Elle ne vient pas s'y substituer, elle l'enrichit, pour créer un nouveau mode permettant d'explicitier des concepts mathématiques : cette typographie est complètement intégrée au mode textuel. Cette intégration – interpénétration – est de nature fondamentalement différente de celle qui est utilisée pour faire coopérer les modes graphique ou image au texte, où la coopération tient plutôt de la juxtaposition.

Un mode de communication à part entière

Avant de s'intéresser plus précisément à la notation typographique mathématique, il est nécessaire d'en comprendre sa spécificité et son utilisation. Cela permettra d'éviter bon nombre des faux débats qui se traduisent par des utilisations abusives de la notion "d'équations mathématiques". Certes, les mots n'ont que le sens qu'on leur accorde, mais

derrière ces mots, bon nombre de personnes pense à tout autre chose que ce qui fait la spécificité de la notation typographique mathématique : une extension du discours écrit pour permettre l'expression d'un réel discours scientifique.

Sans prétendre cerner l'activité scientifique humaine, il semble qu'elle puisse appartenir à deux registres différents qui sont, d'une part, celui de la recherche et de l'élaboration et, d'autre part, celui de la communication et de la diffusion de résultats. À chacune de ces activités correspond des outils de travail différents, qui utilisent les concepts mathématiques sous-jacents à l'activité, selon un mode différencié.

L'activité de recherche, d'élaboration de résultats, est par définition extrêmement difficile à cerner et à expliciter ; le propos ici est de se centrer seulement sur les modes de communication utilisés lors de cette activité. Les éléments de base utilisés sont, de façon non exhaustive, la parole (ou son substitut qui est le mail sur Internet), les formules mathématiques, ainsi que les graphiques issus d'outils de représentation de ces mêmes formules. Depuis ces dernières années, un autre élément de communication apparaît autour des outils de modélisation et des outils de calcul numérique. Avec cette nouvelle catégorie d'outils, les données sont souvent stockées dans des formats propriétaires ; ce sont de réelles descriptions d'un concept mathématique, souvent vu d'un point de vue comportementaliste, voire opératoire.

La communication liée à l'activité de recherche se fait surtout à destination des "pairs", qui sont souvent eux-mêmes partie prenante du travail en cours : ce qui est échangé présuppose une connaissance commune importante des travaux en cours. Du coup, il est possible d'échanger des fragments d'information qui ne soient pas en eux-mêmes "auto-suffisants", et la communication peut être sommaire et partielle. Ainsi, deux personnes peuvent échanger un modèle mathématique électronique, issu d'un logiciel de calcul numérique . Un coup de téléphone, voire un mail d'accompagnement suffira à décrire l'objet envoyé, le récepteur ayant une excellente connaissance du contexte dans lequel l'information qu'il doit recevoir a été élaborée.

En revanche, ce mode n'est pas suffisant, dès lors que l'échange à un but de diffusion, de communication de résultats. En effet, la communauté de "pairs" devient alors plus étendue et les connaissances partagées ne sont pas aussi bien définies que précédemment. Il s'agit alors d'expliquer, d'explicitier, de (dé)montrer, tout en accompagnant le lecteur vers le résultat. Pour ce faire, le moyen depuis longtemps privilégié est l'expression écrite, dans lequel la notation typographique mathématique prend toute sa valeur, dès lors que l'expression devient scientifique. Même si aujourd'hui, avec l'avènement du Web et des CD-ROM, le support de consultation électronique pourrait devenir prépondérant, les modèles de communication reposeront toujours sur la culture commune de l'écrit, et il semble aujourd'hui encore difficile de s'en séparer.

Cela ne veut pas pour autant dire que les objets électroniques issus des outils de modélisation n'ont pas de place dans les extensions possibles du discours écrit pour mieux utiliser les nouvelles fonctionnalités offertes par la consultation électronique. Cela veut simplement dire que pendant un temps encore important, le discours écrit conservera toute sa spécificité et son pouvoir de représentation, de communication et de partage d'idées. Dans ce cadre, les objets électroniques issus d'outils de modélisation mathématique ne remplaceront pas la notation typographique mathématique, ils s'intégreront par juxtaposition, comme cela est actuellement réalisé sur papier pour utiliser le pouvoir de représentation de l'image et du graphisme. C'est aussi déjà de cette façon que fonctionnent bon nombre d'éditeurs d'information électronique, pour intégrer la vidéo, le son et les objets 3D à leur produits.

Spécificités de la typographie mathématique

La typographie mathématique propose de sortir d'une certaine linéarité du mode textuel pour faire appel au symbolisme qui synthétise des concepts et par là même densifie les contenus écrits. Pour ce faire, la typographie mathématique consiste à :

Étendre le jeu de caractères par de nouveaux caractères et des symboles. Ainsi, aux caractères de la typographie traditionnelle s'ajoutent des jeux de caractères qui n'existent que pour la typographie mathématique comme les notions d'infini, d'orthogonalité, etc. Ces caractères et symboles sont en perpétuelle évolution et renouvellement, dès lors que l'on considère le travail réalisé par les chercheurs en mathématiques qui continuent à faire vivre cette science, et donc à la faire évoluer.

Formaliser l'utilisation de constructeurs graphiques. C'est une nouvelle spécificité de la typographie mathématique. Par exemple, certains constructeurs, comme la racine, l'intégrale, la division ou encore la sommation, ont des spécificités d'étirement en fonction de leur contenu.

Définir des règles d'assemblage entre les objets graphiques. Les caractères et symboles mathématiques suivent des règles d'assemblage particulières, qui rompent avec l'assemblage séquentiel de la typographie traditionnelle. Ainsi, des positionnements relatifs sont utilisés pour, par exemple, délimiter les bornes d'une intégrale.

Formaliser l'utilisation de polices de caractères. Dans la typographie mathématique, une différenciation claire est réalisée entre les enrichissements utilisés pour définir des variables, des fonctions ou des nombres.

À l'instar de la typographie utilisée pour le mode textuel, la typographie mathématique ne fait que décrire les éléments mathématiques d'un discours. Même si sa formalisation semble plus codifiée, elle a les mêmes limites que le mode textuel : elle ne prétend pas porter un sens intrinsèque, complet et représentatif d'un concept mathématique qu'elle présente. C'est l'activité intellectuelle du lecteur qui donnera un sens au graphisme représenté. Le lecteur doit, pour ce faire, avoir connaissance d'un alphabet et d'une syntaxe spécifique : c'est sa culture scientifique.

La typographie mathématique peut alors s'autoriser des ambiguïtés et des imprécisions : elle gagne en densité et concision, toute chose qui lui confère un pouvoir pédagogique lui permettant de mettre en valeur des éléments de discours.

Éléments typographiques de la notation mathématique

Le nombre de constructeurs nécessaire à la notation typographique mathématique est relativement faible : il repose sur un ensemble de fonctionnalités graphiques permettant de spatialiser du texte ou des symboles graphiques, spécifiques à l'expression scientifique.

Tout d'abord, peu de notations mathématiques ont une définition typographique étroitement liée à la symbolique mathématique qu'elles représentent : un système doit, au minimum, connaître les sémantiques graphiques associées aux fractions et aux racines. Ces deux éléments ont la particularité de requérir une modification complexe de leur symbolisation en fonction des contenus.

D'autres éléments typographiques sont ensuite généralisés : ils reposent tous sur la notion de réserves graphiques hiérarchisées les unes par rapport aux autres, avec étirement de certains symboles. Le constructeur de base – nécessairement puissant – est alors celui qui permet le positionnement relatif d'informations les unes par rapport aux autres, pour permettre de définir des exponentielles, des indices et des vecteurs, etc.

À ce système s'ajoute des notions de sur/sous-lignages ; les traits horizontaux doivent s'étendre en fonction de la taille de la boîte référencée. Ces systèmes sont de même nature que le système vertical de parenthésage, qui inclut les notions d'intégration, de sommation

et de produit. En effet, une intégration peut, par exemple, être comprise comme étant une parenthèse gauche spécifique, s'appliquant à une boîte : l'intégrante. La spatialisation de boîtes dans un espace à deux dimensions permet, grâce aux parenthèses, de mettre en place la notion de matrices.

Pour accompagner ces constructeurs, le format de codage doit proposer, outre le jeu de caractères du mode textuel, l'accès à l'alphabet grec et un jeu de symboles mathématiques conséquent. Pour présenter ce dernier, les caractères et symboles seront accompagnés de polices de caractères identifiantes du mode mathématique et de la nature des objets sous-jacents, comme les variables ou les fonctions.

Recommandations(s)

■ ■ *Le langage de balisage mathématique (MathML) version 2.0*

Recommandation, version 2.0, du 21-02-2001

Document sur <http://www.yoyodesign.org/doc/w3c/mathml2/overview.html>

■ ■ *Mathematical Markup Language*

Recommandation, version 3.0, du 20-10-2010

Document sur <http://www.w3.org/TR/MathML3/>

■ ■ *Mathematical Markup Language Specification*

Recommandation, version 1.01, du 07-07-1999

Document sur <http://www.w3.org/TR/REC-MathML/>

■ ■ *Mettre des mathématiques sur le Web avec MathML*

Note, version 2003, du 2003

Document sur <http://www.yoyodesign.org/doc/w3c/math-on-the-web-with-mathml/>

■ ■ *Putting mathematics on the Web with MathML*

Note, version 2003, du 2003

Document sur <http://www.w3.org/Math/XSL/>

TIRÈME SARL

OASIS Tables, XML Exchange Table Model Document Type Definition

Rédaction : Pierre Attar

Issu du Département de la défense américaine et repris ensuite par toute la communauté SGML, le modèle *OASIS Tables* est maintenant adapté à XML. Il est principalement utilisé chez les éditeurs et pour la documentation technique. Sa structure est peu différente de celle des tableaux HTML, qui s'en est beaucoup inspiré.

Son avenir ? Si les notions de colonnes (`col` et `colgroup`) et de têtes (`thead` et `tfoot`) sont implémentées dans les outils XML, ce modèle pourra sans problème disparaître au profit du modèle de HTML. Dans l'attente, les bases de documents structurées avec *OASIS Tables* pourront continuer à exister, la conversion vers HTML étant triviale.

Objectifs

Le tableau fait partie des constructeurs indispensables pour augmenter la capacité d'expression du discours écrit : support à la communication. Son intérêt est de mettre en relation des idées dans un but de clarté et de concision. Discursif, le tableau s'appuie sur une *spatialisation* de l'information sur un espace bidimensionnel constitué par la page (ou par toute autre sorte de "fenêtre"), dès lors que l'information est consultée de façon électronique. Les idées émises pèsent alors le poids relatif que leur confère leur position relative dans le tableau.

Dans le discours écrit, le tableau apparaît toujours juxtaposé au texte il trouve sa place entre des paragraphes. Cette juxtaposition est spécifique et ne relève pas de l'illustration (comme l'image ou le graphique) ; elle relève beaucoup plus d'une coopération avec le mode textuel. Le tableau n'est pas autosuffisant ; il nécessite très souvent le mode textuel pour expliquer les données que lui-même présente.

Typologie des structures tableaux

D'un point de vue pratique, l'utilisation d'une structuration de l'information sous forme de tableau(x) répond à différents objectifs :

ordonnement : le tableau est un mode de classement. Les fameuses structures de listes unidimensionnelles, qui hiérarchisent entre elles des séries d'informations, en sont une illustration ;

représentation graphique de données et, plus généralement, de faits : le tableau n'est alors qu'une représentation graphique d'un ensemble d'informations par elle-mêmes organisées selon un modèle sous-jacent. La valeur du tableau résidera, de fait, dans sa capacité de présentation. Des outils graphiques évolués seront nécessaires pour permettre à l'utilisateur de manipuler directement la sémantique des données sous-jacentes :

organisations complexes d'informations : ces tableaux "libres" ne sont pas structurés par leurs données, mais plutôt par le sens relatif que donne le concepteur aux différents composants. Les tableaux mettent en regard des éléments d'information ;

outil de mise en page : "libre", là encore, le tableau est devenu et est utilisé sur les supports de consultation électroniques comme l'outil privilégié de toute mise en page. Il pallie l'absence de définition de ce que peut être une page électronique.

Les tableaux peuvent donc être utilisés, d'une part, pour représenter, de façon ergonomique et lisible, des données fortement structurées par elles-mêmes et, d'autre part, pour supporter un mode discursif de l'expression écrite. En conséquence, il convient d'adapter la terminologie pour différencier ces modes d'usage relatifs aux "tableaux à structure de contenu" de ceux à "structure libre".

Le tableau a parfois une valeur graphique intrinsèque qui ne dépend pas d'un quelconque modèle de données, mais beaucoup plus des intentions de l'auteur. Cela ne signifie pas pour autant que toutes les sémantiques sous-jacentes, si elles existent, ne soient pas utiles. Cela veut simplement dire qu'il est nécessaire de prendre en compte, dans les formats de codage textuels, cette double nature des sources d'information représentées par les tableaux. Cette prise en compte pourra alors être réalisée au niveau du codage des données elles-mêmes - pour les structures "libres" - ou au niveau des spécifications de présentation des données, pour les "structures de contenu".

Avec ses feuilles de styles **FOSI**, la communauté **CALS** a très bien compris cette idée de double nature de l'information tabulaire. Le même concept est également repris dans **XSL** et l'avait été, auparavant, dans **DSSSL**. En effet, grâce à ces technologies, des constructeurs graphiques génériques de création de tableaux sont utilisables lors de la définition des feuilles de styles. Ils s'appuient sur les notions de colonnes, de lignes et de cellules. Charge alors au concepteur d'une feuille de styles particulière d'utiliser ces constructeurs pour "mettre en tableau" des structures de contenu ou pour les appliquer à des structures "libres", saisies sous forme de tableau dès l'origine.

Fonctionnalités et formats de codage des tableaux à structure "libre"

Tous les modèles utilisés par les communautés **SGML** puis **XML** ont un ensemble de caractéristiques communes. Leur objectif est toujours de proposer un codage de données permettant d'émuler les informations nécessaires à un système générique de composition de tableaux. Les fonctionnalités sont essentiellement graphiques, calquées sur la façon dont sont actuellement utilisés les tableaux sur un support exclusivement papier. Parmi celles-ci, on distingue :

- mise en colonne, ligne et cellule,
- graphisme d'encadrement du tableau,
- filets autour d'une cellule ou d'un ensemble de cellules,
- prise en compte des recouvrements horizontaux et verticaux,
- prise en compte d'en-tête et de pied de tableau,
- prise en compte des alignements horizontaux et verticaux à l'intérieur d'une cellule,
- prise en compte des couleurs de fonds de cellules.

D'un point de vue plus structurel, un tableau comporte un titre et un ensemble de sous-tableaux, pour définir un tableau de tête, un corps et, parfois, un tableau de fin.

Un sous-tableau comporte :

un ensemble de définitions de colonnes : celles-ci n'apparaissent pas en tant que telles dans le tableau présenté. Elles permettent de donner au système de formatage des informations sur le nombre et la taille de chaque colonne. À titre indicatif, seul le modèle proposé par la société GRIF, dans son éditeur/composeur [SGML](#) prenait en compte, de façon complète, la notion de colonne. Dans les autres modèles, la colonne ne sert que de référence pour définir la position d'une cellule ;

des lignes contenant des cellules : pour une cellule, il est possible de définir des recouvrements verticaux ou horizontaux ("spanning" en anglais), des filets et des alignements horizontaux ou verticaux.

Positionnements de cellules dans un tableau "libre"

Cellule simple	Cellule simple
cellule s'étendant sur trois colonnes en largeur	
cellule s'étendant sur deux lignes et deux colonnes, en hauteur et largeur	
Cellule s'étendant sur deux lignes en hauteur	

Le format de tableaux OASIS Open

Le modèle du Consortium est une simplification et une amélioration du modèle CALS. Son propos est de rendre ce dernier modèle plus adéquat à l'échange d'informations documentaires saisies sous forme tabulaires, tout en garantissant la représentation des données saisies avec différents outils en environnement hétérogène. Proposé début 1996 pour [SGML](#), il existe aujourd'hui adapté à [XML](#). Son utilisation est généralisée dans toute l'industrie et dans beaucoup de secteurs de l'édition.

Principes

Le modèle de tableaux proposé est actuellement certainement le modèle le plus utilisé par la communauté [SGML](#) et, de façon plus générale, dans les applications traitant de documents structurés [XML](#) (bon nombre d'applications [SGML](#) ont maintenant migré sur [XML](#)). Il est reconnu et repris dans toute la documentation technique du Département de la défense américaine, mais aussi par beaucoup de [DTD](#) industrielles sectorielles, comme celles de l'aéronautique civile (DTD ATA – Air Transport Association), de l'aéronautique militaire européenne (AECMA), des constructeurs de matériels informatiques (DocBook), de l'industrie automobile (J2008), des constructeurs de matériaux semi-conducteurs (Pinnacles) ou encore des télécommunications (Telecom/Technical Interchange Markup – TIM). Les éditeurs et, notamment, les éditeurs de logiciels, lorsqu'ils intègrent un modèle de tableau, utilisent très souvent ce modèle.

Ces tableaux "libres" sont bidimensionnels et contiennent des lignes et des colonnes. Des règles de formatage de base sont proposées à tous les niveaux : colonnes, lignes et cellules, ensemble du tableau. Ces règles s'intéressent aussi bien aux contenus (alignements) qu'aux filets d'encadrement.

D'un point de vue structurel, la structure proposée est hiérarchisée. Une table contient un titre et le contenu du tableau lui-même :

- spécifications de colonne* (`colspec`) : le modèle présuppose un mécanisme d'héritage de styles et, plus généralement, de présentation. Ainsi, les spécifications de colonne s'appliqueront par défaut à l'ensemble des trois sous-tableaux, sauf spécification contradictoire à un niveau inférieur ;

- tête de tableau* (`tthead`) : la tête du tableau est identifiée de façon séparée du corps du tableau. Cette identification permet, sur papier, une présentation différenciée des contenus. Sur un support électronique, elle peut aussi être traitée de façon différenciée pour, par exemple, rester présente dans une sous-fenêtre de consultation du tableau ;

- corps de tableau* (`tbody`) : le corps du tableau contient des données. Celles-ci sont organisées sous forme de lignes contenant des cellules. Le corps est composé de lignes (`row`) qui elles-mêmes contiennent des cellules (`entry`).

D'un point de vue graphique, des fonctionnalités sont proposées, au niveau des cellules, pour mettre en place :


- des recouvrements* (utilisation de `namest` et `nameend`, en conjonction avec les définitions de colonnes) : ces chevauchements peuvent être horizontaux ou verticaux ;

- des filets* (`rowsep` et `colsep`) : ce sont des filets encadrant l'ensemble du tableau ou une cellule seulement ;

- des alignements* : à l'intérieur d'une cellule, des alignements verticaux ou horizontaux peuvent être mis en œuvre.

À de rares exceptions près, ces fonctionnalités sont largement suffisantes pour la plupart des documentations techniques et, plus généralement, pour les tableaux à structure libre. C'est ce qui participe à l'acceptation généralisée de ce modèle. En revanche, les utilisateurs d'**HTML** reprocheront à ce modèle la pauvreté des types de filets et des couleurs de fonds... ceci est certainement lié à cet héritage qui nous vient du monde de la documentation technique à qui l'on doit la création de ce modèle. Quoi qu'il en soit, ce modèle ne se définit pas comme étant un modèle de création de structures de pages : seulement de "tableaux libres".

Recommandations(s)

 *XML Exchange Table Model Document Type Definition - OASIS Technical Memorandum*

Recommandation, version TR 9901:1999, du 29-09-1999
Document sur <http://www.oasis-open.org/html/tm9901.htm>

 *Exchange Table Model Document Type Definition - OASIS Technical Resolution*

Recommandation, version TR 9503:1995, du 08-05-1996
Document sur <http://www.oasis-open.org/html/a503.htm>

 *CALS Table Model Document Type Definition - OASIS Technical Memorandum*

Recommandation, version TM 9502:1995, du 19-10-1995
Document sur <http://www.oasis-open.org/html/a502.htm>

SVG, Scalable Vector Graphics

« SVG is a language for describing two-dimensional graphics in XML. SVG allows for three types of graphic objects: vector graphic shapes (e.g., paths consisting of straight lines and curves), images and text. Graphical objects can be grouped, styled, transformed and composited into previously rendered objects. Text can be in any XML namespace suitable to the application, which enhances searchability and accessibility of the SVG graphics. The feature set includes nested transformations, clipping paths, alpha masks, filter effects, template objects and extensibility.

SVG drawings can be dynamic and interactive. The Document Object Model (DOM) for SVG, which includes the full XML DOM, allows for straightforward and efficient vector graphics animation via scripting. A rich set of event handlers such as onmouseover and onclick can be assigned to any SVG graphical object. Because of its compatibility and leveraging of other Web standards, features like scripting can be done on SVG elements and other XML elements from different namespaces simultaneously within the same Web page. »

Source. Scalable Vector Graphics Specification, version 1.1

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

■ **La spécification sur les Graphiques Vectoriels Adaptables (SVG)**

Recommandation, version 1.0, du 04-09-2001

Document sur <http://www.yoyodesign.org/doc/w3c/svg1/index.html>

🇺🇸 **Scalable Vector Graphics Specification**

Recommandation, version 1.1, du 14-01-2003

Document sur <http://www.w3.org/TR/SVG11/>

🇺🇸 **Scalable Vector Graphics Requirements**

Projet en cours, version 19981029, du 29-10-1998

Document sur <http://www.w3.org/TR/WD-SVGReq>

🇺🇸 **Mobile SVG Profiles: SVG Tiny and SVG Basic**

Recommandation, version 20030114, du 14-01-2003

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

🇺🇸 **Scalable Vector Graphics (SVG) Tiny 1.2 Specification**

Recommandation, version 1.2, du 22-12-2008

Document sur <http://www.w3.org/TR/SVGMobile12/>

🇺🇸 **SVG Tiny Version 1.2 Requirements**

Projet en cours, version 1.2, du 10-08-2006
Document sur <http://www.w3.org/TR/SVGTiny12Reqs/>

TIRÈME SARL

VoiceXML, Voice Extensible Markup Language

Recommandation(s) liée(s) : [WML](#)




« *Voice Extensible Markup Language is designed for creating audio dialogs that feature synthesized speech, digitized audio, recognition of spoken and DTMF key input, recording of spoken input, telephony, and mixed initiative conversations. Its major goal is to bring the advantages of web-based development and content delivery to interactive voice response applications.* »

Source. *Voice Extensible Markup Language (VoiceXML) , version 2.1*

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

- *Le langage de balisage extensible vocal (VoiceXML)*
Recommandation, version 2.0, du 16-03-2004
Document sur <http://www.yoyodesign.org/doc/w3c/voicexml20/>
-  *Voice Extensible Markup Language (VoiceXML)*
Recommandation, version 2.1, du 19-06-2006
Document sur <http://www.w3.org/TR/voicexml21/>
-  *Voice Browser Call Control: CCXML*
Recommandation Candidate, version 1.0, du 01-04-2010
Document sur <http://www.w3.org/TR/ccxml/>
-  *Introduction and Overview of W3C Speech Interface Framework*
Projet en cours, version 20001204, du 04-12-2000
Document sur <http://www.w3.org/TR/voice-intro/>

TIRÈME SARL

BiblioML, BiblioML

« Une application [XML](#) pour des références bibliographiques, basée sur le Format Bibliographique **Unimarc** [sur internet : www.ukoln.ac.uk/metadata/desire/overview/rev_17.htm], et pour des notices d'autorités, basée sur Unimarc / Autorités. Ce site donne des informations sur le projet [BiblioML](#) lancé en 1999 par le Ministère de la culture et de la communication - Mission de la recherche et de la technologie et réalisé par AJLSM sous la direction de Martin Sévigny.

BiblioML et AuthoritiesML sont des formats basés sur XML pour l'échange d'enregistrements bibliographiques et d'autorités entre applications. Ils ont la même granularité que le format Unimarc d'origine, c'est-à-dire les mêmes éléments de base. Mais le marquage est utilisé pour indiquer de manière lisible, non seulement la structure de l'enregistrement bibliographique, mais aussi la sémantique précise de chaque élément. »

BiblioML [sur internet : 90plan.ovh.net/~adnx/biblioml/doku.php?id=fr:introduction]

Remarque : la différence de [BiblioML](#) par rapport à [RDF](#) ou [Dublin Core](#) ? Les formats [BiblioML](#) forment un tout global et cohérent par rapport à Unimarc. Cette représentation n'est alors pas, *a priori*, définie pour être extensible : elle doit être intégrée en tant que telle dans une application.

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

■ ■ BiblioML

Recommandation, version 0.51, du 03-06-2005

Document sur [http://90plan.ovh.net/~adnx/biblioml/doku.php/doku.php?](http://90plan.ovh.net/~adnx/biblioml/doku.php/doku.php?id=fr:biblioml051)

[id=fr:biblioml051](http://90plan.ovh.net/~adnx/biblioml/doku.php/doku.php?id=fr:biblioml051)



Dublin Core, Dublin Core

Rédaction : Pierre Attar

La norme de Métadonnées du [Dublin Core](#) propose un ensemble d'éléments, simples mais efficaces, pour décrire une grande variété de ressources en réseau. Elle comprend 15 éléments dont la sémantique a été établie par un consensus international de professionnels provenant de diverses disciplines telles que la bibliothéconomie, l'informatique, le balisage de textes, la communauté muséologique et d'autres domaines connexes.

Ces champs peuvent être utilisés pour définir des Métadonnées. Le format de définition de ces Métadonnées est alors dépendant de l'environnement applicatif d'utilisation : [HTML](#), [RDF](#) ou encore [Schema](#).

Objectifs

L'objectif du [Dublin Core](#), au travers de la définition d'un ensemble de champs de Métadonnées standard, est de faciliter l'utilisation de ces informations dans des environnements applicatifs, par nature, hétérogènes comme l'est, par exemple, le Web aujourd'hui.

L'avantage de définir des champs standard ? Les moteurs de recherche du Web seront, par exemple, capables de comprendre les différentes sémantiques et de les utiliser de façon différenciées. Pour mieux différencier un mot-clé auteur d'un mot-clé purement descriptif : c'est la différence entre un article écrit par "Pierre Dupont" et un article qui parle, entre autres, de "Pierre Dupont".







Principes

La recommandation définit 15 champs standard de Métadonnées qui sont les suivants :

Contenu	Propriété intellectuelle	Instanciation *
Couverture	Collaborateur	Date
Description	Créateur	Format
Type	Éditeur	Identifiant
Relation	Droits	Langue
Source		
Sujet		
Titre		

* En **Dublin Core**, l'instanciation représente une occurrence spécifique d'une source d'information. Par exemple, le présent fichier est l'instanciation en français d'un document original anglais. Il pourrait également avoir une instanciation en **HTML**, en **XHTML**, en **XML** et même sur papier, ce qui modifierait d'autant la notice **Dublin Core**.

Recommandations(s)

-  *Information and documentation. The Dublin Core metadata element set*
Standard ISO, version 15836:2003(E), du 26-02-2003
Document sur <http://www.niso.org/international/SC4/n515.pdf>
-  *Information et documentation. L'ensemble des éléments de Métadonnées Dublin Core*
Standard ISO, version ISO 15836:2003, du 2003
Document sur <http://www.iso.ch/iso/fr/CombinedQueryResult.CombinedQueryResult?queryString=dublin+core>
-  *Dublin Core Metadata Element Set: Reference Description*
Recommandation, version 1.1, du 20-12-2004
Document sur <http://dublincore.org/documents/dces/>
-  *Guidelines for implementing Dublin Core in XML*
Proposition de recommandation, version 20020909, du 09-09-2002
Document sur <http://dublincore.org/documents/2002/09/09/dc-xml-guidelines/>
-  *Dublin Core Qualifiers*
Recommandation, version 2000/07/11, du 11-07-2000
Document sur <http://dublincore.org/documents/2000/07/11/dcmes-qualifiers/>
-  *An XML Encoding of Simple Dublincore Metadata*
Proposition de recommandation, version 2001/04/11, du 11-04-2001
Document sur <http://dublincore.org/documents/2001/04/11/dcmes-xml/>

TIRÈME SARL

EAD, Encoded Archival Description for Libraries

« EAD is a set of rules for designating the intellectual and physical parts of archival finding aids so that the information contained therein may be searched, retrieved, displayed, and exchanged in a predictable platform-independent manner. The EAD rules are written in the form of a Standard Generalized Markup Language (SGML) Document Type Definition (DTD), which uses coded representations of elements for efficient machine-processing by SGML authoring and viewing software. Because DTDs are intended to be read chiefly by computers, they are often accompanied by documentation designed for human comprehension. This tag library or master list of elements is one such piece of documentation. It serves as a reference tool for archivists who need to decide which EAD elements to use when designating the content of their organizations' finding aids. Through the use of natural-language definitions and examples, it assists archivists in achieving effective and consistent markup by rendering EAD's SGML codes more understandable. Despite translating SGML terminology and concepts into a more accessible and familiar language, the tag library is still a technical document that presupposes readers will possess a minimal understanding of SGML and finding aids. Novice finding aid encoders will need to supplement their use of the tag library by consulting EAD application guidelines, attending introductory EAD workshops and institutional training classes, and referring to other information sources. »

Source. EAD Tag Library, version 2002

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *EAD Tag Library*

Recommandation, version 2002, du 08-2002

Document sur <http://www.loc.gov/ead/tglib/index.html>

 *EAD Application Guidelines for Version 1.0*

Recommandation, version 1.0, du 1999

Document sur <http://www.loc.gov/ead/ag/aghome.html>

The logo for Tirème SARL, featuring the company name in white, uppercase letters on a dark blue rectangular background.

HyTime, Hypermedia/Time-based Structuring Language

« *The Hypermedia/Time-based Structuring Language (HyTime), defined in this International Standard, provides facilities for representing static and dynamic information that is processed and interchanged by hypertext and multimedia applications. HyTime is an application of ISO 8879, the Standard Generalized Markup Language (SGML).*

HyTime supports the classic bibliographic model of information referencing, whereby it is possible to represent links to anything, anywhere, at any time, in a variety of ways. The extension of this model to the computerized information age, known as "integrated open hypermedia" (IOH), is the field of application of HyTime.

HyTime provides standardized mechanisms for specifying interconnections (hyperlinks) within and between documents and other information objects, and for scheduling multimedia information in time and space.

Without HyTime, such information is typically embedded in the processing instructions of hypermedia "scripts" that govern the rendition of such documents, and is therefore not usable for other forms of processing. When HyTime is used, those properties of the information that are independent of specific processing are available for processing by applications and platforms other than the one on which the information was created.

It is for the application designer and user to decide which properties can be isolated from the scripts in this way. In an ideal world, the sole consideration would be whether the properties are intrinsic to the information, regardless of how it is processed. For example, the title of this clause is intrinsic information; the font that it appears in normally is not.

In the real world, representation strategies will vary from one situation to another and will depend on such other considerations as the expected uses of the information, the flexibility of the scripting language, and performance considerations. For this reason, HyTime is highly modularized so that application designers need use only the facilities for the properties they care to describe in a standardized way.


HyTime's rules for the standardized expression of hypermedia structuring are expressed as an "enabling architecture", consisting of a number of "architectural forms" and their associated semantics. The HyTime standard's formal definition as an architecture conforms to the Architectural Form Definition Requirements in A SGML Extended Facilities of this International Standard. »

Source. *Information processing - Hypermedia/Time-based Structuring Language (HyTime 2d Edition), version ISO/IEC 10744:1997*

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *Information processing - Hypermedia/Time-based Structuring Language (HyTime 2d Edition)*

Standard ISO, version ISO/IEC 10744:1997, du 09-05-1997

Document sur <http://www.y12.doe.gov/sgml/wg8/document/n1920/>

TIRÈME SARL

MADS, Metadata Authority Description Schema

Recommandation(s) liée(s) : [MODS](#)

« The Library of Congress' Network Development and MARC Standards Office has developed the Metadata Authority Description Schema ([MADS](#)), an XML schema for an authority element set that may be used to provide metadata about agents (people, organizations), events, and terms (topics, geographics, genres, etc.). [MADS](#) was created to serve as a companion to the Metadata Object Description Schema ([MODS](#)). As such, [MADS](#) has a relationship to the MARC 21 Authority format, as [MODS](#) has to MARC 21 Bibliographic, but it simplifies data. Influenced by the FRBR development, [MADS](#) relates to [MODS](#) in new and useful ways. [MADS](#) is expressed using the XML schema language of the World Wide Web Consortium. The standard will be developed and maintained by the Network Development and MARC Standards Office of the Library of Congress with input from users.

The [MADS](#) draft schema is now available for broad review to inform its completion. Based on input from prospective users, the schema will be revised and made available for experimentation. After the review period, changes will be made to [MODS](#) to make the two schemas more consistent. »

Site de Metadata Authority Description Schema [sur internet : www.loc.gov/standards/mads/]

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 [Metadata Authority Description Schema](#)

Projet en cours, version 1.0, du 21-04-2005

Document sur <http://www.loc.gov/standards/mads/mads.xsd>

TIRÈME SARL

MODS, Metadata Object Description Schema

Recommandation(s) liée(s) : [MADS](#)

« The Library of Congress' Network Development and MARC Standards Office, with interested experts, has developed a schema for a bibliographic element set that may be used for a variety of purposes, and particularly for library applications. As an XML schema, the "Metadata Object Description Schema" (MODS) is intended to be able to carry selected data from existing MARC 21 records as well as to enable the creation of original resource description records. It includes a subset of MARC fields and uses language-based tags rather than numeric ones, in some cases regrouping elements from the MARC 21 bibliographic format. MODS is expressed using the XML schema language of the World Wide Web Consortium. The standard is maintained by the Network Development and MARC Standards Office of the Library of Congress with input from users. »

Source. *Metadata Object Description Schema, version 3.0*

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *Metadata Object Description Schema*
Projet en cours, version 3.0, du 05-12-2003
Document sur <http://www.loc.gov/standards/mods>

OAI, Open Archives Initiative

« L'Open Archive Initiative Protocol for Metadata Harvesting (OAI-PMH) ou « Protocole de collecte de Métadonnées de l'Initiative Archives Ouvertes » définit un standard pour transférer des collections de métadonnée. Il a été conçu dans le cadre de l'Initiative Archives Ouvertes pour implanter des bases interopérables de pré-publications scientifiques soumises par leurs auteurs. L'objectif de l'Initiative Archives Ouvertes est donc très spécifique aux communautés de chercheurs. Pourtant le protocole est apparu très intéressant pour les applications patrimoniales.

Fonctionnellement, le protocole permet de centraliser les métadonnée référençant diverses ressources mais laisse ces ressources à leur emplacement initial. Il est alors possible d'accéder à ces ressources en architecture distribuée, en rassemblant les *Métadonnées* et en les exploitant pour les besoins spécifiques d'un service.

En général, les ressources patrimoniales sont gérées dans de nombreux établissements différents. Pour fournir un accès commun à ces ressources (catalogues nationaux par exemple), un système permet d'en rassembler les descriptions dans une base commune. L'OAI-PMH propose des fonctionnalités pour cette configuration, en définissant un standard pour la mise à disposition et le transfert des métadonnée sur les ressources patrimoniales. »

Muriel Foulonneau

The Open Archives Initiative Protocol for Metadata Harvesting (referred to as the OAI-PMH in the remainder of this document) provides an application-independent interoperability framework based on metadata harvesting. There are two classes of participants in the OAI-PMH framework:


- . Data Providers administer systems that support the OAI-PMH as a means of exposing metadata; and
- . Service Providers use metadata harvested via the OAI-PMH as a basis for building value-added services.

Source. *The Open Archives Initiative Protocol for Metadata Harvesting, version 2.0*

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

-  *The Open Archives Initiative Protocol for Metadata Harvesting*
Recommandation, version 2.0, du 14-06-2002
Document sur <http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm>

 *Implementation Guidelines for the Open Archives Initiative Protocol for Metadata Harvesting*

Note, version 20020614, du 14-06-2002

Document sur <http://www.openarchives.org/OAI/2.0/guidelines.htm>

TIRÈME SARL

RDF, Resource Description Framework

Rédaction : Pierre Attar

Recommandation(s) liée(s) : [OWL](#) - [RDF-S](#) - [RDFa](#) - [Topic Maps](#) - [XTM](#) - [XHTML](#)

Si le Web a été conçu, au départ, pour permettre à des intelligences humaines de lire des documents – accessibles par des hyper-liens –, la démultiplication de l'information disponible nécessite aujourd'hui d'autres types d'accès, basés sur des automatismes gérés par des programmes informatiques. C'est dans ce cadre que l'on voit fleurir bon nombre de moteurs de recherche sur Internet et beaucoup de technologies, encore à l'état expérimental, comme celles des "agents".

RDF permet de rendre plus "intelligente" l'information nécessaire à ces moteurs de recherche et, plus généralement, nécessaire à tout outil informatique analysant de façon automatisée des pages Web. Pour ce faire, RDF propose d'associer à toute ressource du Web un ensemble de descriptifs qui caractérisent au mieux cette ressource : on parle alors de métadonnées (en anglais, *metadata*). Présupposant un monde ouvert ou tout le monde parle de tout, RDF permet d'agréger des informations issues de différents éditeurs, sur un même sujet.

Du métalangage de définition de métadonnées, il est nécessaire de dériver des langages de définition de corpus homogènes de métadonnées. C'est la fonction d'un métalangage de définition de modèles (RDF-S), souvent appelés "vocabulaires". Ces vocabulaires RDF (qui relèvent plutôt de l'ordre de la syntaxe) seront spécifiques à une profession ou à une communauté particulière.

Une application pourra alors, par simple analyse des types de données définies dans le modèle, reconnaître un certain nombre de sémantiques applicables aux métadonnées liées à ce modèle. Elle pourra aussi déduire de nouvelles informations et c'est là où l'intelligence artificielle refait son apparition, avec ses bases de faits (les métadonnées) sur lesquelles des programmes vont pouvoir "raisonner" et "inférer".

Décrire est une activité bien connue. Les bibliothécaires, les archivistes pratiquent la description depuis bien longtemps. Par ailleurs, HTML lui-même, dans ses versions récentes, proposait un certain nombre de possibilités de description dans les entêtes de document (éléments meta contenus dans élément head). RDF se propose alors de définir un cadre (*framework*) de définition de métadonnées, sans se prononcer plus avant sur la nature des métadonnées elles-mêmes.

Pour ce qui concerne XHTML, l'association à RDFa est maintenant formalisée dans RDF qui propose une méthode pour identifier des métadonnées n'importe où dans le corps d'une page Web (et pas seulement dans l'entête comme cela se faisait préalablement).

À noter, pour finir, que **RDF-S** prend toute sa signification avec l'apparition d'**OWL** qui permet de mettre en relation des vocabulaires (on parlera d'ontologies) présentant des concepts similaires.

Objectifs

L'objectif de **RDF** est de proposer un cadre formel de définition de métadonnées, sans préjuger des vocabulaires et syntaxes utilisés pour écrire ces métadonnées : en ce sens, **RDF** est un *métalangage* spécialisé dans les métadonnées. Son objectif : rendre plus pertinent le traitement automatisé des informations contenues sur le Web, par la possibilité de fournir aux outils de traitement une information plus sémantique que les seuls mots contenus dans un document. De ce métalangage, il sera possible de définir des langages de description de données : ce sera l'objectif des vocabulaires **RDF**.

Un autre objectif de **RDF** est de fédérer les vocabulaires et syntaxes de description de métadonnées existantes dans un *cadre (framework)* commun. Cela ne veut pas dire qu'il s'agit de définir **LE** modèle de métadonnées, mais plutôt de permettre à chaque modèle de s'insérer harmonieusement dans les métadonnées décrivant une ressource particulière. Dans ce cadre, les vocabulaires **RDF** permettront de mieux contrôler les métadonnées au regard de leur modèle.

Dernier objectif, et pas des moindres, proposer une solution dans ce monde Internet totalement ouvert où "tout le monde peut dire quelque chose sur n'importe quel sujet". Du coup, il faut pouvoir identifier, mais sans préjuger du fait que quelqu'un pourra identifier un même concept d'une autre façon. Il faut aussi permettre à différentes entités, distribuées sur le Web, de parler de la même chose, pourquoi pas de façon contradictoire.

Pour atteindre ses objectifs, le principe de **RDF** est de définir des *ressources* par des *propriétés*. Comme toutes les ressources sont *identifiées*, c'est l'agrégation des propriétés qui donnera une "complétude" aux métadonnées de la ressource. Ainsi, un éditeur pourrait donner un ensemble de métadonnées descriptives sur les livres qu'il commercialise. Un site de vente par correspondance pourra alors utiliser ces métadonnées pour les agréger avec différents autres sites commercialisant l'ouvrage et proposer l'ensemble à l'internaute.

Principes

Dans ses principes de base, **RDF** dissocie trois types d'objets : une "ressource" (*resource*) est définie par des valeurs (*object*) de "propriétés" (*properties*) ; l'association d'une ressource à une propriété par une valeur de propriété est une "assertion" (*statement*). Une assertion est donc un triplet : l'association des trois informations *sujet, prédicat, objet*.

Ainsi, il est possible de décrire, avec **RDF**, des assertions du type : *http://www.tireme.fr/glossaire a pour titre "Glossaire de Pierre Attar": http://www.tireme.fr/glossaire est la ressource (le sujet dont on parle) ; "a pour titre" est une propriété dont la valeur est "Glossaire de Pierre Attar"*.

La syntaxe utilisée par **RDF** étant **XML**, la déclaration précédente pourra s'écrire :

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <rdf:description about="http://www.tireme.fr/glossaire/"
    <titre>"Glossaire de Pierre Attar"</titre>
```

```
</rdf:description>
</rdf:RDF>
```

Un outil automatisé comprendra alors que l'on est en train de parler d'une ressource avec un propriété de type "titre" dont la valeur de propriété est "Glossaire de Pierre Attar". Que veut dire la propriété "titre" ? C'est à ce niveau que doivent intervenir les vocabulaires et syntaxes spécialisés (le [Dublin Core](#) propose de d'utiliser `dc:title` en lieu et place de `titre` dans l'exemple ci-dessus).

La syntaxe XML proposée peut prendre différentes formes et la seule chose finalement importante est le fait que l'ensemble des triplets forment un graphe orienté, fait de ressources (les nœuds) reliés par des propriétés (arcs orientés).

L'identification de ressource est une [URI](#), que celle-ci identifie une information relative, dans le même fragment XML ou n'importe quelle autre ressource identifiée.

Une assertion est alors la conjonction :

- *sujet* : ce dont on parle, une [URI](#),
- *prédicat* : comment on en parle, selon un vocabulaire défini dans une [URI](#),
- *objet* : la valeur de la propriété. À noter que l'objet peut prendre une valeur littérale (51kg) ou être lié à une ressource par ailleurs existante. Si l'objet est un littéral, il peut être typé et les types des [Schema](#) sont utilisables.

Ici, la force de [RDF](#) est de *ne pas* se prononcer sur le sujet et de laisser aux personnes définissant leurs métadonnées le choix du (des) vocabulaire(s) utilisé(s). Ainsi, il serait, par exemple, possible de définir plusieurs propriétés "créateur" : une qui soit compatible avec le [Dublin Core](#), spécification de métadonnées extrêmement généraliste et, une qui soit compatible avec un modèle privé, recensant tous les créateurs dans une base de données *ad hoc*. Le fragment s'écrirait alors :

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/metadata/dublin_core#"
  xmlns:mutu="http://www.mutu-xml.org/modèles/meta">
  <rdf:description about="http://www.mutu-xml.org">
    <dc:creator>projet MUTU-XML</dc:creator>
    <mutu:créateur nom-en-base="MUTU-XML" />
  </rdf:description>
</rdf:RDF>
```

Pour aller plus loin, [RDF](#) définit un certain nombre d'autres constructeurs, permettant d'assembler des métadonnées entre elles :

- les conteneurs (*container*) permettent d'assembler des valeurs propriétés entre elles. Cet assemblage peut définir un ensemble simple (*bag*), une séquence ordonnée (*sequence*) ou encore une alternative de valeurs (*alternative*) ;
- il est enfin possible de faire des déclarations sur des déclarations, on parle alors de "réification". Par exemple, pour dire des choses de type : "*Mr Dupont a dit que Mutu-XML est le créateur de http://www.mutu-xml.org*".

Recommandations(s)

■ ■ *Initiation à RDF*

Recommandation, version 20040210, du 10-02-2004

Document sur <http://www.yoyodesign.org/doc/w3c/rdf-primer/>

■ ■ *Cadre de Description des Ressources (Resource Description Framework ou RDF) : Concepts et Syntaxe Abstraite*

Recommandation, version 20040210, du 10-02-2004

Document sur <http://www.lacot.org/w3c/REC-rdf-concepts-20040210/>

■ ■ *Spécification de la syntaxe RDF/XML (révisée)*

Recommandation, version 20040210, du 10-02-2004
Document sur <http://www.yoyodesign.org/doc/w3c/rdf-syntax-grammar/>

■ ■ *Sémantique RDF*

Recommandation, version 20040210, du 10-02-2004
Document sur <http://www.yoyodesign.org/doc/w3c/rdf-mt/>

■ ■ *Langage de description de vocabulaire RDF 1.0 : RDF Schema*


Recommandation, version 20040210, du 10-02-2004
Document sur <http://www.yoyodesign.org/doc/w3c/rdf-schema/>

■ ■ *Jeux d'essais RDF*

Recommandation, version 20040210, du 10-02-2004
Document sur <http://www.yoyodesign.org/doc/w3c/rdf-testcases/>

 *RDF Primer*


Recommandation, version 20040210, du 10-02-2004
Document sur <http://www.w3.org/TR/rdf-primer/>

 *Resource Description Framework (RDF): Concepts and Abstract Syntax*

Recommandation, version 20040210, du 10-02-2004
Document sur <http://www.w3.org/TR/rdf-concepts/>

 *RDF/XML Syntax Specification (Revised)*

Recommandation, version 20040210, du 10-02-2004
Document sur <http://www.w3.org/TR/rdf-syntax-grammar/>

 *RDF Semantics*

Recommandation, version 20040210, du 10-02-2004
Document sur <http://www.w3.org/TR/rdf-mt/>

 *RDF Vocabulary Description Language 1.0: RDF Schema*

Recommandation, version 20040210, du 10-02-2004
Document sur <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>

 *RDF Test Cases*

Recommandation, version 20040210, du 10-02-2004
Document sur <http://www.w3.org/TR/rdf-testcases/>

TIRÈME SARL

RDF-S, Resource Description Framework Semantics

Rédaction : Pierre Attar

Recommandation(s) liée(s) : [RDF](#) - [RDFa](#) - [OWL](#) - [Topic Maps](#) - [XTM](#)

Si [RDF](#) est un métalangage de définition de Métadonnées, il peut être important de lui associer une définition des termes et propriétés utilisés pour ces Métadonnées. L'intérêt ? Pouvoir partager sur Internet la sémantique des données.

Par exemple, "*Droopy a une fourrure blanche*" ne présuppose en aucun cas de qui est Droopy. En revanche, il peut être intéressant de savoir que Droopy est de la classe des mammifères, qui plus est de la classe des chiens. C'est la fonction de [RDF-S](#), que de permettre d'identifier des classes et des relations (propriétés) entre les classes.

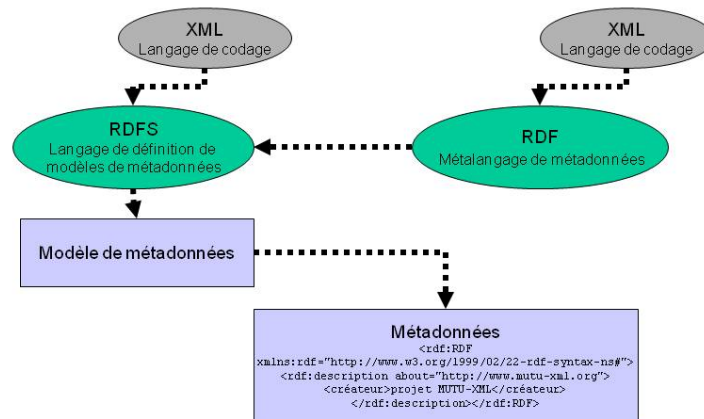
Du coup, il deviendra possible d'*inférer* en utilisant les classes et les propriétés ! La connaissance du Web par des programmes devient alors plus riche : c'est bien l'objectif du Web sémantique.

[RDF-S](#), pour se définir, utilise [RDF](#) : l'ensemble forme alors un tout cohérent. À noter, enfin, l'existence de [OWL](#) qui formalise davantage les possibilités de définitions de classes et de relations entre elles.

Objectifs

En définissant des classes et des propriétés, l'objectif de [RDF-S](#) est de définir des modèles de Métadonnées permettant de raisonner sur les données fournies, dans le but de permettre à des outils intelligents de mieux comprendre les contenus du Web. Ainsi, de par l'appartenance des individus (les ressources du Web) à des classes, il sera possible d'ingérer de nouvelles propriétés sur ce qu'ils représentent.

Utiliser [RDF-S](#) revient alors à définir un réel vocabulaire de termes et de liens entre termes. Le langage est basé sur [RDF](#).



Principes

RDF-S permet de définir

- des classes (par exemple la classe des Femmes Mariées)

Pour aller plus loin, le standard permet aussi de définir des hiérarchies de classes : "Femme Mariée" → "Femme" → "Humain" → "Animal"...

- Des propriétés (par exemple nom de jeune fille)

Si on souhaite associer une propriété (*nom de jeune fille*) à une classe (*Femme Mariée*) c'est uniquement pour pouvoir déduire de la propriété une information sur la nature de la ressource dont on parle.

Ainsi, il peut être intéressant de déduire de l'assertion "*Le nom de jeune fille de Marie est Dupont*" que Marie est une *Femme Mariée*... qui plus est c'est une *Femme* vu la relation d'héritage des classes définie plus haut.

Ainsi donc, dans **RDF-S**, tout est fait pour permettre la mise en place d'inférence sur le vocabulaire dans un objectif de déduire des informations nouvelles à partir des Métadonnées trouvées.

On le voit, les primitives de **RDF-S** sont assez simples. Du coup, les manques suivants seront comblés avec **OWL**:

- contraintes de cardinalité sur les propriétés par exemple une personne (Person) a exactement un seul père biologique ;

- inférer la transitivité : par exemple, si Pierre a pour ancêtre (ex : `hasAncestor`) Bernard, et que Bernard a pour ancêtre Charles, alors Pierre a pour ancêtre Charles ;

- une propriété donnée permet une identification unique pour les instances d'une classe particulière (autre façon de pouvoir inférer que, si une personne a deux mères, les deux personnes représentent en fait le même individu);

- équivalence de classes... particulièrement utile pour "fusionner" des Métadonnées provenant de deux vocabulaires différents ;

contraintes sur la cardinalité d'une propriété qui dépendent de la classe d'une ressource à laquelle s'applique une propriété... par exemple pouvoir dire que, pour une équipe de football, la propriété `ex:hasPlayers` a 11 valeurs, alors que, pour une équipe de basket ball, la même propriété ne devrait avoir que 5 valeurs ;
décrire de nouvelles classes en fonction de combinaisons (par exemple, unions et intersections) d'autres classes, ou de dire que deux classes sont disjointes

Recommandations(s)

RDF Semantics

Recommandation, version 20040210, du 10-02-2004

Document sur <http://www.w3.org/TR/rdf-mt/>

Sémantique RDF

Recommandation, version 20040210, du 10-02-2004

Document sur <http://www.yoyodesign.org/doc/w3c/rdf-mt/>

TIRÈME SARL

OWL, Web Ontologies Language

Rédaction : Pierre Attar

Recommandation(s) liée(s) : [RDF](#) - [RDF-S](#) - [RDFa](#) - [Topic Maps](#) - [XTM](#)

OWL s'inscrit dans les langages du Web permettant de définir des Métadonnées sur les informations disponibles sur Internet. Basé sur [RDF](#), OWL étend le pouvoir de description du langage de modélisation. Ainsi, si [RDF-S](#) (la partie sémantique de [RDF](#)) s'est attaché à définir des vocabulaires, le W3C s'est rapidement aperçu de la faiblesse du pouvoir d'expression de ce langage de modélisation qui ne permettait pas de pouvoir aisément mettre en relation des vocabulaires, des ontologies. Pour pallier cette faiblesse, OWL a été standardisé qui étend le pouvoir de [RDF-S](#) tout en lui restant compatible.

OWL permet donc de définir des ontologies qui sont l'association de définitions de **termes** d'un domaine particulier et des **relations** entre ces termes. L'objectif ? Pourvoir inférer sur les modèles pour en déduire de nouvelles propriétés des Métadonnées.

Ainsi, si *Droopy est un chien* et si par ailleurs on sait que *les chiens sont des mammifères*, on pourra en déduire que *Droopy est un mammifère*. Cette inférence est simple ; elle était déjà disponible avec [RDF-S](#). OWL ajoute alors de nouvelles possibilités de description, comme les définitions de classe par une méthode ensembliste (la classe des animaux contient "tous les mammifères" et "tous les animaux qui ne sont pas des mammifères"). Le standard ajoute aussi toute sorte de restrictions possibles en terme d'énumération de valeurs de propriétés.

Objectifs

L'objectif de OWL est donc de fournir au travers des ontologies des moyens de raisonner sur des Métadonnées et de déduire de nouveaux faits/assertions.

Fondamentalement, le principe est le suivant : on étudie les assertions proposées et en identifiant les catégories et les propriétés, on déduit de nouvelles assertions. C'est ce qui s'appelle raisonner... pour un programme informatique.

OWL fournit de nouveaux moyens de raisonner qui manquaient à [RDF-S](#).

Recommandations(s)

- [Vue d'ensemble du langage d'ontologie Web OWL](#)
Recommandation, version 10-02-2004, du 10-02-2004
Document sur <http://www.yoyodesign.org/doc/w3c/owl-features-20040210/>
- [Le langage d'ontologie Web OWL Guide](#)

Recommandation, version 10-02-2004, du 10-02-2004

Document sur <http://www.yoyodesign.org/doc/w3c/owl-guide-20040210/>

■ ■ *La référence du langage d'ontologie Web OWL*

Recommandation, version 10-02-2004, du 10-02-2004

Document sur <http://www.yoyodesign.org/doc/w3c/owl-ref-20040210/>

■ ■ *La sémantique et la syntaxe abstraite du langage d'ontologie Web OWL*

Recommandation, version 10-02-2004, du 10-02-2004

Document sur <http://www.yoyodesign.org/doc/w3c/owl-semantics-20040210/>

■ ■ *Les jeux d'essais du langage d'ontologie Web OWL*

Recommandation, version 10-02-2004, du 10-02-2004

Document sur <http://www.yoyodesign.org/doc/w3c/owl-test-20040210/>

■ ■ *Les cas et conditions d'utilisation du langage d'ontologie Web OWL*

Recommandation, version 10-02-2004, du 10-02-2004

Document sur <http://www.yoyodesign.org/doc/w3c/webont-req-20040210/>

■ *OWL Web Ontology Language Overview*

Recommandation, version 10-02-2004, du 10-02-2004

Document sur <http://www.w3.org/TR/owl-features/>

■ *OWL Web Ontology Language Guide*

Recommandation, version 10-02-2004, du 10-02-2004

Document sur <http://www.w3.org/TR/owl-guide/>

■ *OWL Web Ontology Language Reference*

Recommandation, version 10-02-2004, du 10-02-2004

Document sur <http://www.w3.org/TR/owl-ref/>

■ *OWL Web Ontology Language Abstract Syntax and Semantics*

Recommandation, version 10-02-2004, du 10-02-2004

Document sur <http://www.w3.org/TR/owl-semantics/>

■ *OWL Web Ontology Language Test Cases*

Recommandation, version 10-02-2004, du 10-02-2004

Document sur <http://www.w3.org/TR/owl-test/>

■ *OWL Web Ontology Language Use Cases and Requirements*

Recommandation, version 10-02-2004, du 10-02-2004

Document sur <http://www.w3.org/TR/webont-req/>

■ *OWL Web Ontology Language XML Presentation Syntax*

Note, version 11-06-2003, du 11-06-2003

Document sur <http://www.w3.org/TR/owl-xmlsyntax/>



RDFa, Resource Description Framework attributes

Rédaction : Pierre Attar

Recommandation(s) liée(s) : [OWL](#) - [RDF-S](#) - [RDF](#) - [RDFa](#) - [Topic Maps](#) - [XTM](#) - [XHTML](#)

RDFa est une recommandation du W3C dont l'objectif est de permettre l'annotation des contenus de n'importe quel document du Web par des Métadonnées compatibles avec le modèle RDF.

Avec RDFa, il devient possible d'inclure des Métadonnées dans toutes les parties du document XHTML, et pas seulement sur le document dans tout son ensemble.

Ainsi, sur une page de blog présentant une publication coopérative, il sera possible d'identifier la page elle-même comme appartenant à la classe "Publication". Le titre de la publication sera marqué par la propriété "dc:title" (en utilisant le [Dublin Core](#)). Enfin, chaque auteur sera marqué par la propriété "dc:author".

```
<body>
  <div xmlns:dc="http://purl.org/dc/elements/1.1/"
    about="http://www.tireme.fr/laPublicationPassionnante.html"
    typeof="Publication">
    <p>Je viens de lire une publication passionnante <span
      property="dc:title">De la culture des petits pois en
      alaska</span>
    cette publication est éditée par <span
      property="dc:publisher">Tirème Editions</span></p>
    <p>Parmi les auteurs on trouvera bien sûr <span
      property="dc:author">Pierre Attar</span> mais aussi <span
      property="dc:author">Paul Att</span>. A lire donc avec
    attention !</p>
  </div>
</body></html>
```

Toute application RDFa pourrait alors identifier dans la page de blog ci-dessus un ensemble d'informations tout à fait pertinentes sur la publication en question (dont l'adresse est identifiée dans la balise `div`).

Sans nuire à la lisibilité de la page, sans obliger de saisir deux fois la même information, et sans non plus obliger à restructurer les contenus d'une page, il devient possible de rendre les pages du web lisible par des processus automatisés qui pourront "comprendre" la signification des contenus eux-mêmes.

RDFa devient alors la pierre angulaire décisive de la mise en place du Web sémantique.

Objectifs


Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *RDFa dans XHTML — Syntaxe et traitement Une collection d'attributs et de règles de traitement pour étendre XHTML et gérer RDF*


Projet en cours, version 20081014, du 14-10-2008

Document sur <http://www.yoyodesign.org/doc/w3c/rdfa-syntax/Overview.html#>

 *RDFa Primer. Embedding Structured Data in Web Pages*

Projet en cours, version 20080317, du 17-03-2008

Document sur <http://www.w3.org/TR/xhtml1-rdfa-primer/#>

 *RDFa in XHTML: Syntax and Processing*

Recommandation, version 20081014, du 17-03-2008

Document sur <http://www.w3.org/TR/rdfa-syntax/>

Topic Maps, Topic Maps

Recommandation(s) liée(s) : [RDF](#) - [XLink](#) - [XTM](#)

This International Standard provides a standardized notation for interchangeably representing information about the structure of information resources used to define topics, and the relationships between topics. A set of one or more interrelated documents that employs the notation defined by this International Standard is called a "topic map". In general, the structural information conveyed by topic maps includes:

- . groupings of addressable information objects around topics (occurrences), and*
- . relationships between topics (associations).*

A topic map defines a multidimensional topic space - a space in which the locations are topics, and in which the distances between topics are measurable in terms of the number of intervening topics which must be visited in order to get from one topic to another, and the kinds of relationships that define the path from one topic to another, if any, through the intervening topics, if any.

In addition, information objects can have properties, as well as values for those properties, assigned to them externally. These properties are called "facet types".

Several topic maps can provide topical structure information about the same information resources. The Topic Maps architecture is designed to facilitate merging topic maps without requiring the merged topic maps to be copied or modified. Because of their extrinsic character, topic maps can be thought of as "overlays" on, or extensions to, sets of information objects

The base notation of Topic Maps is SGML; an interchangeable topic map always consists of at least one SGML document, and it may include and/or refer to other kinds information resources. A set of information resources that comprise a complete interchangeable topic map can be specified using the bounded object set (BOS) facility defined by the HyTime architecture in ISO/IEC 10744:1997. »


Source. *Topic Maps: Information Technology -- Document Description and Markup Languages, version ISO/IEC 13250:2000*

Voir aussi [XTM](#).


Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *Topic Maps: Information Technology -- Document Description and Markup Languages Standard ISO, version ISO/IEC 13250:2000, du 03-12-1999*

Document sur <http://www.y12.doe.gov/sgml/sc34/document/0129.pdf>

 *Topic maps, roadmap for further work*

Projet en cours, version ISO/IEC JTC 1/SC34 N°278, du 17-12-2001

Document sur <http://www.y12.doe.gov/sgml/sc34/document/0278.htm>

TIRÈME SARL

XTM, XML Topic Maps

Recommandation(s) liée(s) : RDF - Topic Maps - XLink

XTM est le résultat du portage de la norme ISO sur les Topic Maps en XML.

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

- *XML Topic Maps (XTM 1.0)*
Recommandation, version 1.0, du 06-08-2001
Document sur <http://xmlfr.org/topicmaps.org/xtm/1.0/>
- *XML Topic Maps*
Recommandation, version 1.0, du 06-08-2001
Document sur <http://www.topicmaps.org/xtm/1.0/>

TIRÈME SARL

SMIL, Synchronize Multimedia Integration Language

« Defines an XML-based language that allows authors to write interactive multimedia presentations. Using SMIL, an author may describe the temporal behaviour of a multimedia presentation, associate hyperlinks with media objects and describe the layout of the presentation on a screen... »

Source. SMIL Specification, version 3.0

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

■ ■ *Langage d'Intégration Multimédias Synchronisés (SMIL)*

Recommandation, version 2.0, du 07-08-2001

Document sur <http://xmlfr.org/w3c/TR/smil20/>

■ ■ *SMIL Animation*

Recommandation, version 20010904, du 04-09-2001

Document sur <http://www.yoyodesign.org/doc/w3c/smil-animation/>

🇺🇸 *SMIL Specification*

Recommandation, version 3.0, du 01-12-2008

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

🇺🇸 *SMIL Animation*

Recommandation, version 20010904, du 04-09-2001

Document sur <http://www.w3.org/TR/smil-animation/>

🇺🇸 *SMIL Document Object Model*

Projet en cours, version 20000225, du 25-02-2000

Document sur <http://www.w3.org/TR/smil-boston-dom/>

🇺🇸 *XHTML+SMIL Profile*

Note, version 20020131, du 31-01-2002

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

TIRÈME SARL

DITA, Darwin Information Typing Architecture

Rédaction : Pierre Attar

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

À la différence de [DocBook](#) – dont l'approche est extrêmement narrative dans un livre –, le principe de [DITA](#) est de construire un ensemble de modules documentaires, spécialisés (*topics*) et assemblables par configuration, afin de réaliser une publication.

Les *topics* sont typés selon une classification de base – concept, tâche et référence – extensible selon des méthodes basées sur l'héritage. Ce typage permet de réaliser des recherches pertinentes par type de *topics* et donc de présenter à l'utilisateur uniquement l'information qui lui convient pour réaliser une tâche particulière.

Plus généralement, [DITA](#) utilise le terme de spécialisation : celle-ci s'entend à deux niveaux, spécialisation des *topics* et spécialisation des modèles de contenus. Dérivée des architectures-objet, cette notion de spécialisation permet à un ensemble d'entreprises d'utiliser un cadre de référence commun tout en se permettant de mieux adapter les concepts sous-jacents à leurs besoins – tant en termes d'ingénierie documentaire qu'en termes d'organisation documentaire.

Malgré ce découpage en fragments d'information que représentent les *topics*, une publication reste l'assemblage de l'ensemble des fragments nécessaires pour fournir l'information adéquate à un utilisateur, pour un produit donné. Il est donc nécessaire de fournir cet outil d'assemblage de *topics* et [DITA](#) propose la notion de *map* : un nouvel objet [XML](#) permettant de définir comment assembler un ensemble de *topics*.

Objectifs

À la différence de [DocBook](#) où toute l'information est dans un même livre, l'objectif de [DITA](#) est de constituer des référentiels documentaires dont les contenus sont assemblables – à la demande –, dans des publications. L'idée de base est, sur un sujet donné, de créer l'ensemble des modules d'informations qui seront nécessaires à la compréhension et à "l'opérationnalité" de l'utilisateur. Ceci fait, les modules s'assemblent par configuration, soit en fonction de parcours de cet utilisateur, sur un support électronique, soit en fonction d'un ordre pré-établi, pour une publication papier.

Principes

DITA se repose sur une organisation de l'information en *topics* qui sont des fragments d'information autonomes ; charge aux applications d'assembler au mieux les *topics* pour réaliser des publications.

À la base, il existe trois types de *topics* : concept, tâche et référence, tous trois dérivés d'un *topic* de base. Cette spécialisation permet au rédacteur de se pencher sur la nature de l'information qu'il écrit, et par là-même, de la rendre réutilisable dans différents contextes.

Concept

La notion de concept doit permettre de répondre aux questions suivantes :

- Qu'est-ce ?
- Pourquoi est-ce important pour moi ?
- Comment est-ce lié à d'autres concepts ?
- Quand dois-je l'utiliser ?

Tâche

La notion de tâche permet de guider l'utilisateur d'un point à un autre en répondant à la question "comment faire cela ?"

La notion de tâche différencie :

- le contexte,
- les pré-requis,
- les étapes de la tâche,
- les post-requis,
- le résultat obtenu.

Référence

Permet de donner de l'information de référence, par exemple des listes d'identification de variables ou des listes de commandes. L'idée : "passons à l'essentiel".

Ces quatre *topics* (un généraliste et trois spécialisés) sont prévus pour pouvoir être eux-mêmes dérivés, de façon à mieux coller aux structures des entreprises réalisant de la documentation avec DITA. Par ailleurs, il est aussi possible d'étendre le modèle documentaire accessible dans un *topic* particulier ou pour tous les *topics* d'une entreprise : on parle alors de "spécialisation de domaine".

Enfin, DITA repose sur un jeu d'objets connus et réutilisables : les Métadonnées de chaque *topic* utilisent autant que faire se peut le [Dublin Core](#) ; les modèles de contenu utilisent les tableaux [OASIS Tables](#) et une base rédactionnelle non spécialisée liée à [XHTML](#).

Pour finir, la question que devra se poser toute personne souhaitant utiliser ce standard est la suivante : "*comment ferai-je pour retrouver un topic particulier parmi les 1000 modules de données (10 000 modules ?) que j'ai dans ma base de données ?*". DITA étant une spécification liée à l'industrie du logiciel, il ne peut – par essence – pas exister de référentiels génériques de classement des *topics* pour n'importe quel produit logiciel. On peut alors se reposer sur des classifications qui mixent des natures d'information (un "*getting started*", par exemple) avec des objets composés utiles à l'utilisateur (par exemple : "*pour comprendre ce concept, il faut, d'une part sa définition (un topic), d'autre part, un exemple (un autre topic) et, enfin, un "how-to" expliquant comment le mettre en place (un nouveau topic)*").

La méthode proposée par DITA repose sur les "*maps*" qui deviennent aussi spécialisables ; leurs fonctionnement et rôle est défini dans la notion de "*design patterns [sur internet : www-128.ibm.com/developerworks/xml/library/x-dita7/index.html]*" de DITA. Il devient alors possible de considérer qu'un fonds documentaire est composé d'un ensemble de


maps qui sont, par la suite, assemblées pour réaliser des publications. La structure de stockage dans un fonds documentaire est alors la *map*, accessible selon des classifications fonctionnelles-produit (question : "*quelle est la fonction décrite ?*").

Recommandations(s)

 *OASIS DITA Architectural Specification*

Recommandation, version 1.0, du 15-03-2005

Document sur <http://xml.coverpages.org/DITA-CD2-Archspec11883.pdf>

 *OASIS DITA Language Specification*

Recommandation, version 1.0, du 15-03-2005

Document sur <http://xml.coverpages.org/DITA-CD2-Langspec11883.pdf>

TIRÈME SARL

DocBook, DocBook

Rédaction : Pierre Attar

Recommandation(s) liée(s) : [DITA](#)

[DocBook](#) est un modèle de documentation technique actuellement maintenu par un comité technique du consortium *OASIS Open*. Son utilisation s'avère particulièrement intéressante dès lors qu'il s'agit de créer des documents techniques ou des articles.

Cette recommandation est issue d'utilisateurs de l'industrie de l'informatique et de l'électronique. Du coup, il existe beaucoup de structures permettant de coder des informations liées à ce domaine : qu'il s'agisse des messages de spécification d'interface utilisateur ou d'objets et autres classes liées à une documentation issue d'une modélisation objet.

Les modèles issus de [DocBook](#) sont de conception extrêmement moderne et permettent d'ajouter ses propres éléments dans le modèle de base (la recommandation parle alors de "*wrapper*"). De même, si des objets ne sont pas utiles, ils peuvent être aisément supprimés. Beaucoup d'industriels se sont donc emparés de ce modèle et l'ont adapté à leurs propres besoins.

*Note. Certains trouvent que [DocBook](#) est trop difficile à paramétrer et, par conséquent, inutilisable. Les concepteurs travaillent alors aujourd'hui sur une vision simplifiée de [DocBook](#) appelée "*Simplified DocBook*".*

Toujours d'un point de vue conception, le modèle est *disponible* et *extrêmement bien documenté* sur Internet. Mais aussi, sont disponibles des *feuilles de styles*, utilisant [XSLT](#) et [XSL](#), destinées à faciliter la diffusion de ce type d'informations sur le *Web*, sur *Wap*, et sur *papier*.

Quelle sera la prochaine version de [DocBook](#) ? C'est un débat qui agite en ce moment la communauté *OASIS Open*, entre les "pro" de la spécification actuelle et ceux qui veulent, soit en faire quelque chose de plus modulaire quant à sa mise en œuvre, soit quelque chose de plus indépendant de l'industrie de l'informatique et de l'électronique, soit, enfin, quelque chose se rapprochant de [DITA](#) et de l'organisation de la documentation en modules autonomes d'information.

Objectifs

[DocBook](#) est le modèle de documentation technique actuellement maintenu par un comité technique du consortium *OASIS Open*.

[DocBook](#) a maintenant plus de dix ans. À l'origine créé par l'éditeur O'Reilly, son objectif était de simplifier l'échange de documentations UNIX. Puis, avec la création du groupe Davenport, son objectif s'est élargi, pour devenir le format d'écriture et d'échange de documentations techniques issues du monde des industries de l'informatique et de l'électronique.

La vision des documentations proposées par [DocBook](#) est proche du livre. Un document [DocBook](#) est soit un ensemble de livres, soit un livre seul, voire un seul article. Dans ce livre, tout peut être défini : qu'il s'agisse des Métadonnées de gestion comme les copyrights ou noms d'auteurs ; que ce soit aussi l'organisation des contenus par sections, avec des annexes ; qu'il s'agisse, enfin, des informations d'accès que représentent les tables des matières et autres index.

Dans les contenus eux-mêmes, le modèle différencie le niveau des composants (*component*) de celui des objets apparaissant directement dans le texte. Pour les deux types d'éléments, il existe toujours des éléments de présentation et des éléments plus sémantiques, liés à l'industrie visée.

Qui utilise [DocBook](#) ? Tout d'abord la communauté UNIX (plus particulièrement aujourd'hui la communauté LINUX), pour ses manuels. Ensuite, ce sont aussi les industriels et surtout les secteurs de l'Après-Vente. Enfin, avec l'avènement de "simplified [DocBook](#)", une vision simplifiée de [DocBook](#), beaucoup s'emparent du modèle pour l'adapter à leurs besoins, dès lors qu'ils ont besoin d'un modèle "documentaire" de base pour commencer à travailler.

Ce qui fait le succès de [DocBook](#) est, certes, le fait d'exister et d'être soutenu par le consortium *OASIS Open*. C'est aussi le fait que beaucoup d'outils implémentent ce modèle, tant pour créer les documents (*Frame SGML*, *Epic*, *XMLmind*, *XML Spy*, *XMetaL*, etc.) que pour utiliser les données issues de ce modèle. Dans le cadre de l'utilisation, on citera en tout premier lieu les feuilles de styles disponibles en "open source" proposées par **Norman Walsh**, l'éditeur du standard au sein d'*OASIS Open* ; on citera aussi les différentes implémentations disponibles dans les produits libres et/ou commerciaux (*Frame SGML*, *Epic*, *XPP*, etc.)

Principes

Les différents éléments du modèle

Les éléments de [DocBook](#) peuvent être classés selon la hiérarchie suivante :

- . ensembles ;
- . livres ;
- . divisions ;
- . articles ;
- . sections.

À la différence des standards de documentation technique basés sur la réutilisation de fragments pour créer une publication (voir par exemple [S1000D](#)), [DocBook](#) contient toute l'information nécessaire à une documentation technique. Ainsi, et par exemple, un livre comprendra des informations d'identification formelle, tous les composants d'index permettant la navigation, une préface, des chapitres, des annexes, des glossaires et une bibliographie.

Bien sûr, tous ces éléments sont optionnels et c'est à l'utilisateur de choisir ce qu'il veut mettre en place. L'utilisateur peut être restreint dans ces choix et il faut alors paramétrer le modèle.

Aux niveaux les plus bas on trouve les notions de :

Blocs

Ce sont des éléments de type paragraphe, avec un interligne avant et après dès lors qu'ils sont composés.

On trouve parmi eux différentes catégories de listes, d'avertissements, de synopsis, de tableaux, de figures, d'exemples, de média comme la vidéo, etc.

On trouve aussi des éléments plus sémantiques comme ce qui permet de décrire, en ingénierie logicielle, des options et des paramètres d'une commande. Pour décrire des programmes, on trouvera ce qui permet d'identifier des fonctions, leurs arguments et valeurs de retour, etc ; il y aura de la même façon les notion de message, voire de procédure de maintenance.

La provision pour ce type d'élément est extrêmement vaste, d'où la nécessité, souvent, d'adapter [DocBook](#) à ses propres besoins, afin de ne pas être obligé de manipuler tous les composants [DocBook](#).

Éléments en ligne

Ce sont les éléments que l'on retrouve directement mixés avec les phrases, les mots et les caractères.

À nouveau, on y trouve des éléments de présentation comme la mise en relief, les citations, les abréviations et autres acronymes. On trouve aussi tout ce qui permet de gérer les renvois (à des sections, des tableaux, des médias, etc.).

À nouveau aussi, on trouve un certain nombre d'informations liées à l'industrie visée comme des objets d'identification d'interface utilisateur, d'objets d'un langage de programmation, d'un système d'exploitation, etc.

Avec cette provision d'éléments, il est possible de réaliser ses propres documentations techniques, dès lors que l'on appartient à l'industrie visée. Qu'en est-il alors des utilisateurs d'une industrie différente ? Qu'en est-il aussi des utilisateurs de cette industrie qui ont des besoins beaucoup plus limités ?

C'est là qu'intervient la force de [DocBook](#). En effet, le modèle est conçu pour qu'il soit possible soit d'ajouter de nouvelles structures, soit d'en supprimer. Les deux opérations se font par redéfinition et la méthode fonctionne tant lorsque l'on utilise les [DTD](#) que lorsque l'on utilise les [Schema](#).

Modèle simplifié

Si [DocBook](#) contient autant d'objets structurels, c'est qu'ils sont nécessaires pour atteindre tous les objectifs documentaires de l'industrie visée. On a par ailleurs dit qu'il était possible de simplifier l'utilisation de [DocBook](#) par redéfinition du modèle.

Pour prouver la faisabilité de la chose et pour proposer aux implémenteurs une solution finalisée simple, les membres du comité [DocBook](#) proposent un modèle dit simplifié.

Dans ce modèle, beaucoup d'éléments ont été supprimés pour ne conserver que le strict minimum, toujours dans le cadre d'un utilisation en documentation technique.

Ce travail a été réalisé dans les "règles de l'art", par redéfinition du modèle complet.

Pour réaliser sa propre application, il est alors possible de choisir parmi les possibilités suivantes :

- utilisation de tout [DocBook](#) ;
- utilisation du modèle simplifié ;
- redéfinition de [DocBook](#) ;
- redéfinition du seul modèle simplifié.

Après avoir travaillé en redéfinition sur les deux modèles, la question est de savoir s'il existe une préférence sur celui sur lequel s'appuyer. Le choix est uniquement lié à la complexité du modèle souhaité et à ce que l'on souhaite utiliser dans la provision d'éléments **DocBook**. Il faut donc, dans un premier temps choisir le cadre dans lequel on s'insère, puis le redéfinir.

Informations connexes

On notera l'implémentation de **DocBook** réalisée dans le cadre de *Tirème* pour mettre en place la technologie SCHEMADOC : un atelier de documentation de modèles issus d'**XML** et représentés sous forme de **Schema**, voire de **DTD**.

Recommandations(s)

 *DocBook XML*

Recommandation, version 4.5, du 01-10-2006

Document sur <http://www.docbook.org/specs/docbook-4.5-spec.html>

 *DocBook SGML*

Recommandation, version 4.2, du 17-07-2002

Document sur <http://www.oasis-open.org/docbook/sgml/4.2/>

 *"Simplified" DocBook*

Recommandation, version 1.0, du 19-11-2002

Document sur <http://www.oasis-open.org/docbook/specs/cs-docbook-simple-1.0.html>

TIRÈME SARL

UIML, User Interface Markup Language

Objectifs

Recommandations(s)

 *UIML 3.0*

Projet en cours, version 3.0, du 12 February 2002


Document sur <http://www.uiml.org/specs/docs/uiml30-revised-02-12-02.pdf>

TIRÈME SARL

XUL, XML User Interface Language

Objectifs

Recommandations(s)

-  *XML User Interface Language*
Projet en cours, version 1.0, du 2001
Document sur <http://www.mozilla.org/projects/xul/xul.html>



J2008, SAE J2008

Issue du gouvernement fédéral américain, la **J2008** est une norme d'échange d'information documentaire Après Vente entre les *constructeurs automobiles* et *leurs clients*. Elle permet d'échanger des fragments documentaires, ainsi que l'information d'applicabilité nécessaire à l'utilisation de ces fragments. La **J2008** ne s'intéressent qu'à l'*échange unidirectionnel d'un constructeur vers un client*, les structures d'information proposées sont liées au point de vue de l'utilisateur, plus qu'à celui du producteur de documentation. Elle se rapproche alors des grandes normes sectorielles, telles celles de l'aéronautique civile ou militaire.

Une dernière remarque : si la **DTD** est disponible, la documentation de celle-ci est indispensable pour comprendre le modèle. Cette documentation n'est pas disponible en accès libre et doit être commandée directement auprès des concepteurs de la recommandation.

Objectifs

La **J2008** définit et organise la façon dont de l'information documentaire est transmise au client. En ce sens, c'est un format sémantique permettant de transférer des éléments documentaires de réparation, l'information nécessaire à la gestion de configuration de ces informations et la façon dont ils peuvent s'insérer dans le système d'information de destination.

A priori, la **J2008** ne s'intéressant qu'à l'échange unidirectionnel d'un constructeur vers un client. Du coup, les structures d'information proposées sont liées à un point de vue utilisateur de documentation, plus qu'à un point de vue de producteur de documentation. Elle se rapproche alors des grandes normes sectorielles, telles celles de l'aéronautique civile ou militaire.

La structure d'un transfert **J2008** est organisée de la façon suivante :

- Liste des éléments d'information de service (*ListofSIEs*)

- Elle s'apparente à une déclaration définissant la liste des SIE (*Service Information Elements*) transférés ;

- Ensembles de SIE (*ServInfoPool*) :

- Contient, de façon non ordonnée, tous les SIE transférés. Chacun est identifié par un numéro unique (*servinfoSGMLid*) qui est référencé dans la liste des SIE. Il est aussi possible de définir des SIE dans les PATH.

- Définition du modèle de donnée utilisé (*OEMinfo*) :

- C'est l'espace de définition des objets et critères qui permettent d'organiser l'information. A part quelques informations standardisées par la **J2008** (langue, pays, jeux de caractères, types de freinage, type de conduite, identification de constructeur, etc.), le constructeur est responsable de sa base de configuration. Sa seule obligation : la transférer pour que le récepteur puisse en comprendre la nature.

Cependant, la base doit être structurée selon un modèle conceptuel défini dans la J2008. Qui identifie, par exemple, des véhicules et leurs variantes, des configurations, etc. De façon macroscopique, 14 classes d'informations sont définies. Chaque élément d'une classe est identifié de façon unique.

Chemins d'accès (Paths) :

les paths permettent de regrouper des SIE, ou des références à des SIE par ailleurs définis dans le transfert, selon des critères. Les critères des path sont autant de références aux éléments définis dans les OEMInfos.

Toute l'information d'un transfert est contenue dans un même et unique fichier XML.

Exemple

L'exemple le plus simple correspond au transfert d'une information élémentaire de maintenance quelconque.


Le fichier à écrire pourrait être le suivant :

```
<!DOCTYPE J2008 SYSTEM "XMLJ2008.ent">
<J2008 mfrcode="111111111" revdate="1-12-2000"
mktgdivcode="22222222" charsetnbr="8">
  <ListofSIEs>
    <SIE servinfoSGMLid="nb010902001" ldup="1dec2000"
      update="original"/>
  </ListofSIEs>
  <ServInfoPool>
    <ServInfo servinfoSGMLid="nb010902001" update="original"
      ldup="1dec2000" svcinfotypenbr="XXX" svcinfoqualnbr="XXX">
      <Title>Mon unité de maintenance</Title>
      <Ptxt>Voici le texte qui correspond au démontage de ...</
      Ptxt>
    </ServInfo>
  </ServInfoPool>
  <OEMinfo>
    <Vehicles>
      <Vehicle vehSGMLid="XBZ" countrycode="FRA" year="2000"
        yeartype="production" modelid="XXX" vehtypenbr="XXX"
        modeldesc="XXX" update="original" ldup="1jan2000"/>
    </Vehicles>
  </OEMinfo>
  <Paths>
    <Path1 vehSGMLid="XBZ" >
      <ServInfo ref servinfoSGMLid="nb010902001" />
    </Path1>
  </Paths>
</J2008>
```

Principes

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *SAE J2008 : Recommended Organization of Vehicle Service Information for Interchange*

Recommandation, version 4.1, du 01-06-1998

Document sur <http://www.sae.org/servlets/otherProduct?>

PROD_TYP=SFTWR&PROD_CD=J2008CD

TIRÈME SARL

S1000D, Specification S1000D

Rédaction : Yves Keraron, Pierre Attar

Issue de l'industrie aéronautique et spatiale de défense européenne (autrefois *AECMA*, aujourd'hui *ASD* comme *AeroSpace and Defense*), la norme **S1000D** s'est internationalisée depuis la version 2.1 qui a été produite conjointement avec l'*AIA* (*Aerospace Industries Association of America*).

La norme **S1000D** est donc une spécification internationale pour l'approvisionnement et la production de publications techniques et son domaine d'application s'étend, depuis la version 2, aux domaines civil et militaire et aux équipements aéronautiques, terrestres et maritimes.

La spécification s'appuie sur les standards technologiques **SGML/XML** et **CGM** pour définir les règles de production de modules documentaires stockés dans une *base de données source commune* et à partir de laquelle sont réalisées les publications destinées à l'utilisateur final.

Objectifs

L'élaboration de la **S1000D** vise les principaux objectifs généraux suivants :

- réduction des coûts,
- organisation plus économique du soutien logistique,
- définition de règles communes pour les participants d'un projet,
- amélioration de l'interopérabilité,
- amélioration de la clarté des contenus et réduction des coûts de traduction par l'utilisation de l'*Anglais simplifié* (*SE, Simplified English*), norme également supportée par l'*ASD*.

La norme **S1000D** est associée à la norme **S2000M** pour les échanges d'informations relatifs au processus d'approvisionnement des pièces de rechange.

Principes

L'information est stockée dans des modules de données dont les structures sont définies par des **DTD SGML/XML** ou des **Schema** modulaires. L'*approche modulaire* vise à créer des publications au moyen de *Data Module* (DM) stockés dans une *base de données source commune* : les modules sont créés une fois et utilisés plusieurs fois.

La *structure matricielle de l'information* entre le sujet concerné, identifié par sa place dans une *arborescence*, et la nature de l'information concernée (description, procédure, ...) est identifiée par un **code information**.

Le module de données ou *Data Module* (DM) est une unité d'information autonome et réutilisable comprenant des descriptions, des procédures ou des données opérationnelles pour un matériel ou un composant. Il est produit de telle manière qu'il peut être classé et retrouvé dans une *base de données source commune* (CSDB, *Common Source Data Base*) avec, comme identifiant, son code de module de données (DMC, *Data Module Code*).

Le module de données comprend une identification, un statut et un contenu. Il appartient à une arborescence. Plusieurs arborescences sont proposées dans la S1000D selon le type d'équipement à documenter. Un DM peut être utilisé sur l'équipement principal, sur un équipement de soutien ou pour la formation. Un DM est affecté à un ou plusieurs niveaux de maintenance.

Le contenu du module de données est produit en SGML/XML selon des DTD et des Schema spécifiques correspondant à des contenus typés.

LA CSDB est un espace de stockage pour les modules de données, les illustrations et les publications. Les DM stockés dans la CSDB participent à la constitution de publications papier ou électronique sous la forme d'ensemble d'information (*InfoSet*). Les DM sont stockés dans la CSDB selon un processus d'échange formel. Chaque livraison est accompagnée d'un bon de livraison et d'une liste de DM. Le DMC est un code structuré permettant d'identifier un module de données et d'en faciliter le classement et la recherche au sein de la CSDB. L'ICN (*Illustration Control Number*) est un code permettant d'identifier une illustration et d'en faciliter le classement et la recherche au sein de la CSDB. L'*applicabilité/effectivité* définit la *version/lot/n° série* du matériel auquel s'adresse le module de données.

Le DMC codifie le DM et permet de nommer le fichier informatique correspondant. Le code comprend les éléments suivants :

- . le code d'identification du modèle (MIC),
- . le code variante (SDC, *System Difference Code*),
- . le code classement dans l'arborescence (SNS, *Standard Numbering System*),
- . le code de désassemblage et sa variante (DC),
- . le code information et sa variante (IC),
- . le code de localisation de l'item et de son contexte de maintenance – par exemple sur l'installation, déposé??, en atelier – (IL).

Le statut contient des informations de gestion sur la sécurité, la confidentialité, la société responsable, l'applicabilité, la date de dernière révision, la raison de mise à jour, l'indicateur assurance-qualité...

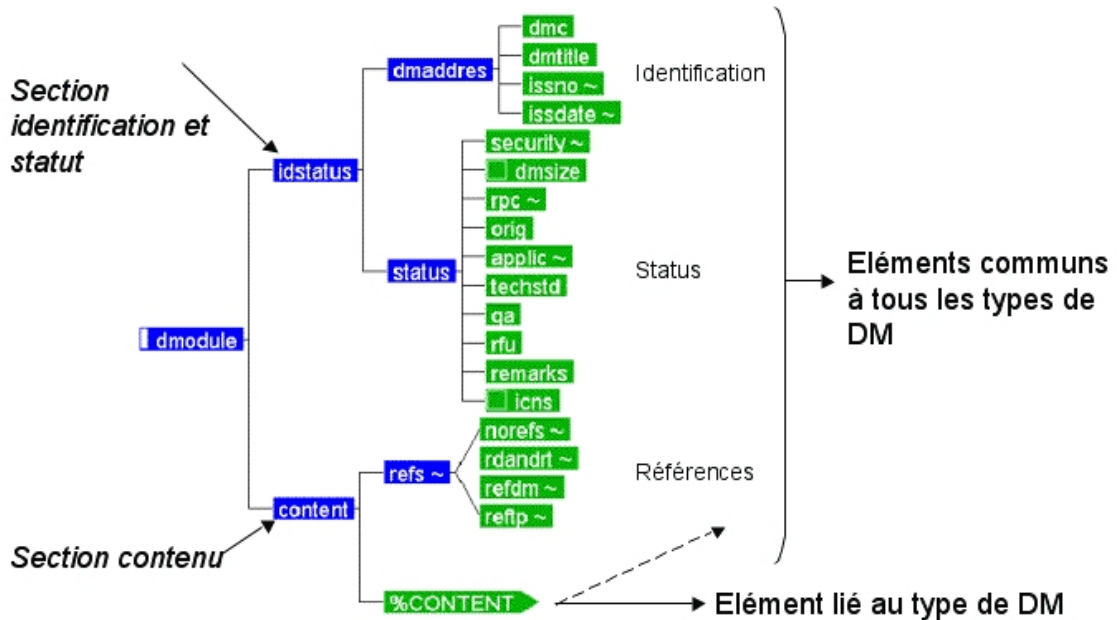
Chaque type de contenu fait l'objet d'une DTD modulaire :

- . description et fonctionnement,
- . procédures,
- . localisation de pannes,
- . maintenance programmée,
- . catalogue illustré,
- . personnel navigant.

Le contenu est structuré avec des éléments SGML/XML qui se regroupent en plusieurs familles : fonction, statut, données techniques, pré-requis, information, illustration, navigation, applicabilité, révision, publication.

Figure. La structure d'un module de données

Figure 1 - Formalisation de la structure du DM (DTD)



La S1000D dispose à présent d'un retour d'expérience de plusieurs années depuis la production des modules jusqu'à leur utilisation. La S1000D ne traite pas des outils qui facilitent l'appropriation du contenu des modules. Elle ne traite pas, par exemple, des aspects liés à la production et à la gestion d'annotations. Elle n'en constitue pas moins une très bonne base pour l'implémentation d'outils complémentaires d'aide à l'appropriation des contenus et à l'évolution du fonds documentaire.

Recommandations(s)

 *International Specification for Technical Data Publications utilising a common source Data Base, Specification 1000D*

Recommandation, version 4.0.1, du 05-06-2009

Document sur <http://public.s1000d.org/Downloads/Pages/S1000DDownloads.aspx>

TIRÈME SARL

FOSI, Formatting Output Specification Instance

Recommandation(s) liée(s) : [DSSSL](#) - [XSL](#)

Définie sous forme de document [SGML](#), une [FOSI](#) (*Formatting Output Specification Instance*) est une feuille de style qui permet de formater un document [SGML](#) pour une sortie papier.

Cette spécification est issue du Département de la Défense américain et son objectif est de permettre la composition de documentations techniques. À ce titre, elle ne permet pas de réaliser toutes les opérations nécessaires à l'activité traditionnelle de pré-presse.

Par ailleurs, cette spécification correspond à une stratégie d'attente de la norme internationale [DSSSL](#). On retrouve alors dans la [FOSI](#) tous les concepts de [DSSSL](#) et, donc, ceux qui forment les fondements de [XSLT](#) et de [XSL](#).

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *Markup Requirements and Generic Style Specification for Exchange of Text and its Presentation*

Recommandation, version MIL-PRF-28001C, du 02-05-1997

Document sur <http://www.dt.navy.mil/tot-shi-sys/des-int-pro/tec-inf-sys/sgm/sgm-std/doc/28001c.pdf>

TIRÈME SARL

ebXML, Electronic Business XML

Recommandation(s) liée(s) : *ebXML Registry*

« *ebXML Mission: to provide an open XML-based infrastructure enabling the global use of electronic business information in an interoperable, secure and consistent manner by all parties.* »

Source. *ebXML Technical Architecture Specification, version 1.0.4*

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *ebXML Technical Architecture Specification*

Recommandation, version 1.0.4, du 16-02-2001

Document sur <http://www.ebxml.org/specs/ebTA.pdf>

 *Collaboration-Protocol Profile and Agreement Specification*

Recommandation, version 2.0, du 23-09-2002

Document sur <http://www.ebxml.org/specs/ebcpp-2.0.pdf>

 *Message Service Specification, ebXML Transport, Routing & Packaging*

Recommandation, version 2.0, du 01-04-2002

Document sur <http://www.ebxml.org/specs/ebMS.pdf>

 *ebXML Business Process Specification Schema*


Recommandation, version 1.10, du 18-10-2003

Document sur <http://xml.coverpages.org/UNCEFACT-ebBPSS-v1pt10.pdf>

 *ebXML Requirements Specification*

Expression des besoins, version 1.06, du 08-05-2001

Document sur <http://www.ebxml.org/specs/ebREQ.pdf>

 *ebXML Registry Information Model*

Recommandation, version 2.1, du 06-2002

Document sur http://www.oasis-open.org/committees/regrep/documents/2.1/specs/ebrim_v2.1.pdf

 *ebXML Registry Services Specification*

Recommandation, version 2.1, du 06-2002

Document sur <http://www.oasis-open.org/committees/regrep/documents/2.1/specs/ebrs.pdf>

TIRÈME SARL

BizTalk, BizTalk Framework

« As a result, domain-specific standards bodies and industry initiatives have started to adopt XML and XML-based schema languages to specify both their vocabularies and content models. These schemas are becoming widely published and implemented to facilitate communication between both applications and businesses. Wide support of XML has also resulted in independent solution providers developing solutions that enable the exchange of XML-based information with other third-party or custom-developed applications. Several solution- or middleware/platform-specific approaches have been taken to address the lack of middleware-neutral, application-level communication protocols. However, no single proprietary solution or middleware platform meets all the needs of a complex deployment environment.

These proprietary initiatives have generally resulted in customers facing broad interoperability issues on their own. The BizTalk Framework addresses these interoperability challenges in a platform- and technology-neutral manner. It provides specifications for the design and development of XML-based messaging solutions for communication between applications and organizations. This specification builds upon standard and emerging Internet technologies such as Hypertext Transfer Protocol (HTTP), Multipurpose Internet Mail Extensions (MIME), Extensible Markup Language (XML), and Simple Object Access Protocol (SOAP). Subsequent versions of the BizTalk Framework will be enhanced to leverage additional XML and Internet-related, messaging-standards work as appropriate.


It is important to note that the BizTalk Framework does not attempt to address all aspects of business-to-business electronic commerce. For instance, it does not deal directly with legal issues, agreements regarding arbitration, and recovery from catastrophic failures, nor does it specify specific business processes such as those for purchasing or securities trading. The BizTalk Framework provides a set of basic mechanisms required for most business-to-business electronic exchanges. It is expected that other specifications and standards, consistent with the BizTalk Framework, will be developed for the application- and domain-specific aspects. »

Source. BIZTALK Framework: Document and Message Specification, version 2.0

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 BIZTALK Framework: Document and Message Specification
Recommandation, version 2.0, du 18-04-2001

Document sur <http://www.microsoft.com/biztalk/techinfo/whitepapers/2000/framwork20.mspx>

TIRÈME SARL

EDI, Echange de données électroniques


« The primary purpose of the UN/EDIFACT process is to provide standard messages to users to cover their information exchange needs. In theory, taking into account the diversity of information exchange supporting all kinds of business, the number of messages that may be standardised, if not infinite, is great. However, the design approach adopted by UN/EDIFACT of using generic rather than specific data structures promotes reusability and hence helps to limit the number of potential messages. This decision was taken in order to ensure that the UN/EDIFACT standard is not merely a collection of disparate structures corresponding to specific requirements. Rather, the generic approach is based upon shared common exchange structures and allows for the design of fewer structures to support a wide range of requirements thus reducing the maintenance work load »

Source. *Message and Code Handbook. Best Practices for Designers, version 1.0*

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *Message and Code Handbook. Best Practices for Designers*

Note, version 1.0, du 29-01-1998

Document sur <http://www.unece.org/trade/untdid/download/98g104.doc>

TIRÈME SARL

XBRL, eXtensible Business Reporting Language

« *XBRL is a language for the electronic communication of business and financial data which is set to revolutionise business reporting around the world. It provides major benefits in the preparation, analysis and communication of business information. It offers cost savings, greater efficiency and improved accuracy and reliability to all those involved in supplying or using financial data.*

XBRL stands for eXtensible Business Reporting Language. It is one of a family of "XML" languages which is becoming a standard means of communicating information between businesses and on the internet.

XBRL is being developed by an international non-profit consortium of approximately 250 major companies, organisations and government agencies. It is an open standard, free of licence fees. It is already being put to practical use in a number of countries and implementations of XBRL are growing rapidly around the world. »

Source. *Extensible Business Reporting Language (XBRL), version 2.1*

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *Extensible Business Reporting Language (XBRL)*

Recommandation, version 2.1, du 25-04-2005

Document sur [http://www.xbrl.org/Specification/XBRL-](http://www.xbrl.org/Specification/XBRL-RECOMMENDATION-2003-12-31+Corrected-Errata-2005-04-25.htm)

[RECOMMENDATION-2003-12-31+Corrected-Errata-2005-04-25.htm](http://www.xbrl.org/Specification/XBRL-RECOMMENDATION-2003-12-31+Corrected-Errata-2005-04-25.htm)

 *XBRL 2.1 Conformance Suite 1.0*

Recommandation Candidate, version 2.1, du 25-04-2005

Document sur <http://www.xbrl.org/2005/XBRL-CONF-CR1-2005-04-25.htm>

TIRÈME SARL

SDMX, Echange de données et de Métadonnées statistiques (SDMX)

Rédaction : Pierre Attar

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

« The BIS, ECB, EUROSTAT, IMF, OECD, UN, and the World Bank have joined together to focus on business practices in the field of statistical information that would allow more efficient processes for exchange and sharing of data and metadata within the current scope of our collective activities.

The goal is to explore common e-standards and ongoing standardization activities that could allow us to gain efficiency and avoid duplication of effort in our own work and possibly for the work of others in the field of statistical information. We intend to do this by taking advantage of existing and emerging: # dissemination formats, such as that implicit in the IMF's Dissemination Standards Bulletin Board (DSBB); and # e-standards, such as Extensible Markup Language (XML).

We intend to do this by taking advantage of existing and emerging :

- . exchange protocols, such as GESMES/CB which was implemented by central banks for exchanging time series;*
- . dissemination formats, such as that implicit in the IMF's Dissemination Standards Bulletin Board (DSBB); and*
- . e-standards, such as Extensible Markup Language (XML).*

Commission économique des Nations Unies pour l'Europe

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *Statistical Data and Metadata Exchange (SDMX) - Technical Standards*

Recommandation, version 2.0, du 11-2005

Document sur http://www.sdmx.org/index.php?page_id=16#package

TIRÈME SARL

GESMES, Message statistique générique (GESMES)

Rédaction : Pierre Attar

Recommandation(s) liée(s) : *SDMX*

« GESMES/TS uses the EDIFACT syntax (ISO 9735) and United Nations (UN/ECE) segment directories, both of which are international standards. EDIFACT messages can sometimes look syntactically complex. However, GESMES/TS has been designed so that its syntactic structure is simple, being predefined and frozen; its flexibility derives exclusively from the power of its underlying data model.

GESMES/TS is maintained by the SDMX initiative led by seven international and European organisations (BIS, ECB, Eurostat, IMF, OECD, UN, World Bank). »

Banque Centrale Européenne [sur internet : www.ecb.int/stats/services/gesmes/html/index.en.html]

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *Generic statistical message/time series (GESMES/TS)*

Recommandation, version 28-02-2003, du 28-02-2003

Document sur http://www.sdmx.org/Data/GesmesTS_rel3.pdf

TIRÈME SARL

ISO 12083 EMS, The Electronic Manuscript Standard

« *This International Standard presents four document type definitions and additional facilities conforming to ISO 8879. This International Standard is intended to provide document architectures for the creation and interchange of books, articles and serial publications.*

It specifies:

- a) the SGML declaration defining the syntax used by the document type definitions and document instances;*
- b) the document type definitions for the following document classes: i) Books; ii) Articles; iii) Serials, which are collections of articles;*
- c) a document type definition for Mathematics which may be embedded in other SGML applications.*

This International Standard provides a toolkit for developers of these types of publications to use as a basis for developing custom applications.

The applications developed may be for authors, publishers, libraries, their users, and/or their database vendors.

The document type definitions included in this International Standard have been designed to be flexible enough to expand or limit markup depending on the use of the application.

For example, if the application is intended for authors, the copyright information could be removed. The markup is sufficiently general to enable its use for similarly structured documents, e.g. the Book DTD can also be used for technical reports. »

Source. *Electronic Manuscript Preparation and Markup, version 12083*

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *Electronic Manuscript Preparation and Markup*

Standard ISO, version 12083, du 17-04-1995

Document sur http://www.techstreet.com/cgi-bin/detail?product_id=52643

TIRÈME SARL

NewsML, News markup language

« *NewsML is an XML encoding for news which is intended to be used for the creation, transfer, delivery and archiving of news.*

NewsML is media independent, and allows equally for the representation of the evening TV news and a simple textual story. Specifically, NewsML provides the following features:

- . All formats and media types recognised equally*
- . Facilitates the development of NewsItems*
- . Collections of NewsItems*
- . Named relationships within and between NewsItems*
- . Structure consisting of ContentItems, NewsComponents and named relationships between NewsComponents*
- . Alternative representations within the same NewsComponent*
- . Explicit inclusion, inclusion by reference and exclusion of NewsComponents and alternatives*
- . Attachment of metadata from standard and non-standard Controlled Vocabularies*
IPTC

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *NewsML, Functional Specification*

Recommandation, version 1.2, du 10-10-2003

Document sur http://www.newsml.org/IPTC/NewsML/1.2/specification/NewsML_1.2-spec-functionalspec_8.html

TIRÈME SARL

Open Office XML, Open Office XML

The purpose of the TC (Technical Committee) is to create an open, XML-based file format specification for office applications. The resulting file format must meet the following requirements:

- it must be suitable for office documents containing text, spreadsheets, charts, and graphical documents,*
- it must be compatible with the W3C Extensible Markup Language (XML) v1.0 and W3C Namespace in XML v1.0 specifications,*
- it must retain high-level information suitable for editing the document,*
- it must be friendly to transformations using XSLT or similar XML-based languages or tools,*
- it should keep the document's content and layout information separate such that they can be processed independently of each other, and*
- it should 'borrow' from similar, existing standards wherever possible and permitted.*

Since the OpenOffice.org XML format specification was developed to meet these criteria and has proven its value in real life, this TC will use it as the basis for its work.

Sun Microsystems intends to contribute the OpenOffice.org XML Format to this TC at the first meeting of the TC, under reciprocal Royalty Free terms.


A standard for office document processing and interchange will be of great utility to many users and software companies developing applications, and should be made available as soon as possible. »

OASIS Open Office XML Format TC [sur internet : www.oasis-open.org/committees/office/charter.php]

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

-  *Open Document Format for Office Applications (OpenDocument) Recommendation, version 1.0 (second edition), du 04-03-2005*
Document sur <http://www.oasis-open.org/committees/download.php/12027/office-spec-1.0-cd-3.pdf>



Open XML, Open XML


« Ecma a finalisé la normalisation du format Open XML et qui est désormais référencé "Ecma 376". L'Open XML est un format ouvert et documenté, basé sur les schémas XML, à l'origine défini par Microsoft. Il accompagne notamment la suite bureautique Microsoft Office 2007 disponible en version licence depuis le 30 novembre 2006 et à partir du 30 janvier 2007 en boîtes et CEM. Open XML assure une compatibilité ascendante avec les formats de documents, dits binaires, jusqu'alors générés sous les précédentes versions d'Office (bien que le XML a fait son apparition dans Office 2003). Ce qui concerne aujourd'hui des milliards de documents dans le monde. »

[sur internet : www.vnunet.fr/fr/vnunet/news/2006/12/08/l-open-xml-d-sormais-un]vnunet

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *ISO - Office Open XML File Formats — Part 1: Fundamentals and Markup Language Reference*

Recommandation, version ISO/IEC 29500-1:2008, du 15-11-2008

Document sur [http://www.iso.ch/iso/search.htm?](http://www.iso.ch/iso/search.htm?qt=29500&searchSubmit=Search&sort=rel&type=simple&published=on)

[qt=29500&searchSubmit=Search&sort=rel&type=simple&published=on](http://www.iso.ch/iso/search.htm?qt=29500&searchSubmit=Search&sort=rel&type=simple&published=on)

 *ISO - Office Open XML File Formats - Part 2: Open Packaging Conventions*

Recommandation, version ISO/IEC 29500-2:2008, du 15-11-2008

Document sur [http://www.iso.ch/iso/search.htm?](http://www.iso.ch/iso/search.htm?qt=29500&searchSubmit=Search&sort=rel&type=simple&published=on)


[qt=29500&searchSubmit=Search&sort=rel&type=simple&published=on](http://www.iso.ch/iso/search.htm?qt=29500&searchSubmit=Search&sort=rel&type=simple&published=on)

 *ISO - Office Open XML File Formats - Part 3: Markup Compatibility and Extensibility*

Recommandation, version ISO/IEC 29500-3:2008, du 15-11-2008

Document sur [http://www.iso.ch/iso/search.htm?](http://www.iso.ch/iso/search.htm?qt=29500&searchSubmit=Search&sort=rel&type=simple&published=on)

[qt=29500&searchSubmit=Search&sort=rel&type=simple&published=on](http://www.iso.ch/iso/search.htm?qt=29500&searchSubmit=Search&sort=rel&type=simple&published=on)

 *ISO - Office Open XML File Formats - Part 4: Transitional Migration Features*

Recommandation, version ISO/IEC 29500-4:2008, du 15-11-2008

Document sur [http://www.iso.ch/iso/search.htm?](http://www.iso.ch/iso/search.htm?qt=29500&searchSubmit=Search&sort=rel&type=simple&published=on)

[qt=29500&searchSubmit=Search&sort=rel&type=simple&published=on](http://www.iso.ch/iso/search.htm?qt=29500&searchSubmit=Search&sort=rel&type=simple&published=on)

 *ISO - Freely Available Standards - Office Open XML File Formats. Part 1, 2, 3, 4*

Recommandation, version First Edition. Part 1-4, du 15-11-2008

Document sur <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>

 *Standard ECMA-376. Office Open XML File Formats Open XML*

Recommandation, version ECMA-376 Edition 2, du 08-12-2008

Document sur <http://www.ecma-international.org/publications/standards/Ecma-376.htm>

TIRÈME SARL

TEI, Text Encoding Initiative


« *The TEI is an international project to develop guidelines for the encoding of textual material in electronic form for research purposes* »

Source. *Guidelines for Electronic Text Encoding and Interchange, version P4*

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 *Guidelines for Electronic Text Encoding and Interchange*
Recommandation, version P4, du 15-06-2002
Document sur <http://www.tei-c.org/P4X/>

 *TEI Lite*
Recommandation, version U5, du 2002
Document sur http://www.tei-c.org/Lite/teiu5_en.html

TIRÈME SARL

OEB, Open EBook

Rédaction : Pierre Attar

Recommandation(s) liée(s) : [ePub](#)

La spécification Open EBook définit les données nécessaires à la création d'un livre électronique. Elle différencie le "*package*" (l'ensemble des informations d'identification des Métadonnées et des fichiers utiles) et les contenus informationnels eux-mêmes. Grâce à ce format standardisé, il devient alors possible de créer un livre électronique formel, qui peut ensuite s'adapter à un matériel particulier.

Attention : cette spécification est maintenant remplacée par [ePub](#) qui lui succède.

Objectifs

La recommandation a pour objectif de formaliser les fonctionnalités minimales que devraient partager n'importe quel livre électronique, définissant par là-même un format de fichier destiné à échanger des livres à publier. Il devient alors possible de créer des livres électroniques de façon abstraite, pour, ensuite, les intégrer sur différents matériels physiques.

En théorie, cela fonctionne parfaitement. En pratique, cet objectif doit être revu à la baisse : autant cet objectif de compatibilité peut être effectif pour deux matériels de même nature, autant il devient difficile à réaliser si le même livre électronique fait l'objet d'une publication, d'une part, sur un assistant personnel de type "palmPilot" et, d'autre part, sur un livre électronique de taille "A5". En effet, l'organisation et la nature de l'information sera différente, car la taille de l'écran et les moyens d'interactions sont différents.

Par ailleurs, est-ce qu'un matériel livre électronique doit être "compatible OEB" ? Certainement pas. En effet, il doit obligatoirement utiliser le format le plus efficace pour organiser ses données et assurer la confidentialité de celles-ci. En revanche, il doit exister des outils permettant de convertir un *package* OEB en ce format propriétaire. Ainsi, *Microsoft Reader*, utilise un format binaire dénommé "lit", mais il existe différents outils qui permettent de convertir de l'OEB en "lit".

Du coup, l'intérêt de cette spécification est de définir un jeu de fonctionnalités et son langage de codage. Un éditeur pourra alors réaliser, si nécessaire, différents livres électroniques, pour différents matériels, mais il utilisera toujours le même langage et les mêmes fonctionnalités. Son investissement, en termes de connaissances, sera alors partagé pour tous les différents matériels.

Principes

Un livre **OEB** est défini dans un "*package*" où sont données les identifications du livre, ses composants élémentaires et son organisation.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE package PUBLIC "-//ISBN 0-9673008-1-9//DTD OEB 1.0
Package//EN" "oebpkg1.dtd">
<package unique-identifiant="guide">
  <metadata>
    <dc-metadata xmlns:dc="http://purl.org/dc/elements/1.0/"
xmlns:oebpackage="http://openebook.org/namespaces/oeb-
package/1.0/">
      <dc:Title>Guide de L'ORB</dc:Title>
      <dc:Type>text</dc:Type>
      <dc:Identifiant>A28</dc:Identifiant>
      <dc:Editeur>MUTU éditions, Paris</dc:Editeur>
      <dc:Date>2001</dc:Date>
      <dc:Créateur role="aut" file-as="Attar, Pierre">Pierre
Attar</dc:Créateur>
      <dc:Description>Le premier guide XML sous forme de livre
électronique</dc:Description>
      <dc:Langue>fr</dc:Langue>
      <dc:Rights>Tous droits de traduction, d'adaptation,
réservés</dc:Rights>
    </dc-metadata>
  </metadata>
  <manifest>
    <item id="TOC" href="./TOC.html" media-type="text/x-oeb1-
document"/>
    <item id="guide" href="./guide.html" media-type="text/x-oeb1-
document"/>
    <item id="logo_img" href="./logo.gif" media-type="image/gif"/>
    <item id="image_img" href="./img.gif" media-type="image/jpg"/>
  </manifest>
  <spine>
    <itemref idref="guide_therapeutique"/>
    <itemref idref="TOC"/>
  </spine>
  <guide>
    <reference type="toc" title="Sommaire" href="TOC.html"/>
  </guide>
</package>
```

Le livre ci-devant est donc un guide, avec un titre, un auteur, un éditeur, etc., ... tout ceci défini sous forme de Métadonnées **Dublin Core**. Ce livre est composé de deux fichiers textuels et de deux images. Les deux fichiers textuels apparaissent directement dans le corps du livre, en commençant par le fichier "TOC". Ce même fichier joue aussi le rôle de table des matières.

Les fichiers textuels sont codés selon une version réduite de **HTML** version 4. Celle-ci fait un usage intensif des feuilles de styles **CSS**, y compris pour y ajouter de nouveaux éléments comme les en-têtes et pieds de pages.

Recommandations(s)

 *Open eBook Publication Structure*

Recommandation, version 1.2, du 27-08-2002

Document sur <http://www.idpf.org/oebps/oebps1.2/index.htm>

TIRÈME SARL

ePub, ePub

Rédaction : Pierre Attar


Comme OEB à qui elle succède, la spécification Open EBook définit les données nécessaires à la création d'un livre électronique.

".epub" is the file extension of an XML format for reflowable digital books and publications. ".epub" is composed of three open standards, the Open Publication Structure (OPS), Open Packaging Format (OPF) and Open Container Format (OCF), produced by the IDPF. ".epub" allows publishers to produce and send a single digital publication file through distribution and offers consumers interoperability between software/hardware for unencrypted reflowable digital books and other publications. The Open eBook Publication Structure or "OEB", originally produced in 1999, is the precursor to OPS.

Objectifs

Descriptif en cours de rédaction.

Recommandations(s)

 *EPUB 3 Overview*

Projet en cours, version 3, du 15-02/2011

Document sur <http://www.idpf.org/epub/30/spec/epub30-overview.html>

TIRÈME SARL

UML, Unified Modeling language

Rédaction : Pierre Attar

Recommandation(s) liée(s) : *XMI*






« *The Unified Modeling Language (UML) is a language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of the best engineering practices that have proven successful in the modeling of large and complex systems.* »

Source. *Unified Modeling Language (UML), version 1.5*

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

-  *Unified Modeling Language (UML)*
Recommandation, version 1.5, du 06-03-2003
Document sur <http://www.omg.org/cgi-bin/doc?formal/03-03-01>
-  *UML 2.0 Infrastructure Specification*
Recommandation, version 2.0, du 12-2003
Document sur <http://www.omg.org/cgi-bin/doc?ptc/2003-09-15>
-  *UML 2.0 Superstructure Specification*
Recommandation, version 2.0, du 08-2003
Document sur <http://www.omg.org/cgi-bin/doc?ptc/2003-08-02>
-  *UML 2.0 Diagram Interchange Specification*
Recommandation, version 2.0, du 01-09-2003
Document sur <http://www.omg.org/cgi-bin/doc?ptc/2003-09-01>
-  *UML 2.0 OCL Specification*
Recommandation, version 2.0, du 14-10-2003
Document sur <http://www.omg.org/cgi-bin/doc?ptc/2003-10-14>

TIRÈME SARL

XMI, XML Metadata Interchange

Rédaction : Pierre Attar

Recommandation(s) liée(s) : [UML - DTD - Schema](#) - [RELAX NG](#) - [DSDL](#)


« *XMI is a widely used interchange format for sharing objects using XML. Sharing objects in XML is a comprehensive solution that build on sharing data with XML. XMI is applicable to a wide variety of objects: analysis (UML), software (Java, C++), components (EJB, IDL, Corba Component Model), and databases (CWM). Over 30 companies have XMI implementations.* »

Source. XML Metadata Interchange Specification, version 2.0

Objectifs

Information descriptive non finalisée ; n'hésitez pas à nous contacter pour rédiger et/ou maintenir cette information à jour.

Recommandations(s)

 [XML Metadata Interchange Specification](#)
Recommandation, version 2.0, du 02-05-2003
Document sur <http://www.omg.org/cgi-bin/doc?formal/2003-05-02>